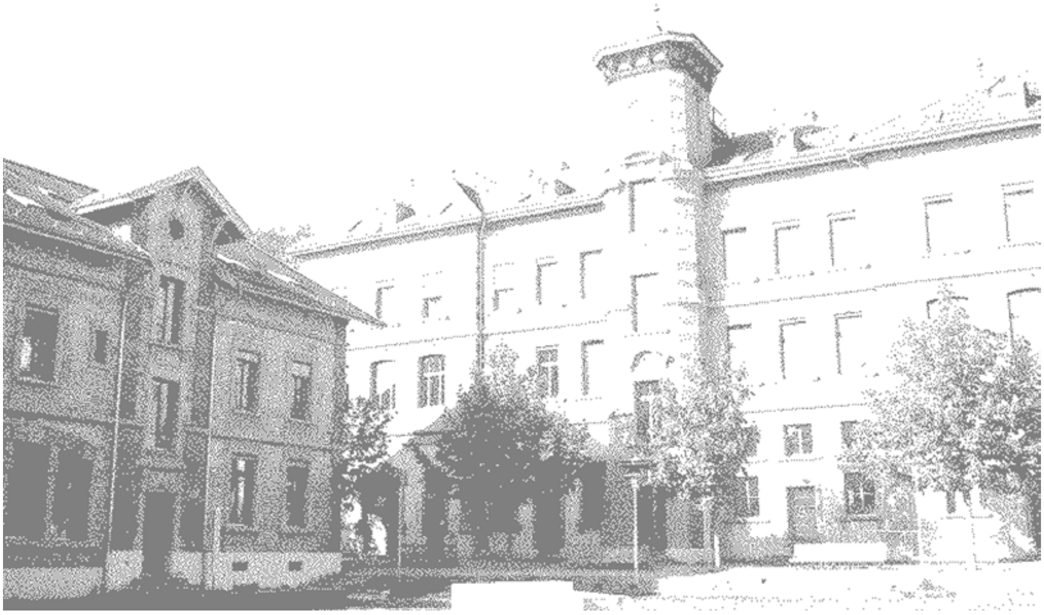
2014 Doctoral Workshop on Distributed Systems

H. Mercier, T. Braun, P. Felber, P. Kropf, P. Kuonen (eds.)

Technical Report IAM-14-001, July 24, 2014

Institut für Informatik und angewandte Mathematik, www.iam.unibe.ch



Doctoral Workshop on Distributed Systems

Hugues Mercier, Torsten Braun, Pascal Felber, Peter Kropf, Pierre Kuonen (eds.)

Technical Report IAM-14-001, July 24, 2014

CR Categories and Subject Descriptors:
C.2.1 [Computer-Communication Networks]: Network Architecture and Design; C.2.2 [Computer-Communication Networks]: Network Protocols; C.2.3 [Computer-Communication Networks]: Network Operations; C.2.4 [Computer-Communication Networks]: Distributed Systems

General Terms:
Design, Management, Measurement, Performance, Reliability, Security

Additional Key Words:
Wireless networks, content-centric networking, cloud computing, software transactional memory, fault tolerance, cellular networks, localization, wireless sensor networks, Internet of Things, privacy, security, energy efficiency, distributed file systems

Institut für Informatik und angewandte Mathematik, Universität Bern

Abstract

The Doctoral Workshop on Distributed Systems has been held at Kandersteg, Switzerland, from June 3-5, 2014. Ph.D. students from the Universities of Neuchâtel and Bern as well as the University of Applied Sciences of Fribourg presented their current research work and discussed recent research results. This technical report includes the extended abstracts of the talks given during the workshop.

2014 Doctoral Workshop on Distributed Systems

Hotel Alfa Soleil
Kandersteg
3 – 5 June

UNIVERSITÄT
BERN

UNIVERSITÉ DE
NEUCHÂTEL

ECOLE D'INGÉNIEURS ET
D’ARCHITECTES DE FRIBOURG

Workshop Program

Tuesday, June 3

Session 1 - Cloud Infrastructure

13:30 Cloud-Based Architecture for Environmental Modeling
Andrei Lapin

14:10 SLA-Driven Simulation of Multi-Tenant Scalable Cloud-Distributed Enterprise Information Systems
Alexandru-Florian Antonescu

14:50 Elastic Scaling of a High-Throughput Content-Based Publish/Subscribe Engine
Raphaël Barazzutti

Session 2 - Wireless Networks

16:00 Swarm Intelligence Based Construction Strategy for Multi-Objective Data Aggregation Tree in Wireless Sensor Networks
Yao Lu

16:40 Scheduling Policies Based on Dynamic Throughput and Fairness Tradeoff Control in LTE-A Networks
Ioan Comsa

17:20 Sensor Context-Aware Adaptive Duty-Cycled Opportunistic Routing
Zhongliang Zhao

Wednesday, June 4

Session 3 - Reliable and Secure Clouds

8:30 DynamiK: Lightweight Key Management for Privacy-Preserving Pub/Sub
Emanuel Onica

9:10 Modeling the Impact of Different Replication Schemes
Verónica Estrada Galiñanes

9:50 Building a Multi-site Consistent File System
Raluca Halalai

Session 4 - Network Measurements

11:00 Multi-Level Power Estimation in Virtualized Distributed Systems
Mascha Kurpicz

11:40 Time-Based Indoor Localization for Narrow-Band System
Zan Li

12:20 RSS-Based Indoor Localization: Algorithms, Results and Improvements
Islam Alyafawi

Thursday, June 5

Session 5 - Software Tools

8:30 TransGC: Concurrent Garbage Collection using Hardware Transactional Memory
Maria Carpen-Amarie

9:10 Multi-core Programming with Message Passing and Transactional Memory in the Actor Model
Yaroslav Hayduk

9:50 Enterprise Integration of Smart Objects using Semantic Service Descriptions
Matthias Thoma

Session 6 - Content-Centric Networks

11:00 Dynamic Transmission Modes to Support Opportunistic Information-Centric Networks
Carlos Anastasiades

11:40 Load Balancing in LTE Mobile Networks with Information-Centric Networking
André Gomes

12:20 Network Coding in Content-Centric Networks
Jonnahtan Saltarin

Workshop Proceedings

RSS-based Indoor Localization: Algorithms, Results, and Improvements

Islam Alyafawi
University of Bern
alyafawi@iam.unibe.ch

Abstract

This study deals with indoor positioning using GSM radio, which has the distinct advantage of wide coverage over other wireless technologies. In particular, we focus on passive localization systems that are able to achieve high localization accuracy without any prior knowledge of the indoor environment or the tracking device radio settings. In order to overcome these challenges, newly proposed localization algorithms based on the exploitation of the received signal strength (RSS) are proposed. We explore the effects of non-line-of-sight communication links, opening and closing of doors, and human mobility on RSS measurements and localization accuracy.

Keywords: Indoor localization; SDR system; proximity algorithms.

1 Introduction

In recent years, wireless devices became pervasive and more computationally powerful, making radio-based localization an attractive solution for the success of businesses in the commercial sector [1]. Our interest focuses on cellular networks and GSM (Global System for Mobile Communications) in particular due to its large scale adoption; it is still the cellular technology with largest coverage worldwide.

Depending on the participation of the tracked device, two types of localization systems can be distinguished [2]. In active systems the mobile device (MD) communicates with several anchor nodes (ANs) and the localization process is a collaborative effort. On the contrary, in a passive system the ANs are hidden for the target and only overhear the target's radio transmissions. RSS is widely used in wireless networks for link quality estimation and power control. It can be obtained from most off-the-shelf wireless network equipment. The easy RSS acquisition and relatively low deployment costs of the system makes RSS-based localization a preferable solution.

Passive localization techniques based on GSM signal overhearing have not been addressed in existing studies to our best knowledge [3]. GSM signals are actively used in fingerprinting with the involvement of the MD. Although fingerprinting has shown very good results for both WiFi and GSM, it is a cumbersome process, which needs periodic recalibration. Thus, we omit it. We aim to deliver a solution that copes with passive system requirements and show high localization accuracy without any prior knowledge of the deployment environment.

2 Work accomplished

In this section, we propose two novel proximity-based algorithms. The algorithms use RSS information to adapt the weights for the ANs in the set of ANs involved in the localization process. We consider two types of geometric methods: (1) the use of centroids and (2) the use of circumcenters as the target position. We take LWC as the benchmark method due to its common use by most studies. In LWC approach, the weight of AN_i is proportional to its RSS level, i.e., $w_i = \text{RSS}_i$.

2.1 Combined Differential RSS

Combined differential RSS (CDRSS) builds upon a differential RSS (DRSS) approach, in which the calculation of the weights does not rely on the absolute RSS values registered by each AN but on the difference in RSS between the ANs [4]. Working with absolute RSS has the drawback that different MDs may have different P_{tx}, making the organization of the RSS levels in ranges and their mapping to weights challenging. Considering the log-normal shadowing model and all ANs with equal gain settings, DRSS become as follows:

$$\text{DRSS}_{i,j} = P_{rx}(d_i) - P_{rx}(d_j) = 10\alpha\text{log}\left(\frac{d_j}{d_i}\right) - \psi'_{i,j} \tag{1}$$

where $\psi'_{i,j}$ is the difference between the ψ_i and ψ_j shadowing effects. Given that ψ_i and ψ_j are independent random variables, $\psi'_{i,j}$ has log-normal distribution with zero mean and $\sqrt{\sigma_i^2 + \sigma_j^2}$ standard deviation [5].

2.2 Weighted Circumcenter

The circumcenter is the center of the circumscribed circle around the polygon formed by the ANs such it is equidistant to all ANs. Using the geometric circumcenter has the same disadvantage as using the centroid, i.e., the same estimated location is taken for multiple target locations as long as the same set of anchor nodes is used. Therefore, we introduce the weighted circumcenter (WCC) concept, which allows us to shift the triangle circumcenter closer to the target's actual position by integrating information on the registered RSS levels.

2.3 Localization Accuracy

All experiments on localization accuracy were performed during working days when people were moving in the office and doors were not closed all the time. Table 1 contains the localization error mean μ and error deviation σ for all conducted experiments.

3 Work in progress and future work

The design of an indoor radio-based localization system is difficult because of the multiple paths taken by the radio waves and the various signal strength of each path [6, 7]. It is not easy to model the radio propagation in indoor environments due to specific site parameters, which vary from one environment to another such as floor layout, moving objects, and the number of reflecting surfaces. In future work, we plan to detect and filter the outlying measurements in the RSS measurement set before being fed to the localization algorithm.

Table 1: Error comparison of Centroid and Circumcenter based approaches (meters).

MD Location	LWC		CDRSS		WCC	
	μ	σ	μ	σ	μ	σ
L1	3.36	0.28	2.52	0.93	2.72	1.18
L2	6.45	0.09	4.47	0.20	3.19	0.02
L3	6.68	0.21	4.21	0.91	4.38	0.87
L4	3.76	0.08	2.68	1.49	2.14	1.46
L5	5.17	0.12	3.67	0.19	1.83	0.21
L6	1.94	0.76	0.74	0.17	1.02	0.48
L7	6.52	0.19	4.88	0.77	3.54	1.29
L8	8.42	0.20	4.50	0.19	3.67	1.18
L9	4.52	0.10	1.15	0.11	0.97	0.13
L10	5.91	0.11	3.67	0.20	2.01	0.51
L11	6.37	0.27	4.85	0.11	3.10	0.40
L12	2.25	1.15	2.17	1.54	2.00	2.65
L13	4.58	0.83	2.50	0.41	1.25	1.01
L14	0.57	0.06	2.95	1.01	2.32	1.89
Average	4.75	0.31	3.21	0.54	2.43	0.90

References

[1] A. Mesmoudi, M. Feham, and N. Labraoui. “Wireless sensor networks localization algorithms: a comprehensive survey”. CoRR abs/1312.4082, 2013.

[2] S. A. Ahson and M. Ilyas. “Location-based services handbook: Applications, technologies, and security”.

[3] G. Deak, K. Curran, and J. Condell. “A survey of active and passive indoor localization systems”. *Computer Communication*, pages 1939 – 1954, 2012.

[4] Y. H. Kwon, S. G. Choi, and J. K. Choi. “Movement detection mechanism using differential RSSI in mobile mpls network”. *8th International Conference Advanced Communication Technology*, pages 594 – 598, 2006.

[5] P. Agrawal and N. Patwari. “Correlated link shadow fading in multi-hop wireless networks”. *IEEE Trans.*

[6] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta. “A novel approach to indoor rssi localization by automatic calibration of the wireless propagation model”. *IEEE 69th Vehicular Technology Conference*, pages 1 – 5, 2009.

[7] B. Roberts and K. Pahlavan. “Site-specific RSS signature modeling for wifi localization”. *IEEE GLOBECOM*, pages 1 – 6, 2009.

Dynamic Transmission Modes to Support Opportunistic Information-Centric Networks

Carlos Anastasiades
University of Bern
anastasi@iam.unibe.ch

Abstract

We describe dynamic unicast adaptation to increase communication efficiency in opportunistic content-centric networks. The approach is based on broadcast requests to quickly find content and dynamically creating unicast links to the content sources without the need of neighbor discovery. The links are kept temporarily as long as they deliver content and they are quickly removed otherwise. Evaluations in mobile networks showed that this approach maintains flexibility to support seamless mobile communication and achieves up to 98% shorter transmission times as with broadcast. Dynamic unicast adaptation unburdens listener nodes from processing unwanted content resulting in lower processing overhead and power consumption at these nodes. The approach can be easily included into existing ICN architectures using only available data structures.

Keywords: information-centric networking; one-hop broadcast; wireless networks; mobility.

1 Introduction

Information-centric networking (ICN) has attracted much attention in recent years as a new networking paradigm for the future Internet. Forwarding is based on content names but not on endpoint identifiers. This makes ICN very attractive for communication in mobile and opportunistic networks, where neighbor nodes may change frequently. Instead of maintaining connectivity to a specific host, content can be retrieved from any neighbor node that holds the desired content. In this work, we base our investigations on the Named Data Networking (NDN) [1] architecture.

While wired Internet solutions use unicast faces, broadcast is used in wireless ICN networks [2], [3], [4], [5]. If next hops cannot be configured statically one-hop broadcast is required to find suitable content sources. However, as soon as a content source is identified, broadcast communication may not be favorable anymore due to several reasons. First, broadcast data rates are significantly lower than unicast rates resulting in considerably lower throughput. If the contact time between two nodes is short, broadcast throughput may not be enough to complete the content transfer in time. Second, broadcast requests may trigger multiple content replies from other nodes resulting in increased collision probability. To avoid collisions and enable duplicate suppression, content transmission must be delayed, which decreases broadcast throughput even more. Third, collisions cannot be quickly detected during broadcast due to missing MAC layer acknowledgments [6]. Fourth, broadcast content transmissions are received by all nodes in the vicinity including nodes that are not interested in the communication, i.e., passive listeners, resulting in increased processing overhead and higher power consumption at these nodes. We have performed measurements on wireless mesh nodes [7] and observed an increased power consumption of 22% at passive listeners compared to idle nodes that do not receive these messages.

Therefore, we investigate dynamic unicast adaptation during opportunistic ICN communication. Requesters can transmit broadcast requests to check if a content source is in the vicinity. If an answer is received, the following requests can be addressed directly to the content source. For this, we assume that every node has a unique node identifier that can be used as temporal locator of content similar to content-locator splits used in literature [8], [9], [10]. The ID can be temporarily included in the forwarding table and requesters can fall back to broadcast if the content source is not reachable anymore.

2 Work accomplished

We have implemented dynamic unicast adaptation in our CCN framework based on OMNeT++. The simulation framework allows us to better investigate collisions on MAC layer and scales well with a large number of mobile nodes. Evaluations in dense wireless environments with many content sources showed that broadcast requests result in many collisions and duplicate content transmissions, since multiple content sources reply to the requests. To reduce the number of collisions and decrease the time to retrieve content, broadcast content transmissions need to be delayed randomly within a specified interval enabling duplicate suppression at multiple content sources. For example, when increasing the broadcast delay from 10ms to 500ms, collisions can be reduced by 89% and the time a requester needs to retrieve content decreases by 82%.

However, if only a few content sources are present, large broadcast delays can result in longer content retrieval times. To avoid long transmission times, we introduced dynamic unicast adaptation. Requesters have the flexibility to use broadcast requests to find a content source but use unicast communication for subsequent data transmission. By that, very large broadcast delays can be used without having a significant impact on content retrieval time. Evaluations have shown that dynamic unicast adaptation can lead up to 98% faster transmission times compared to the best broadcast results. Additionally, passive listeners receive at least 76% fewer content messages and transmit 63% fewer content messages, resulting in smaller processing overhead, enabling them to save energy.

If communication is performed via unicast, only requesters receive the complete content and no other nodes in the vicinity, reducing content density in the network. However, cached content in temporary caches may disappear quickly and we have shown that content density can be maintained by requesters providing previously requested content publicly in extended persistent storage. By that, content retrieval times can be reduced by 90% because requesters can act as content sources and the load, i.e., transmitted content messages, of original content sources are reduced by 95%.

3 Work in progress and future work

We have also integrated dynamic unicast adaptation in the current CCNx 0.8.2 release. Initial evaluation results show similar performance gains as achieved in our simulations. Additionally, we have included an extension that enables content sources to detect multiple concurrent content flows to enforce broadcast communication if a certain threshold of unicast request is reached. As part of our future work, we will evaluate this implementation in mobile scenarios using NS3-DCE [11], an extension of the NS3 network simulator for direct code execution.

In this work, we considered only one-hop communication where routing is replaced by the mobility of the nodes. However, if requesters never meet a content source directly, multi-hop

Interest forwarding is required. We are currently investigating limited Interest forwarding via hop counter along a few hops to find content just outside the requesters' transmission range. Additionally, since NDN communication is synchronous, i.e., data needs to travel the reverse path of Interests, we investigate agent-based content retrieval [12] to support asymmetric opportunistic communication. Requesters can delegate content retrieval to agents that have higher probability to meet a content source and retrieve content if they meet again. We have implemented agent-based content retrieval in C and we are evaluating it using both push- and pull-based notifications in mobile environments using NS3-DCE.

To support content retrieval from mobile nodes, content can be optionally stored in home repositories, which are continuously connected to the Internet. The approach is similar to custodian-based information sharing [13], but instead of storing only a link to content on mobile devices, which may be susceptible to intermittent connectivity, the entire content is synchronized upon connectivity to the Internet. We have implemented the home repository protocol and are currently evaluating its overhead in terms of transmitted messages and energy.

If caching is extended to persistent storage on repositories, cache replacement strategies need to be deployed in repositories as well. In the current CCNx 0.8.2 release it is not possible to delete content from repositories. We have extended the current repository implementation by a delete queue that lists content that has not been requested recently and can be deleted if required. By that, the approach implicitly considers popularity and age of stored content. We are currently evaluating the implementation with different content distributions and popularity.

Knowing available content names is a fundamental requirement to enable information-centric communication. For this reason, we have implemented multiple content discovery algorithms to efficiently traverse name trees of reachable content and enabling opportunistic content retrieval. We have also implemented alias mapping to support location-based content discovery to quickly find relevant content in a requester's environment such as, e.g., a nearby temperature sensor, without any central communication infrastructure. We are currently evaluating discovery and alias mapping mechanisms using NS3-DCE.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Network Named Content," *5th ACM CoNEXT*, pp. 1-12, Rome, Italy, December 2009.

[2] M. Meisel, V. Pappas, and L. Zhang, "Listen-First, Broadcast Later: Topology-Agnostic Forwarding under High Dynamics," *ACITA*, pp. 1-8, London, UK, September 2010.

[3] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang, "Data naming in vehicle-to-vehicle communications," *IEEE Infocom Computer Communication workshop*, pp. 328-333, Orlando, FL, USA, March 2012.

[4] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Rapid traffic information dissemination using named data," *ACM MobiHoc workshop on Emerging Name-Oriented Mobile Networking Design*, pp. 7-12, Hilton Head, SC, USA, June 2012.

[5] M. Amadeo, C. Campolo, A. Molinaro, and N. Mitton, "Named data networking: A natural design for data collection in wireless sensor networks," *IEEE Wireless Days*, pp. 1-6, Valencia, Spain, November 2013.

[6] C. Anastasiades, T. Schmid, J. Weber, and T. Braun, "Opportunistic Content-Centric Data Transmission During Short Network Contacts," *IEEE WCNC*, Istanbul, Turkey, April 2014.

[7] J. Weber, "Automatic detection of forwarding opportunities in intermittently connected content-centric networks," *Bachelor thesis, University of Bern*, March 2013.

[8] F. Hermans, E. Ngai, and P. Gunningberg, "Mobile sources in an information-centric network with hierarchical names: An indirection approach," *7th, SNCNW*, Linköping, Sweden, May 2011.

[9] R. Ravindran, S. Lo, X. Zhang, and G. Wang, "Supporting Seamless Mobility in Named Data Networking," *IEEE FutureNet V*, pp. 5854 - 5869, Ottawa, ON, June 2012.

[10] D.-H. Kim, J.-H. Kim, Y.-S. Kim, H. S. Yoon, and I. Yeom, "Mobility support in content centric networks," *IEEE Sigcomm ICN workshop*, Helsinki, Finland, August 2012.

[11] NS-3: Direct Code Execution,
`http://www.nsnam.org/overview/projects/directcode-execution/`, June 2014

[12] C. Anastasiades, W. El Maudni El Alami, and T. Braun, "Agent-based Content Retrieval for Opportunistic Content-Centric Networks," *12th WWIC*, Paris, France, May 2014.

[13] V. Jacobson, R. L. Braynard, T. Diebert, and P. Mahadevan, "Custodian-based information sharing," *IEEE Magazine*, vol. 50, no. 7, pp. 38-43, July 2012

SLA-Driven Simulation of Multi-Tenant Scalable Cloud-Distributed Enterprise Information Systems

Alexandru-Florian Antonescu
University of Bern
SAP (Schweiz) AG
antonescu@iam.unibe.ch

Abstract

Cloud Computing enables provisioning and distribution of highly scalable services in a reliable, on-demand and sustainable manner. Distributed Enterprise Information Systems (dEIS) are a class of applications with important economic value and with strong requirements in terms of performance and reliability. It is often the case that such applications have complex scaling and Service Level Agreement (SLA) management requirements. In order to validate dEIS architectures, stability, scaling and SLA compliance, large distributed testing deployments are often necessary, adding complexity to the design and testing of such systems. To fill this gap, we present and validate a methodology for modeling and simulating such complex distributed systems using the CloudSim cloud computing simulator, based on measurement data from an actual dEIS distributed system. We present an approach for creating a performance-based model of a distributed cloud application using recorded service performance traces. We then show how to integrate the created model into CloudSim. We validate the CloudSim simulation model by comparing performance traces gathered during distributed concurrent experiments with simulation results using different VM configurations. We demonstrate the usefulness of using a cloud simulator for modeling properties of real cloud-distributed applications by comparing SLA-aware scaling policies using CloudSim simulator and dEIS statistical model.

Keywords: Cloud Computing, Enterprise Information Systems, Service Level Agreements, Scaling, Simulation

1 Introduction

Cloud Computing enables provisioning and distribution of highly scalable services. Often, applications running in these distributed environments are designed to concurrently execute workload, for example, distributed Enterprise Information Systems (dEIS) often interact with large-scale distributed databases.

In order to validate certain properties of cloud systems, distributed experiments with varying number of virtual machines (VMs) are required. However, getting access to such large, distributed testbeds can be difficult, and reproducing results is often not possible, due to the use of shared physical computing and network resources. This leads to the need of modeling cloud applications and resources in order to perform simulations for testing "cloud-aware" software management algorithms.

We set to accurately describe and model a distributed dEIS application based on profiling monitoring information obtained from running a small-scale demonstrator in a physical testbed. We then integrate the generated dEIS model in a cloud simulator and then we compare the performance and efficiency of different Service Level Agreement (SLA)-based resource management policies.

2 Performance Analysis and Modeling of a Distributed Enterprise Information System

A typical dEIS application consists of the following tiers, each contributing to the SLA management problem: consumer/thin client, load balancer, business logic and storage layer. This class of systems is representative for core enterprise management systems, such as ERP [1]. Fig. 1 provides an overview of the overall EIS topology. A detailed description of each dEIS service can be found in [2], [3] and [6].

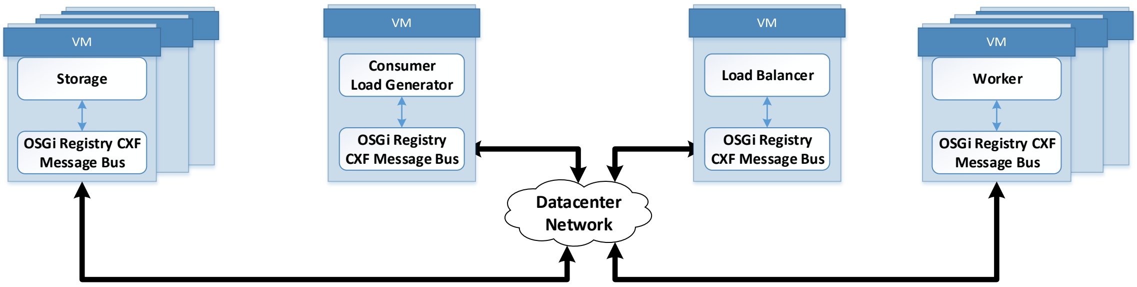

VM
Storage
OSGi Registry CXF Message Bus
VM
Consumer Load Generator
OSGi Registry CXF Message Bus
VM
Load Balancer
OSGi Registry CXF Message Bus
VM
Worker
OSGi Registry CXF Message Bus
Datacenter Network

Figure 1: dEIS Architecture

In order to simulate the dEIS application, we first recorded key performance indicators of dEIS services under constant workload. We then used these monitoring traces to create the dEIS simulation model composed of a large number of single requests' execution measurements. For a given request we measured the round-trip times for the service calls, as well as the duration of the service-local calls.

For building the application performance profile we applied the following algorithm. We instantiated one VM for each (Consumer) CS, (Load Balancer) LB, (Worker) WK and (Storage) ST services. We initialized the concurrent *load* with 1 and the benchmark time duration with 10 minutes. Every 10 milliseconds the number of active requests ar executed by the system was compared to the target *load*, and if ar was below *load*, then a number of $load - ar$ CS requests would be generated. For each end-to-end request, the following parameters were recorded: round trip durations, service-local operation durations and VM-level performance metrics (CPU, memory, network, disk). By recording all these parameters for each distributed request, a performance benchmark profile was generated under the constant workload *load*.

After running the benchmark described above, the dEIS performance profiles for constant concurrent load between 1 and $max.load$ will be known and the performance profile will be generated with all service call round-trip times and service-local operation durations. This profile will then be used during the simulation.

The performance-profiling traces previously gathered can be represented as shown in Eq. 2. $Profile_{load}$ is a matrix with all measurable time values of applications' operations under concurrent *load*. $RT(S_i, S_j)$ is the round-trip time of a remote service call on service S_j from service S_i. $D_i(Op_k)$ is the time of performing service-local operation k on service S_i and $(cpu|mem|net)_{CS|LB|WK|ST}$ is the utilization level of CPU, memory and network on the VMs corresponding to CS, LB, WK and ST services. Each line of this matrix corresponds to a single chain of requests from CS, to WK, to ST, and then back to CS.

$$Profile_{load} = \left\langle \begin{array}{l} RT(S_i, S_j),\ D_i(Op_k), \\ (cpu|mem|net)_{CS|LB|WK|ST} \end{array} \right\rangle \qquad (2)$$

By combining performance profiles for concurrent workload between 1 and $max.load$ end-to-end requests, we form the dEIS performance model $Profile$ as dictionary with key $load$ and corresponding values $Profile_{load}$.

In order to convert the created performance profile to CloudSim [8] entities we transform the durations in milliseconds to their equivalent MIPS rating. We also map the VM resource consumption for CPU, memory and network to CloudSim characteristics of the created cloudlets - CPU and memory *utilization models* and network input and output payloads. More details about transforming the profiling information into CloudSim models, along with examples of how the models are then used for concurrent simulations can be found in [3].

3 SLA-Based Scaling Manager

The SLA Scaling Manager (SSM) is the component responsible for dynamically adjusting the number of VMs for each of the services of the distributed applications and for each of the cloud tenants (company owning virtual cloud resources). It accomplishes this using invariant conditions formed with terms obtained from the performance indicators of the services running in VMs. An example of an invariant condition can be: "average distributed transaction execution time is below one second". The threshold contained in the SLA invariant is used by the SSM for determining the conditions for performing either a scale-out action [9] (creating one or more VMs), or a scale-in action (terminating one or more VMs).

The SSM operates in a loop by calculating the SLA ratio sr as the factor by which the average over the moving time window W of SLA metric m is approaching its maximum threshold $max_{SLA}(m)$. If sr is above a given threshold S^{UP} (e.g. 0.9) and sr is increasing from the last check then a scale-out operation is flagged. Similarly, if sr is below a threshold S^{DOWN} (e.g. 0.6) and sr is decreasing, then a scale-in operation is flagged. Either scale-out or scale-in operations will be executed only if the number of such operations ss is below a given threshold ss^{MAX} (e.g. 2) in the last W_S seconds (e.g. 40 sec, chosen as 1.5 times the time it takes for a VM to become fully operational), for ensuring system stability by preventing (1) fast-succeeding transitory scale-in and scale-out actions, and (2) oscillations in the number of VMs.

In order to validate both the accuracy of the dEIS simulation models in representing real-world workloads we performed a number of experiments and simulations [4] [3]. We have also performed different simulations with different cloud tenants, each having individual SLA policies, showing that it is possible to use the dEIS profiling models for accurate simulations of dynamic cloud workloads.

4 Conclusions and Future Work

We have presented an approach for building a simulation model of a distributed application for concurrent workload processing, by analyzing application's performance traces gathered in small-scale real deployments. We then showed how to integrate the simulation model into CloudSim and how to build a simulation with varying concurrent workload. We have also presented an approach for modeling concurrent computing workloads using the CloudSim simulator.

We have also proposed and validated a CloudSim model for translating application-level performance profiling information to VM-level CloudSim scheduler resource utilization level. We have also identified some possible optimization points in cloud infrastructure management, regarding energy consumption of idle servers. We have shown that SLA guarantees can be used for VM scaling purposes when it is possible to convert them to SLA ratios.

As future work we consider evaluating more complex scaling algorithms by using prediction of both the workload and the SLA usage ratios.

References

[1] Alexis Leon. 2008. Enterprise resource planning. Tata McGraw-Hill Education.

[2] A-F Antonescu and T Braun. 2014. Improving Management of Distributed Services Using Correlations and Predictions in SLA-Driven Cloud Computing Systems. In Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS). IEEE, Poland, Krakow.

[3] A-F Antonescu and T Braun. 2014. Modeling and Simulation of Concurrent Workload Processing in Cloud-Distributed Enterprise Information Systems. In Proc. ACM SIGCOMM Workshop on Distributed Cloud Computing (DCC 2014).

[4] A-F Antonescu and T Braun. 2014. SLA-Driven Simulation of Multi-Tenant Scalable Cloud-Distributed Enterprise Information Systems. In Workshop on Adaptive Resource Management and Scheduling for Cloud Computing, Held in conjunction with PODC 2014, Paris, France, on July 15th, 2014

[5] A-F Antonescu, A-M Oprescu, and others. 2013. Dynamic Optimization of SLA-Based Services Scaling Rules. In Proc. 5th IEEE Internetional Conference on Cloud Computing Technology and Science (CloudCom).

[6] A-F Antonescu, P Robinson, and T Braun. 2012. Dynamic Topology Orchestration for Distributed Cloud-Based Applications. In Proc. 2nd IEEE Symposium on Network Cloud Computing and Applications (NCCA).

[7] A-F Antonescu, P Robinson, and T Braun. 2013. Dynamic SLA Management with Forecasting using Multi-Objective Optimizations. In Proc. 13th IFIP/IEEE Symposium on Integrated Network Management (IM).

[8] T. Goyal et al. Cloudsim: simulator for cloud computing infrastructure and modeling. Procedia Engineering, 2012.

[9] Massimo Re Ferre. 2004. Vmware ESX server: scale up or scale out. IBM Redpaper (2004).

Elastic Scaling of a High-Throughput Content-Based Publish/Subscribe Engine

Raphaël P. Barazzutti
Université de Neuchâtel
raphael.barazzutti@unine.ch

Abstract

Publish/subscribe pattern is an ideal candidate for composing distributed applications on cloud environments thanks to its ease of use and its flexibility. Publish/-subscribe infrastructure provisioning is a tough challenge, the amount of stored subscriptions and incoming publications rate varies over time, and the computational cost depends on the nature of the applications and in particular on the filtering operation required by them (e.g. content-based vs topic-based, encrypted- vs non-encrypted filtering). Thanks to the possibility of providing additional computational resources dynamically on cloud environments, our system aims to provide elasticity. The ability to elastically adapt the amount of resource required to sustain given throughput and delay requirement is key to achieving cost-effectiveness for a pub/sub service running in a cloud environment.

We present the design and the evaluation of an elastic content-based pub/sub system: e-StreamHub. Specific contributions of this paper include: a mechanism for dynamic scaling, both out and in, of stateful and stateless pub/sub operators. a local and global elasticity policy enforcer maintaining high system utilization and stable end-to-end latencies. an evaluation using real-world tick workload from the Frankfurt Stock Exchange and encrypted content-based filtering.

Keywords: Content-based filtering; Scalability; Performance; Elastic scaling

1 Introduction

Elasticity frees cloud users from the need of statically provisioning their infrastructure. Thanks to elasticity, resources can be added (scale out) or removed (scale in) according to the variations of a workload over time.

In order to support elasticity an application has to provide three functionalities. First, it should be *scalable*, secondly it should support *dynamic allocation of resources* and finally it should provide a *decision support*.

2 Work accomplished

Since our implementation, StreamHub, was already providing *scalability*[1], we focused in adding the required features in order to support elasticity.

The overall performance of an *elastic system* depends quite much on the cost of the migrations, for this reason we first evaluated the impact of migrations. The latency is defined as the time elapsed from the submission of a publication in StreamHub and the actual delivery to the interested subscriber.

The measurement of the latency of the three operators (Access Point, Matcher, Exit Point) is low and predictable (see figure 2 and table 3).

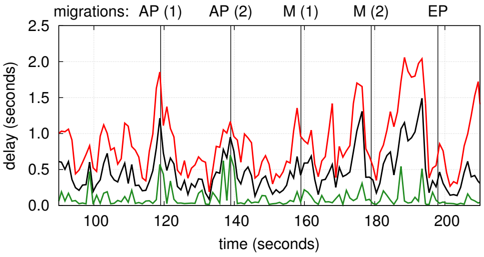
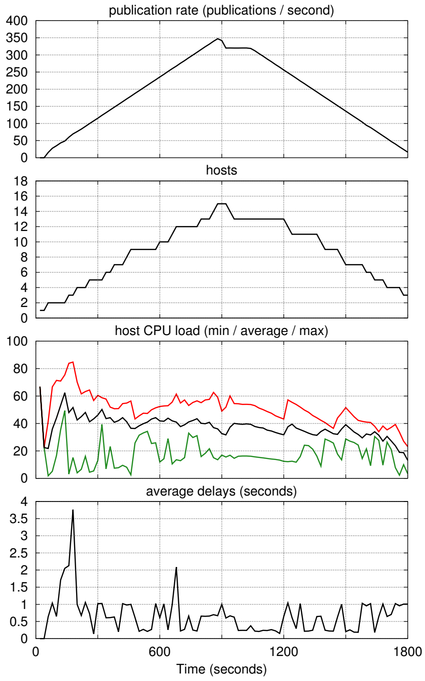


Figure 2: Impact of migrations on latency

	AP	**M** (50 K)	**EP**
average	232 ms	2.53 s	275 ms
std. dev.	31 ms	1.56 s	52 ms

Figure 3: Migration times, 100 publications/s, 50 K subscriptions stored per M operator slice, for a total of 500 K subscriptions, respectively.

The actual elasticity evaluation on was done E-STREAMHUB on a fully synthetic workload (fig. 4) and on a workload based on real traces of the Frankfurt stock exchange (fig. 5).This experience on E-STREAMHUB exhibits well the ability of E-STREAMHUB to allocate and release dynamically resources in order to sustain the load on the system.



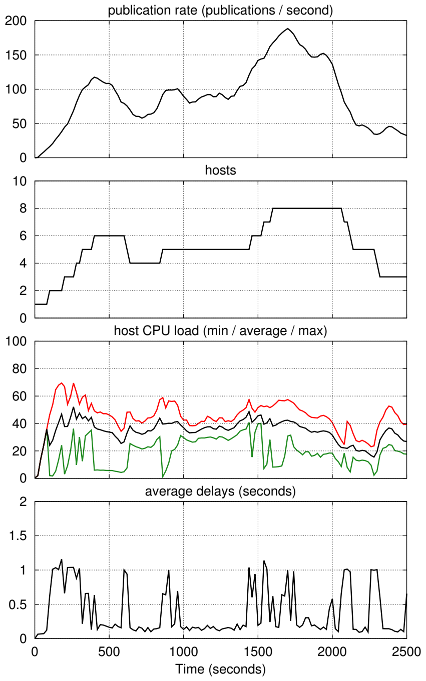


Figure 4: Elastic scaling under a steadily increasing and decreasing synthetic workload and 20 K encrypted subscriptions.

Figure 5: Elastic scaling under load from the Frankfurt stock exchange.

3 Work in progress and future work

Thanks to its architecture, STREAMHUB's libfilter[1] is extensible to use any kind of matching schema. Most of stock exchange models proposed in literature (like Meghdoot [6]) describe more what is done at the level of a broken/bank connected to the stock exchange than the actual matching done by the stock exchange itself.

The matching at the level of the stock exchange is a interesting problem in the case of publish/subscribe systems. Messages (sell or buy orders) are somehow subscriptions and publications at the same time. The system has to provide some extra guarantees since a single order cannot be matched twice.

In order to provide a more realistic simulation of the actual computation done at the stock exchange and replay them on e-StreamHub, the activity on stocks of the Swiss market has been recorded during one month. Each stock was monitored during that period and the variations in its order book were recorded. During the data harvesting phase, the list of all instruments (stocks), the orderbook status, and the list of paid prices are retrieved. After this phase, post-processing is required in order to reconstruct the feed of (buy/sell) orders sent to the stock exchange.

References

[1] Raphaël Barazzutti and Pascal Felber and Christof Fetzer and Emanuel Onica and Jean-François Pineau and Marcelo Pasin and Etienne Rivière and Stefan Weigert "StreamHub: a massively parallel architecture for high-performance content-based publish/subscribe", *Proceedings of the 7th ACM International Conference on Distributed Event-Based Systems, 2013*

[2] P.T. Eugster and P. Felber and R. Guerraoui and A.-M. Kermarrec,"The Many Faces of Publish/Subscribe",*ACM Computing Surveys*, 2003

[3] A. Carzaniga and D. S. Rosenblum and A. L. Wolf, "Design and Evaluation of a Wide-Area Event Notification Service", *ACM Theoretical Computer Science*, 2001

[4] R. Chand and P. Felber, "Scalable distribution of XML content with XNet", *IEEE Transactions on Parallel and Distributed Systems*, 19, 2008

[5] Y. Yoon and V. Muthusamy and H.-A. Jacobsen,"Foundations for Highly Available Content-based Publish/Subscribe Overlays", *International Conference on Distributed Computing Systems*, 2011

[6] Abhishek Gupta and Ozgur D. Sahin and Divyakant Agrawal and Amr El Abbadi, "Meghdoot: Content-Based Publish/Subscribe over P2P Networks.", *Proceedings of the Middleware 2004*, 2004

TransGC: Concurrent Garbage Collection using Hardware Transactional Memory

Maria Carpen-Amarie
Université de Neuchâtel
maria.carpen-amarie@unine.ch

Abstract

The garbage collector (GC) is a fundamental component of managed runtime environments like the Java virtual machine. It automatically frees the memory and prevents the application from accessing invalid references. While garbage collection necessarily pauses the application when reclaiming unused memory on a single-core computer, a large multi-core system would be able to use some of the cores for the collection and let the others continue executing the application. Having such a concurrent collector is highly desirable as it avoids the long pauses caused by GC activity; it is also challenging to implement: one must prevent data races caused by concurrent memory accesses from the application and from the collector. Hardware Transactional Memory (HTM) brings an elegant solution to this problem. Objects are copied during a transaction, which might aborted if the application interacts with the relocated object. The objective of this project is to develop a fully concurrent GC that can exploit the HTM facilities of Intel's new Haswell CPUs.

Keywords: garbage collection; JVM; HTM; concurrency.

1 Introduction

Nowadays, the computers' performance is mainly improved by constantly increasing the number of cores per CPU. However, the concurrent programming paradigms tend to lag behind, becoming overly difficult to implement a correct and efficient concurrent system on today's powerful servers. The use of classical concurrency mechanisms (i.e., locks, mutexes, barriers, etc.) is complex and error-prone, leaving room for many synchronization hazards, such as race conditions or deadlocks. In addition, it is very difficult to identify and repair this kind of bugs. **Transactional memory** [2] is a new concept which offers a feasible solution for decreasing both the development effort level, as well as concurrency bugs occurrence. It basically encapsulates several operations into transactions which are committed atomically. If more threads try to access a critical section at the same time, only one of them will succeed and make visible its changes, while all others will abort their transactions and possibly retry afterwards. It was argued that for Software Transactional Memory, most of the benefits are outweighed by a few essential drawbacks [1, 3]: e.g., a significant overhead due to the code instrumentation, leading to a decreased overall performance. However, this is about to change with the new Haswell processor from Intel, which offers full support for **Hardware Transactional Memory** (HTM). This means that it is able to speculatively execute small transactions (limited by the cache size) and to automatically replace locks with transactions when efficient. Because of the transaction size limit, the solution is not suitable for all concurrent systems, but it perfectly matches the need to handle specialized concurrency issues, such as *garbage collection* [4, 5, 6, 7]. The **Garbage Collector** (GC) represents a critical element in any managed environment, such as Java Virtual Machine (JVM). Its purpose is to automatically reclaim unused memory and protect the application from accessing invalid references. In order to do that, it must stop the application threads while the collection takes place. This happens in single core, as well as in multicore environments.

We start by carefully analyzing the length of the GC pauses, which also gives the motivation of this project. We discover that the GC pauses can reach hundreds of seconds, which is unacceptable for today's applications. As a solution, we propose to implement a *fully concurrent garbage collector, using hardware transactional memory.*

2 Current work: analysis and approach

This section presents the steps we took in the attempt of solving the problems introduced by the GCs in real-life applications. This is still work in progress, thus we only describe the concepts of the algorithms.

2.1 GC pauses analysis

We designed a set of experiments to help us understand the impact of the stop-the-world GC pauses on the application execution. For this, we ran Cassandra DB 2.0.0 [10] in combination with the YCSB Client 1.0 benchmark [9], on a 48-core server, with a 64GB RAM memory. As GC, we used ParallelScavenge [11], one of the most widely used generational collectors, with a heap size of 64GB and a young generation size of 12GB.

We based our experiments on the two possible behaviors of the client benchmark:

- **loading phase**: the client populates the database with records. We ran the experiment with 100 client threads in parallel for one hour (Figure 6(a)) and two hours, respectively (Figure 6(b)). The figures show the duration of the GC pause on the Y axis and the timestamps when the GCs were measured on the X axis. The former shows no full GC taking place; however, the collection of the young generation reaches a peak pause of around 17s, while having several other peaks over 2s. We observe in the latter a full GC introducing a pause duration of more than 160s. The young generation collections are up to 25s.

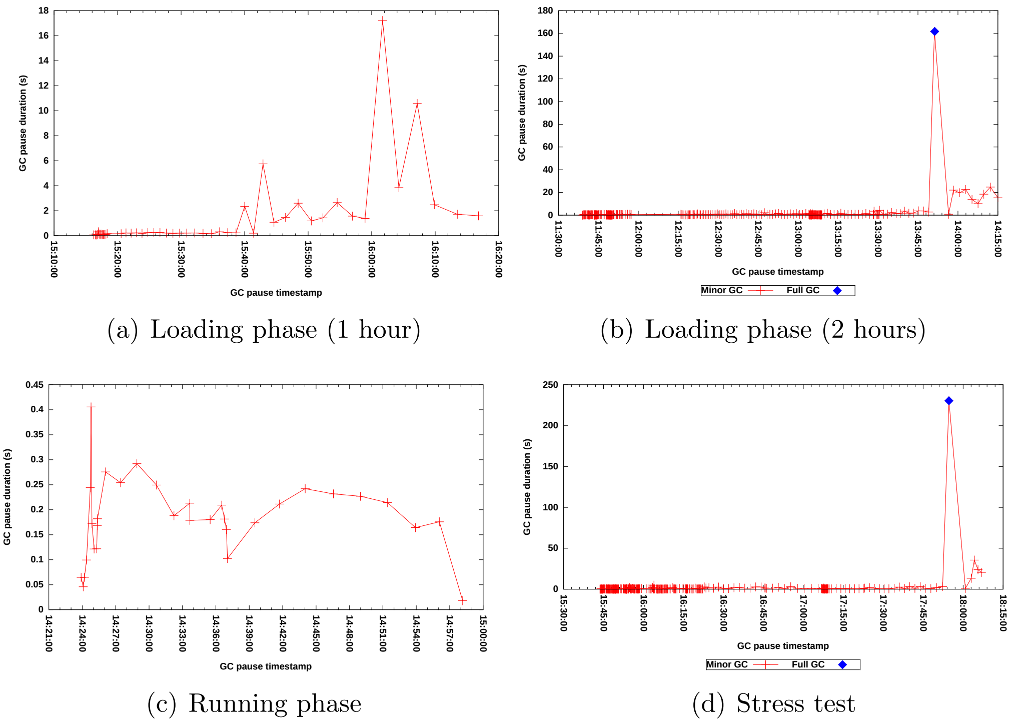
- **running phase**: the client runs a user specified workload on the DB. YCSB benchmark provides a few workloads, that can be further modified or extended by the user. We used a default workload having 50% read operations and 50% update operations, on 50 client threads during an hour. In this case, there was no full GC and the minor GCs paused the application for less than 0.5s, as showed in Figure 6(c).

- **stress test**: we carefully studied Cassandra's internal structures and observed that it benefits from an in-memory caching system. In order to offer fault tolerance, for each insert into the DB the information is written both in the memory, as well as in a *commit log*. The information is flushed to disk if a user-defined caching threshold is exceeded. We configured Cassandra to flush as rarely as possible its records to disk and stress the memory until the server is saturated. Figure 6(d) shows a full GC lasting around 4 minutes.

In conclusion, the *running phase* of the YCSB client did not saturate the server's memory enough, leading only to short pauses. However, the insert-intensive workload of the first phase and the stress test prove that real-life situations allow for significant application stalls, sometimes unacceptably long, depending on the application requirements.

2.2 High level approach

When we detail the response time with Cassandra using YCSB, we can see that most of the highest response-time latencies are related to the GC. This is because during a GC, the



(a) Loading phase (1 hour)

(b) Loading phase (2 hours)

(c) Running phase

(d) Stress test

Figure 6: Application threads pauses for the experiments presented in Section 2.1

application is stopped in order to avoid race conditions when the GC copies the live objects. In JVM, the mechanism to stop all mutators is called **safepoint**.

Our aim is to replace the locking mechanism for the application threads with transactions. Thus, the application will be able to continue working concurrently/speculatively with the GC threads. We try to implement this change at a high level on the JVM's code: when an application thread enters the safepoint handler method (Safepoint::begin()), it will start a hardware transaction and continue the execution. The transaction is committed when the purpose for Safepoint ends.

The major drawback of this high level approach is the size of the transactions. We ran a series of tests and discovered that the abort rate was ~86%, going up to 97%. 32% of the aborts were due to transaction overflow. Moreover, the GC pause times did not decrease in our tests, but were more or less equal to those given by the initial locking strategy. In conclusion, a high level replacement scheme of the locking algorithm with transactions results in workloads too big to be accommodated by HTM, which subsequently overflows often.

2.3 Low level approach

As the logical next step after the results obtained in Section 2.2, we devised a low level approach. Replacing the current blocking mechanism at instruction level would guarantee a suitable size for the transactional workload, resulting in no overflow aborts.

For this algorithm, we assume a garbage collector employing a **Brooks forwarding pointer** [8]. That is, a pointer that indicates the address of the primary copy of an object already copied by the GC to its new location. This is the current behavior for the Concurrent Mark-Sweep collector (CMS) in JVM. Besides this, the application threads in JVM currently benefit from the so-called **access barriers** that protect every store operation of an object. The idea of this algorithm is to encapsulate the access barriers in transactions. The forwarding pointer will indicate if the object is in the process of being copied. Thus, if the object is "busy", then the transaction will abort and retry; otherwise, the transaction will commit and the application thread will continue working without being blocked by the GC actions.

3 Conclusions and future work

Managed Runtime Environments such as JavaVM are used world-wide; the Garbage Collector is a critical component in these systems. Our current work proved that the necessary GC pauses are unacceptable for real-life applications. We propose to solve this problem by replacing the current GC locking system with *hardware transactions*. This implies going deep into the machine code of the JVM, in order to wrap in transactions a suitable workload for HTM. We plan to implement this low level algorithm and thoroughly test it. Finally, the algorithm will be further optimized in order to be cost-effective (the transaction will protect as many instructions as accepted by the cache line) and will provide a fallback path.

References

[1] C. Cascaval, C. Blundell, M. Michael, H. W. Cain, P. Wu, S. Chiras, and S. Chatterjee. “Software Transactional Memory: Why Is It Only a Research Toy,” *Queue - The Concurrency Problem*, vol. 6, no. 5, p. 40, 2008.

[2] M. Herlihy and J. E. B. Moss. “Transactional memory: architectural support for lock-free data structures,” *Proc. 1993 International Symposium on Computer Architecture (ISCA)*, 1993, pp. 289-300.

[3] A. Dragojevic, P. Felber, V. Gramoli, and R. Guerraoui. “Why STM can be more than a research toy,” *Communications of the ACM*, vol. 54, no. 4, pp. 70-77, 2011.

[4] A. Dragojević, M. Herlihy, Y. Lev, and M. Moir. “On the power of hardware transactional memory to simplify memory management,” *Proc. 2011 SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, 2011, pp. 99-108.

[5] P. McGachey, A.-R. Adl-Tabatabai, R. L. Hudson, V. Menon, B. Saha, and T. Shpeisman. “Concurrent GC leveraging transactional memory,” *Proc. 2008 SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)*, 2008, pp. 217-226.

[6] M. T. Higuera-Toledano. “Using Transactional Memory to Synchronize an Adaptive Garbage Collector in Real-Time Java,” *Proc. 2011 International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW)*, 2011, pp. 152-161.

[7] B. Iyengar, G. Tene, M. Wolf, and E. Gehringer. “The Collie: a wait-free compacting collector,” *ACM SIGPLAN Notices*, vol. 47, no. 11, pp. 85-96, 2012.

[8] Rodney A. Brooks. “Trading data space for reduced time and code space in real-time garbage collection on stock hardware,” *Proc. 1984 Symposium on Lisp and Functional Programming (LFP)*, 1984.

[9] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. “Benchmarking cloud serving systems with YCSB,” *Proc. 2010 Symposium on Cloud Computing (SoCC)*, 2010, pp. 143-154.

[10] The Apache Cassandra Project. *http://cassandra.apache.org/*, 2014.

[11] Whitepaper: Memory management in the Java Hotspot virtual machine. *http://java.sun.com/j2se/reference/whitepapers/memorymanagement_ whitepaper.pdf*, 2006.

Scheduling Policies Based on Dynamic Throughput and Fairness Tradeoff Control in LTE-A Networks

Ioan Comsa
École d'ingénieurs et d'architectes de Fribourg
Ioan.Comsa@edu.hefr.ch

Abstract

We address the problem of radio resource scheduling subject of dynamic fairness constraints. Basically, in the OFDMA cellular networks there is a fundamental trade-off between the cell throughput and fairness levels for preselected users which are sharing the same amount of resources at one transmission time interval (TTI). The scheduling policies which are proposed so far are not able to maintain a satisfactory level of fairness at each TTI when a very dynamic environment is considered. In this sense, we propose a novel controller that interacts with LTE-A scheduler by adapting the level of fairness depending on the learned policies. The fairness requirement dynamically changes at each TTI based on observed normalized user throughputs, channel conditions and traffic load. Therefore, five reinforcement learning (RL) algorithms (Q-Learning, QV–Learning, SARSA, ACLA and CACLA) are used in order to find a proper scheduling metric at each TTI for the best matching conditions subject of different fairness constraints. The results indicate that, our proposals are able to assure a faster convergence to the optimal solution in comparison with other existing methods. Finally we prove that the system throughput gain achieves a better performance when the reinforcement learning algorithms are used by minimizing in the same time the percentage of TTIs when the system is declared unfair.

Keywords: LTE-A, TTI, CQI, throughput, fairness, scheduling rule, RL, policy, Q, QV, SARSA, ACLA, CACLA.

1 Introduction

In LTE cellular networks, the packet scheduling and resource allocation represent major and problematic concerns in terms of the objectives to be achieved. Maximizing the sector/cell spectral efficiency subject of different QoS/fairness constraints is not a trivial job since the multi-objective optimization has to be addressed at each TTI. For this purpose, the multi-objective optimization has to maximize the total cell throughput maintaining the system to the fairness performance acceptability border. One way to increase the network capacity is to adopt opportunistic schedulers or channel aware scheduling rules that exploit the multiuser diversity principle. In this way, due to the temporal and frequency diversity, users with relatively better channel conditions are preferred to be scheduled leading to the unfair treatment of users which are located at the cell border, affecting the performance of the cell edge spectral efficiency. Therefore, a special care of the tradeoff concept between system throughput and user fairness should be taken into consideration.

When channel conditions are considered for the fairness degree evaluation, the average user throughput is used by different evaluation metrics. It is the case of the Generalized Proportional Fair scheduling schemes, where different levels of fairness can be obtained when the weights of scheduling metric are chosen accordingly [1]. By imposing a static fairness constraints regardless to the channel conditions and traffic load leads to the unfair treatment of users that experience relatively the same channel conditions. Therefore, the fairness constraints should adapt at each TTI based on the scheduling performance in the previous TTI, traffic and network conditions. According to the Next Generation Mobile Networks (NGMN) specifications, a scheduler is considered to be fair if and only if each user achieves a certain percentage of their normalized user throughput [2]. At each TTI, the normalized user throughputs are evaluated and matched against the fairness constraint and the scheduling rule is adapted in a proper way. Therefore, finding the best scheduling metric that maximizes the system throughput and satisfies the dynamic fairness criteria depending on channel conditions and number of active users is considered to be a crucial task.

The current work focuses on the novel scheduling technique based on the reinforcement learning algorithms which is able to take scheduling decision at each TTI in order to meet the fairness requirement and to maximize the system throughput. In the first stage, sustainable scheduling policies are learned based on the interaction with the LTE scheduler. In the second step, the obtained metrics from the learned policy are applied at each TTI which makes the system suitable for the real time scheduling process.

2 Related Work

The scheduling adaptation methods proposed so far aim to meet different objectives in terms on the following metrics: the Jain Fairness Index (JFI) which is used to achieve a temporal fairness requirement imposed by the operator and the CDF curve in which the distribution of the normalized user throughputs should satisfy the NGMN fairness requirement. In [3] it is proposed an off-line procedure of adapting the parameter subject of different JFI constraints. The expected user throughput is calculated at the beginning of each TTI in order to predict the current state of the average user throughputs before the scheduling decision. But the traffic load is not considered and the method cannot be applied to the real systems due to the high complexity cost when the number of active flows increases. The parameterization of the PF scheduler for the system throughput maximization is intensively discussed in [4]. The impact of the traffic load and user rate constraints are considered when the CDF distribution is determined based on normalized user throughputs is determined. Unfortunately, the adaptation process is achieved at different time scale in order to make the proposal suitable for real time scheduling leading to the inflexible behavior when severe changes in the network conditions may occur. The balance of the system throughput and user fairness tradeoff is analyzed in [5], in which the traffic load is categorized based on the CQI reports. The normalized system throughput and JFI index are considered as a part of the input state. The Q-Learning algorithm is used to learn different policies that converge very well to different tradeoff levels. However, the concept is not extended to a dynamic adaption of the fairness requirement. In order to achieve a desire level of fairness at each TTI and to maximize the system throughput for a dynamic environment, five reinforcement learning algorithms are analyzed and compared with other existing methods. The policy which is proposed in [3] is adapted in order to achieve to achieve the CDF fairness requirement.

3 Proposed System Model

The proposed architecture indicates that the controller and LTE scheduler are two different entities that can operate in two modes: coupled and decoupled. In the first mode the interaction between the two is necessary in order to exchange the relevant information. This procedure includes two stages: exploration and exploitation. The decoupled mode is represented by the self-repetition procedure in order to provide a fine granularity of the output decisions for some RL algorithms. It is the case of the experience replay stage necessary to avoid the forgetting behavior of the neural network (NN) when the system spends too much time on a given part of the input state space [6]. Only the exploitation stage is used for the real time scheduling process whereas the other one are used as an offline processing. In the exploration stage the controller receives from the LTE scheduler a new input state at each TTI. Based on the trial and error principle, the controller takes random actions that are mapped into scheduling decisions by the scheduler. The scheduler ranks the previous scheduling decision at the beginning of the next TTI based on the reward function which is called the controller payoff. The objective of the exploration stage is to form a policy of scheduling decisions that follows those actions that maximize the sum of future rewards for every initial state. Due to the continuous nature of the input state space, the feed-forward backward propagation neural network is used as a function approximation. The feed-forward principle is used in order to approximate the accumulated reward value for the unvisited states pairs. The backward procedure trains the NN until the mean square error between the predicted and the new input states is minimized.

4 Performance Analysis

We consider a dynamic scenario with fluctuating traffic load within the interval of $[10, 120]$ active data flows/users. Moreover, the analyzed scheduling policies are running on parallel schedulers that use the same conditions for shadowing, path loss, multi-path loss and interference models. In order to test the impact of the proposed algorithms in the performance metrics, the number of active users is randomly switched at each 1s.

The RL algorithms are used in order to adapt and to apply the best fairness parameter for a dynamic radio environment in LTE-Advanced networks. We prove that three of these algorithms, such as SARSA, ACLA and CACLA, are able to outperform the other existing methods from the system throughput point of view by minimizing in the same time the resulted policy fluctuations when the number of active users changes dramatically. In the same time SARSA, ACLA and CACLA present the lowest percentage of TTIs when the system is unfair for the whole exploitation period when compared with other RL algorithms.

References

[1] F. Capozzi, G.Piro, L.A. Grieco, G.Boggia, and P.Camarda, "Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey," *IEEE Communications Surveys and Tutorials*, vol.15, no. 2, pp.678-700, 2013.

[2] R. Irmer, "Radio Access Peiformance Evaluation Methodology," in *Next Generation Mobile Networks*, 2008.

[3] S. Schwarz, C. Mehlführer, and M. Rupp, "Throughput Maximizing Multiuser Scheduling with Adjustable Fairness," *IEEE International Conference on Communications*, pp.1-5, 2010.

[4] M. Proebster, C. M. Mueller, and R. Bakker, "Adaptive Fairness Control for a Proportional Fair LTE Scheduler," *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PMIRC)*, pp.1504-1509, 2010.

[5] I.S. Comsa, S. Zhang, M. Aydin, P. Kuonen, and J. F. Wagen, “A Novel Dynamic Q-Learning-Based Scheduler Technique for LTE-Advanced Technologies Using Neural Networks,” *IEEE Conference on Local Computer Networks (LCN)*, pp.332-335, 2012.

[6] S. Adam, L. Busoniu, and R. Babuska, “Experience Replay for Real-Time Reinforcement Learning Control,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*,vol.42, no.2, 2012.

Modeling the Impact of Different Replication Schemes

Veronica Estrada Galiñanes
Université de Neuchâtel
veronica.estrada@unine.ch

Abstract

Storage solutions are commonly based on a multilayer software architecture. The strategy developed in each tier further distinguishes systems. The routing layer is responsible of maintaining routing tables and locate blocks. On top of that layer is built the reliability layer, which is in charge of all the magic related to construct a dependable storage system. This tier is considered the system's core since its design determines the durability model. At the core of all reliable strategies resides the redundancy method. Classical approaches like Reed-Solomon or n-way replication are inadequate to design a storage system that meets our high safety requirements. We explore the use of Helical Entanglement Codes (HEC), a coding technique to generate and propagate redundancy across the system. Our goal is to maximize the object's safety without decreasing performance as the demand for storage space continues to grow. We discuss the advantages of HEC repositories to assure legitimate endurable data storage. To evaluate our system, we compare HEC with traditional techniques and illustrate its strong impact on the object's durability.

Keywords: file durability; fault tolerance; dependability.

1 Introduction

The reliability layer is a key element to assure safe storage in an archival system. This study addresses a variety of aspects that impact on the object's safety. At the core of all strategies resides the redundancy method. Long-established methods like RAID5 only tolerates 1 failure. RAID6 provides double-failure support but has a scalability caveat. RAID array capacity is calculated using the amount of drives multiplied by the capacity of the smaller drive. Therefore such implementations waste capacity when new components are included in the system. Cloud storage solutions use n-way replication as de facto method to generate data redundancy. Recently, erasure codes become the standard answer to address the storage overhead problem inherent to replication. In particular, Reed-Solomon codes are implemented in [1, 2, 3] and Local Reconstruction Codes (LRC) are implemented in [4]. Erasure codes work efficiently in cold storage or in combined scenarios where data with high access frequency is replicated and distributed for load balance.

Previous approaches like Reed-Solomon or n-way replication are inadequate to design a storage system that meets our high safety requirements. Our work concentrates on the definition of a safety vault to assure legitimate endurable data storage. Our design is based on Helical Entanglement Codes (HEC) [5], a protocol to generate and propagate redundancy across the system. HEC generate redundant data that is soon propagated among other files in the system. Therefore, the system can profit from inbuilt redundancy. The strong dependencies among the files stored in the systems increase the system's resilience to failures. On top of that, a logical anti-tamper mechanism ensures that data store in the system was not modified.

We envision a trustworthy storage system that can be constructed through a mash-up of different subsystems. Such system can be used as a storage backend for generic cloud applications. Currently, cloud storage offers are saturating the market. The majority of providers offer an API for provisioning storage space in the cloud and compete using different pricing models. Newcomers offer high standard services at competitive prices; consequently, storage prices drop abruptly[6]. The user, however, might end trapped into one provider solution without being able to migrate data easily to other system. We think that it is the customer loyalty, but not the confined client, that helps to construct a sustainable business model with long-term agreements. Our ultimate goal is to build a dependable platform that leverages existent storage solutions to help the design and implementation of ad-hoc solutions.

2 Work Accomplished

2.1 Theoretical framework

During the development of any archival system many decisions are taken by designers while others are postpone and leave to the system's administrators. Such decisions are difficult to take without knowledge of system's behavior in production. We constructed a comprehensive comparison of noteworthy archival systems.

This survey may be helpful as a decision support framework for researchers as well as for storage providers. This work presents an assessment of the techniques used in storage system domain during the last 15 years. The assessment focus on dependable systems designed for long-persistence stored data. Systems are classified in two categories: p2p-based and client-server solutions. The first category, with the exception of Amazon Dynamo, is build on top of an untrusted infrastructure with a cooperative storage philosophy. In the second category, we grouped commercial-oriented solutions in which critical services reside in a protected trusted environment. There are some hybrid solutions like Tahoe and Wuala.

Comparing proprietary solutions with the systems previously proposed by the research community is difficult. Details of the system operation and assumptions used in the design are not published due to the sensitive business model information that storage providers try to protect. For those cases, we based on previous measurements obtained through reverse engineering. In particular, we examine the relationship between design choices and trade-offs that designers encountered during development. We have seen that there are no designs to satisfy all scenarios. Important challenges that remain in the domain are: data migration, easy tuneable parameters, user control and software able to adapt to different environments.

2.2 Practical framework

To evaluate the ability to recover data after a force majeure event, we conducted simple micro-benchmarks to compare HEC with traditional fault-tolerant techniques. To conduct the experiments, we use a file with the size of digital optical disk and split it in blocks of 1 MB. We create redundancy for this blocks using 5-HEC, Reed-Solomon (k=4,m=12) and 4-way replication. The parameters are chosen to keep the same storage overhead for all techniques. The process to destroy data choose data randomly and deletes different percentages of the data blocks stored in the system. In this process, there are no assumptions regarding the location of this blocks and/or domain failures. We measure the percentage of user data that could not be corrected after applying the repair process. For the traditional techniques the measurements include the percentage of data that is vulnerable, i.e. blocks that loss all its redundant blocks. Table 2 illustrates the strong impact of HEC in object's durability.

Table 2: Data recover ability of different fault-tolerant techniques

	Damage %					
	5	15	25	35	45	55
5-HEC						
Loss user data	0.00	0.00	0.00	0.00	0.00	2.41
4-way replication						
Loss user data	0.00	0.05	0.25	1.72	4.41	9.26
Vulnerable data	0.05	1.27	5.20	10.78	20.29	31.08
Reed-Solomon						
Loss user data	0.00	0.00	0.00	0.10	0.39	3.19
Vulnerable data	0.00	0.00	0.00	0.15	0.88	14.56

3 Work in Progress and Future Work

Configurations can have a big impact on the system reliability. A question that remains to be answered is how different methods impact on the durability of a file in a real-world application. For this study, we have chosen the exabyte-scale open-source storage system Ceph [3] to model reliability. The main reason to choose Ceph is its flexibility, i.e. it allows different system parameters and data redundancy strategies.

A Ceph storage cluster is build upon RADOS [7], a key service that offers reliability by using smartly the contribution of the individual components of the system. The flexibility of the system carries a complex setting. The operator may face a non-trivial task at the time of a cluster configuration to support different failure scenarios. Underlying disks or RAID, replication parameters, rebuild times, and other variables have different impacts in the object's durability. Our on-going work based on Ceph is the design and implementation of a reliability model for estimating the durability of an object stored in a cluster.

References

[1] Wilcox-O'Hearn, Zooko and Warner, Brian, "Tahoe: the least-authority filesystem," *Proceedings of the 4th ACM international workshop on Storage security and survivability*, pp. 21-26, 2008.

[2] LaCie AG, "Secure Cloud Storage,", accessed June 24, 2014, http://www.wuala.com

[3] Weil, Sage et al., "Ceph: a scalable, high-performance distributed file system," *Proceedings of the 7th symposium on Operating systems design and implementation*, USENIX Association, 2006.

[4] Huang, Cheng, et al. "Erasure Coding in Windows Azure Storage," *USENIX Annual Technical Conference* 2012.

[5] Estrada Galiñanes, Verónica and Felber, Pascal, "Helical Entanglement Codes: An Efficient Approach for Designing Robust Distributed Storage Systems," *Stabilization, Safety, and Security of Distributed Systems*, Springer International Publishing, 2013. 32-44.

[6] "Google cuts price of Drive cloud storage,", accessed June 24, 2014, http://www.gizmag.com/google-cuts-drive-cloud-storage-prices/31224/

[7] Weil, Sage A., et al. "Rados: a scalable, reliable storage service for petabyte-scale storage clusters." Proceedings of the 2nd international workshop on Petascale data storage: held in conjunction with Supercomputing'07. ACM, 2007.

Load Balancing in LTE Mobile Networks with Information Centric Networking

André Gomes
Universität Bern | One Source Lda.
gomes@iam.unibe.ch

Abstract

Mobile networks usage rapidly increased over the years, with a boom in terms of connected devices and used bandwidth to sustain access to heavy traffic content. Although technologies such as 3GPP Long Term Evolution brought major improvements and hence higher data rates, performance and resources utilization efficiency are still a challenge. At the same time, strategies such as Wi-Fi offloading cannot cope with those performance and efficiency requirements, bringing only a small improvement. In this extended abstract, mechanisms to use Information Centric Networking as an enabler and value-added technology to perform load balancing in mobile networks are presented. Taking advantage of the network agnostic implementation of Information Centric Networking, it is then possible to provide content delivery over multiple radio technologies with different radio stacks at the same time, and thus efficiently using resources and improving the overall performance of content transfer. With such novel approach, meaningful results were obtained by comparing content transfer over single links with typical strategies to content transfer over multiple links with Information Centric Networking load balancing. Information Centric Networking load balancing between radio technologies increases the performance and efficiency of 3GPP Long Term Evolution mobile networks and improves the Quality of Experience for end users.

Keywords: Information centric networking, load balancing, LTE.

1 Introduction

The evolution of mobile networks in the last few years has been quite intense, with major increase of throughput performance and radio resources efficiency. Such evolution is mostly driven by tremendous demand of bandwidth, with smartphones and other mobile devices playing a major role as content demanders. However, satisfying the content requirements of the current number of users while ensuring a good Quality of Experience (QoE) for everyone is still a challenge.

With such a challenge in mind, new concepts for improvements to the performance and efficiency of 3GPP Long Term Evolution (LTE) networks have emerged. The first one is Information Centric Networking (ICN) [1], which proposes a change in the current paradigm of requesting content. While currently, when a user wants to request a content object, it has to query a specific server for very specific content, the same is not needed with ICN. With ICN, a user sends an Interest message that is then routed by ICN routers, which have forwarding tables based on content naming. These routers will find the content object (if available) and return it directly to the user, without any further interaction. This approach has multiple advantages over the traditional paradigm, as for example: the user does not need to search for the content object (and hence saves bandwidth and time), the content object can be delivered using the best route available (better efficiency and quicker delivery), the content object can be migrated as the user moves to another location (mobility support) and finally caching mechanisms can be used to save bandwidth and improve QoE. The second concept is the concept of Cloud Radio Access Network (C-RAN) [2], which brings the possibility to virtualize the entire radio infrastructure besides the antennas. Using virtualized infrastructures extends the cloud computing concept to the Radio Access Network (RAN), and explores the modularity of the components together with the usage of general-purpose hardware infrastructure to run the evolved Node Bs (eNBs) of 3GPP LTE mobile networks. Such fact is an enabler for the co-location of ICN routers in proximity to (and integrated with) eNBs of LTE mobile networks. Co-location and integration of ICN routers can then support a new set of benefits. For instance, caches become much closer to the users requesting content, and that saves time needed to access cached content is lower (better performance) while saving network resources at the core (less capacity needed). Also, a particular benefit with a very promising outcome may be explored – seamless load balancing between radio technologies. This benefit can be explored due to ICN's inherent characteristics, which overcome the big challenge of having load balancing between a number of different network stacks (one per radio technology).

Current proposals do not address all the relevant aspects for load balancing between radio technologies, either using ICN or other protocols. Proposals such as the one from Detti et al. [3] define strategies for using radio technologies in a cooperative way, avoiding the usage of LTE in some cases but with tremendous scalability and performance issues. Others, such as Montpetit et al. [4], simply rely on Network Coding (NC) to achieve the objective indirectly. In this case, load balancing can be obtained in some cases but it will not adapt to different link states and will create additional overhead.

Our proposal, described in the next section, describes mechanisms to handle load balancing between radio technologies using ICN and tackling the issues left by related proposals.

2 ICN Load Balancing

The proposal described hereafter intends to address mainly the combination of two interfaces (links), the ratio to use when splitting content Interest messages (and corresponding segments) among those interfaces and how to make a well-known ICN implementation compatible with such solutions. To serve as base for this work, CCNx[5] was chosen as the implementation of ICN because it is a reference in the research community and its code is open source.

By default, CCNx already handles the process of sending Interest messages over multiple interfaces (faces), with the possibility to use a number of different forwarding strategies. With our load balancing strategy, multiple faces for the same content prefix will be used when sending Interests messages that match that content prefix, which means multiple faces will be simultaneously used to deliver the segments that are part of the content object.

Although this strategy enables the content object segments to be delivered over two interfaces and in fact performs load balancing by aggregating the links, it is still not efficient and reliable because of the different conditions each link may present. For instance, the LTE link may have a poor quality and that will be eventually worse than not doing load balancing at all. Therefore the amount of data to be sent over it should be less than the amount sent over a good quality Wi-Fi link. To decide on this ratio, a simple score based formula was developed:

$S = \frac{R_{mcs}}{R_{max}} + (1 - \frac{L_i}{L_{max}}) + (1 - E)$

In the formula, we may consider S as the score the link has, ranging from 0 to 3. To calculate it, three factors are considered: data rate, latency and transmission errors.

R_{mcs} is the peak data rate obtained using the Modulation and Coding Scheme (MCS), the channel bandwidth and the Multiple Input Multiple Output (MIMO) setting for the specific radio technology.

R_{max} the maximum data rate for the radio technology (maximum MCS).

L_i is the latency obtained using Internet Control Message Protocol (ICMP) ping at a given moment.

L_{max} is the maximum latency to be considered, which can be set to a high value such as the one for the timeout of the Interest messages.

E is the rate of packet transmission errors obtained from the interface statistics, which is represented as a value between 0 and 1.

With this formula, every link gets a score that can be used to calculate the ratio of Interest messages to be sent over each link. Getting back to our example, the ratio to split the Interest messages over LTE ($Ratio_{LTE}$) and Wi-Fi ($Ratio_{WiFi}$) would be calculated by:

$Ratio_{LTE} = \frac{S_{LTE}}{S_{LTE}+S_{WiFi}}$ $\quad$ $Ratio_{WiFi} = \frac{S_{WIFI}}{S_{WIFI}+S_{LTE}}$

These ratios should be negotiated between the core network and the end user equipment, together with policies to decide on whether load balancing can be used or another single technology is preferred.

3 Evaluation and Conclusions

To test the feasibility and the performance of the proposed solution, a test scenario was defined. The architecture for the evaluation of the proposed system has the main purpose of replicating a possible integration of CCNx routers with the LTE Evolved Packet System (EPS) architecture. Hence, five users were considered with user devices attached to two different CCNx routers, one representing a CCNx router attached to 3GPP LTE eNBs and another close to an ePDG for Wi-Fi integration. These five users are each connected with up to two interfaces, either using 3GPP LTE or using Wi-Fi. However, these links are shaped to assume particular values of bandwidth (Mbps), delay (ms) and the rate of packet transmission errors (%).

To test different shaping scenarios, particular values were chosen for both the links. With these values, three different shaping scenarios were created and will be described in Table 3:

Table 3: Shaping Scenarios

Scenario	Technology	Bandwidth (Mbps)	Latency (ms)	Errors (%)
Proximity	LTE	50	10 ± 5	0
	Wi-Fi	100	5 ± 2	0
Urban Area	LTE	25	20 ± 5	1
	Wi-Fi	50	10 ± 2	0.5
Rural Area	LTE	10	30 ± 5	2
	Wi-Fi	20	15 ± 2	1

The five users are distributed as: 2 devices with Proximity shaping scenario, 2 devices with Urban Area shaping scenario and 1 device with Rural Area shaping scenario. Considering different strategies, multiple tests were conducted for a period of one hour. The strategies being considered are: LTE only, partial Wi-Fi offloading (only if rates are much higher) and Load Balancing. During each of the tests, the requested content objects were the same for the particular content sizes defined (10 MB, 20 MB, 50 MB and 100 MB). As CCNx caching would influence the results and is not evaluated in this work, it was disabled. Each device requested a content object of a given size (decided randomly) at a random interval between 5 to 30 seconds. To measure performance, the time for the user to get the content was selected as the metric. Also, usage of LTE resources is evaluated to minimize the costs for the mobile network operator and reduce the overall cellular network load.

Results showed us that a noticeable difference exists when comparing the different strategies. In fact, due to the average higher data rates of Wi-Fi, offloading users and performing load balancing can easily introduce a quite high performance gain. The performance gain of the load balancing strategy was of about 48% in terms of content object download time on average, and it reaches a performance closer to what would be the theoretic optimal strategy. The Wi-Fi offloading strategy is on average 25% better than using LTE only in the evaluated scenario, which is still a good improvement over the traditional setup but not efficient enough to improve the overall network performance. At the same time, and as expected, a slight increase in performance is present when the content object size increases due to less impact of the protocol overhead.

As for the comparison of LTE overall usage for the ICN load balancing and Wi-Fi offloading strategies, it is based on the percentage of LTE bandwidth (from the total bandwidth required by the devices) required by the connected devices. With the Wi-Fi offloading strategy, usage of LTE resources was still high at around 50%. These results are explained by the need of having devices that are good candidates for offloading (usually when a noticeable improvement in terms of data rate exists). The same does not apply for the load balancing strategy though, as it is possible to have partial offloads (depending on the calculated ratio) and hence increase the efficiency of resources usage. Indeed, the usage of LTE resources in this case was around 17% and thus the bandwidth savings for the operator were much higher than with typical Wi-Fi offloading.

In the end, this evaluation shows that performance gains are high and resources are used more efficiently from the network point of view when ICN load balancing is used. It is then clear that such strategy could be beneficial for all the stakeholders involved, from the mobile network operators to the end users and also considering content providers, which aim at delivering high quality content without major constraints. In future work, our strategy will also be evaluated taking advantage of caching mechanisms and deeper integration into LTE mobile networks.

References

[1] G. Pavlou, "Keynote 2: Information-centric networking: Overview, current state and key challenges," in Computers and Communications (ISCC), 2011 IEEE Symposium on, 2011, pp. 1–1.

[2] B. Haberland, F. Derakhshan, H. Grob-Lipski, R. Klotsche, W. Rehm, P. Schefczik, and M. Soellner, "Radio base stations in the cloud," Bell Labs Technical Journal, vol. 18, no. 1, pp. 129–152, 2013.

[3] A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano, and A. Bragagnini, "Offloading cellular networks with information-centric networking: The case of video streaming," in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, June 2012, pp. 1–3.

[4] M.-J. Montpetit, C. Westphal, and D. Trossen, "Network coding meets information-centric networking: An architectural case for information dispersion through native network coding," in Proceedings of the 1st ACM Workshop on Emerging Name-Oriented

Mobile Networking Design - Architecture, Algorithms, and Applications, ser. NoM '12. New York, NY, USA: ACM, 2012, pp. 31–36.

[5] P. A. R. C. Inc. (2014, June) Project CCNx. [Online]. Available: http://www.ccnx.org/

Building a Multi-site Consistent File System

Raluca Halalai
Université de Neuchâtel
raluca.halalai@unine.ch

Abstract

A distributed file system provides transparent access to files stored at multiple remote sites. Existing geographically distributed file systems typically provide weak consistency guarantees in order to bypass the limitations stated by the CAP theorem - a distributed system can simultaneously support at most two of the following three guarantees: consistency, availability, and partition tolerance. However, weak consistency is only suitable for domain-specific applications, in which programmers are able to anticipate potential conflicts and provide suitable resolution mechanisms. We argue that for general-purpose services such as a file system, strong consistency is more appropriate, as it is more intuitive for its users and does not require human intervention in case of conflicts.

Our goal is to build a geographically distributed file system that guarantees strong consistency in the presence of failures; this implies potentially sacrificing availability in the case of multi-site operations. Our file system prototype is built on top of two abstractions. First, it relies on a set of single-site linearizable storage sites; these are organized as distributed hash tables, in which data blocks are immutable and replicated. Second, our file system prototype uses an atomic multicast abstraction based on the Multi-Ring Paxos protocol to maintain mutable file metadata and orchestrate multi-site operations. Multi-Ring Paxos provides strong order guarantees both within and across data centers.

Keywords: multi-site, file system, strong consistency, paxos

1 Introduction

With the advent of data sharing and backup services (e.g., Dropbox, Google Cloud Storage, Amazon S3/Glacier), building large-scale distributed data stores has gained a lot of attention from the research community. The goal of our work is to build a POSIX-compliant multi-site distributed file sytem.

The CAP theorem [1] states that a distributed service can simultaneously provide at most two out of the following three desireable properties: consistency, availability, and partition tolerance. To surpass the limitation formalized by the CAP theorem, existing distributed file systems, like Ivy [3] and Oceanstore [2], provide weak consistency guarantees .

On the other hand, our goal is to build a multi-site file system that ensures strong consistency despite node failures, at the price of possibly experiencing reduced availability for multi-site operations. We argue that weak consistency can be suitable for domain-specific applications, in the case of which programmers are able to anticipate and provide resolution methods for any possible conflict. However, for general-purpose services such as a file system, strong consistency is more appropriate as it is both more intuitive for its users and in particular does not require human intervention in case of possible conflicts.

2 Work accomplished

2.1 Design

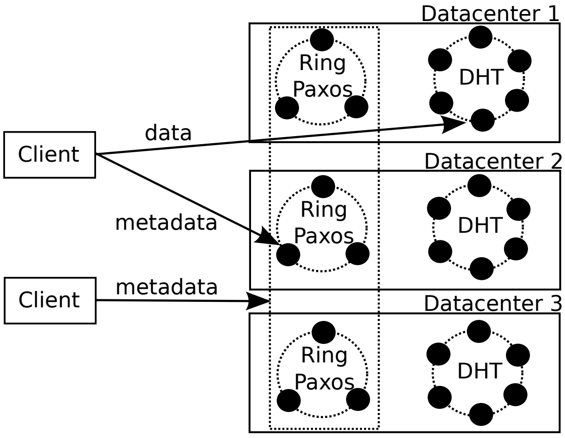


Figure 7: Multi-site file system design

We designed our multi-site file system on top of two abstractions, as presented in Figure 7. First, our file system relies on a set of *single-site* linearizable data stores, located in geographically distributed data centers. These data stores are organized as distributed hash tables (DHTs) and files are replicated and stored as immutable blocks on several storage nodes within each data store. Second, our file system uses an atomic multicast abstraction based on the Multi-Ring Paxos [4] protocol to maintain mutable file metadata and orchestrate *multi-site* operations. Multi-Ring Paxos provides strong order guarantees both within and across data centers.

We exploit the multi-site nature of our file system in order to improve its performance by paying the price of consensus and strong consistency only when necessary. Our file system provides four execution modes corresponding to the operations that can be performed on the file system: (1) single-site operations, (2) multi-site uncoordinated operations, (3) multi-site coordinated operations, and (4) read-only operations. While the first and the latter can be implemented efficiently by accessing a single site, the other two operation types require accessing multiple sites and are ordered by Multi-Ring Paxos.

2.2 Prototype

Data	Paxos	DHT	#Clients	Proxy	Throughput
10 MB	No	Yes	1	random	6.3 MBps
10 MB	No	Yes	10	same	33.3 MBps
10 MB	No	Yes	10	random	34.6 MBps
10 MB	Yes	Yes	1	same	460 KBps
10 MB	Yes	No	1	same	587 KBps

Table 4: Preliminary performance results

We have implemented a complete prototype of the proposed multi-site file system and deployed it on our local cluster and on Amazon EC2. Table 4 shows the results of some preliminary experiments that we have performed on our file system prototype. In all experiments, we write a 10 MB file through our file system. We compare the throughput obtained through our strongly consistent file system to just the data store alone, without consistency guarantees. Our system works as expected, maintaining strong consistency. However, the current performance is very poor, so we need to focus on optimizations.

3 Work in progress and future work

We are currently working on tweaking the performance of our file system and on devising an experimental evaluation plan. Our aim is to study the impact of the strong consistency guarantee on the performance of the system.

References

[1] Eric A. Brewer, “Towards Robust Distributed Systems“, *PODC keynote*, 2000.

[2] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishan Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, Ben Zhao, “OceanStore: An Architecture for Global-scale Persistent Storage“,*ACM SIGPLAN Not.*, vol. 35, no. 11, pp.190-201, 2000.

[3] Athicha Muthitacharoen, Robert Morris, Thomer M. Gil, Benjie Chen, “Ivy: A Read-/Write Peer-to-peer File System“, *ACM SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 31-44, 2002.

[4] Parisa J. Marandi, Marco Primi, Fernando Pedone, “Multi-Ring Paxos“, *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2012.

Multi-core Programming with Message Passing and Transactional Memory in the Actor Model

Yaroslav Hayduk
Université de Neuchâtel
yaroslav.hayduk@unine.ch

Abstract

The actor model has been successfully used for scalable computing in distributed systems. Actors are objects that hold local state that can only be modified by the exchange of messages. An Actor has access to its local memory space only, its state is isolated, and the access to another Actor's memory space is performed using message passing. We have shown in former work that concurrent message processing can be implemented with the help of transactional memory, ensuring sequential processing, when required. This approach is advantageous in low contention phases, however, does not scale for high contention phases. We introduce two mechanisms to overcome this limitation. We dynamically adapt the number of threads to the workload. Second, we extract read-only messages from the transactional context and treat them separately.

Keywords: software transactional memory; actor model; concurrency.

1 Introduction

Current scaling trends at the CPU level let us expect increasing core counts in the following years. This causes, however, limitations of performance/energy gains as it is difficult to program parallel applications efficiently. The current structures used to implement shared memory do not scale well and might need to be abandoned. The actor model, initially proposed by Hewitt [1], is a successful message-passing approach that has been integrated into popular concurrency frameworks. An actor is an independent, asynchronous object with an encapsulated state that can only be modified locally based on the exchange of messages. While the data consistency property of the actor model is important for preserving application safety, it is arguably too conservative in concurrent settings as it enforces sequential processing of messages, which limits throughput and hence scalability.

In previous work [2], we addressed this limitation by proposing a mechanism to boost the performance of the actor model while being faithful to its semantics. Our key idea was to apply speculation, as provided by transactional memory (TM), to handle messages concurrently as if they were processed sequentially. However, we noticed that in cases of high contention, the performance of parallel processing dropped close to or even below the performance of sequential processing. We propose a combination of two approaches to reduce the contention: (1) determining the optimal number of threads that execute transactional operations and (2) relaxing the atomicity and isolation for some read-only operations.

Didona et al. [3] argue that the performance of an application is dependent on the level of concurrency, and propose to dynamically determine the optimal number of threads depending on the workload. We also start from the maximum number of possible threads in the thread pool and then the decision phase is driven by a fixed threshold α dependent on a commit-to-rollback ratio. If the current commit-to-rollback ratio is lower than the predefined threshold α, we divide the number of threads processing STM messages by two; if it is higher, we multiply them by two. After, the rest of the available threads are assigned for processing of read-only messages.

Much of the contention in our tests was caused by read-only operations on TM objects causing conflicts. Rollbacks could be avoided by relaxing the semantics of read operations, sometimes yielding inconsistent values. Scala STM provides an *unrecorded read* facility, in which the transactional read does not create an entry in the read set but bundles all meta-data in an object. At commit the automatic validity check is omitted, but may be done by the caller manually. However, if there are concurrent writes on the same value, the unrecorded read might cause contention. If we limit the reads on single values only, we can omit the unrecorded read and grant direct access, using the *volatile* keyword. This ensures that we read the latest value, although relaxing atomicity. By using direct access for message processing within the actor model, we can process a large amount of single read-only messages, while not conflicting with messages processed in regular transactions (TM messages).

The read-only and the TM messages may require different levels of concurrency. Following this observation, during high contention phases, we reduce the number of threads processing regular TM messages, which in turn allows us to increase the number of threads processing read-only messages. By handling these two message types differently (i.e., providing a separate queue for each of the types) we can optimally use the available resources.

2 Work accomplished

To test our ideas we consider an artificial list benchmark and second on a real-world scientific application. We consider two scenarios for the level of concurrency. First, we specify a static ratio of threads assigned to process STM messages and read-only messages. 90 % of threads are assigned to process STM messages and the rest of the threads process read-only messages. Second, we consider a dynamic ratio. Here, we are able to assign unused resources (threads that would usually work on STM message processing) to the processing of read-only messages. To avoid starvation, we never process less than 10 % of messages of one type. We execute the benchmarks on a 48-core machine equipped with four 12-core AMD Opteron 6172 CPUs running at 2.1GHz.

Our optimizations are expected to be most useful in applications where state is shared among many actors. Hence, to evaluate our approach, we use a benchmark application provided by Imam and Sarkar [4] that implements a stateful distributed sorted integer linked-list. Sequential data structures are typical benchmarks for relaxed atomicity and isolation.

The architecture considers two actor types: *request* and *list* actors. Request actors send requests such as *lookup*, *insert*, *remove*, and *sum*. List actors are responsible for handling a range of values (buckets) of a distributed linked list. We implemented a list element as an object containing a *value* field and a *next* field, which is wrapped in a Scala STM Ref object.

In a list with l actors, where each actor stores at most n elements representing consecutive integer values, the i^{th} list actor is responsible for elements in the $[(i-1)\cdot n, (i \cdot n)-1]$ range. A request forwarder matches the responsible list actors with the incoming requests. For the *sum* operation, we traverse each list element in every list actor. This operation is read-only and does not necessarily report a consistent value. It should not conflict with other accesses to the list.

We run the benchmark with 2-32 threads (multiplying by 2) and average the results of 7 runs. Also, we set the maximal number of list elements to 41,216 and create 8 list actors. We create 500 request actors, where each actor sends 1,000 messages to the list.

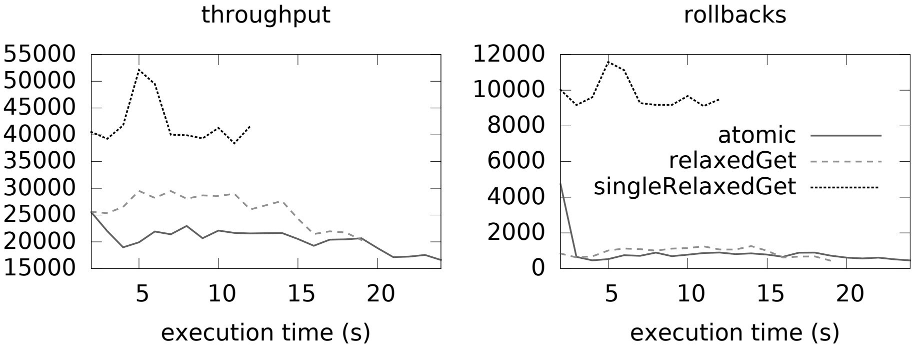


Figure 8: List write-dominated workload: static thread allocation.

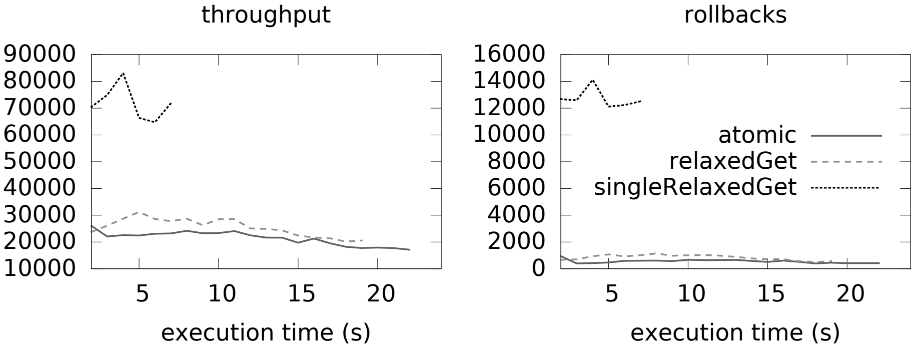


Figure 9: List write-dominated workload: dynamic thread allocation.

To read a value from the list, we consider three different options: (1) the regular transactional read (*node.next.get()*), (2) the unrecorded read accessed via (*node.next.relaxedGet()*) and (3) our direct access (*node.next.singleRelaxedGet()*) method, which we added to the STM Ref object.

In our experiments we consider a write-dominated workload. The write-dominated workload is configured as follows: Each request actor sends 98 % of requests to modify the list (insert or remove), 1 % of lookup requests and 1 % of sum requests.

In the first experiments, we evaluate the impact of different access approaches to the sum operation if the threads are allocated statically. The static approach (90:10) reserves 3 threads (10 %) for the processing of read-only messages and 29 threads (90 %) for processing STM messages. Figure 8 demonstrates the message throughput and the rollback count over time for different types of sum operations. Clearly, the *singleRelaxedGet()* outperforms other operations with respect to both, execution time and throughput. Perhaps surprisingly, we observe a drastic increase of rollbacks. This observation is counter intuitive, as one would expect to have a lower number of rollbacks to achieve higher throughput. It turns out that when we use *get()* and *relaxedGet()* to implement the sum operation, we cause significantly more read-write conflicts, deteriorating performance. Scala STM resolves them by waiting out the write operations, and only then allowing the read operations to take place. On the contrary, when we use the *singleRelaxedGet()* operation, it does not perform a transactional read, implicitly causing an increase of concurrent transactional write executions. As a result, we get more write-write conflicts, which are typically resolved eagerly by rolling back one of the transactions.

Since the performance of transactional operations can be further improved, we dynamically determine the optimal number of threads. For this experiment we set $\alpha = 0.21$. In Figure 9 we see that the throughput and rollback values for the *get()* and *relaxedGet()* operations are not significantly different when compared to the static approach. This behavior is expected as the operations interfere with the concurrently executing transactional read-modify list operations. On the contrary, the *singleRelaxedGet()* operation never conflicts with other list operations.

3 Work in progress and future work

Recall that the value of α was set empirically for each of the scenarios we evaluate. In future work, a combination with hill climbing or gradient descent could be used to dynamically determine this value.

Besides improving dynamic thread allocation, we are working on a novel way of integrating graphical processing units (GPUs) with the actor model. In the baseline actor model actors are expected to run on CPUs, which are optimized for low-latency access to cached data sets but can handle running only a few hardware threads at a time. The GPU, on the other hand, supports the execution of millions of hardware threads and is highly optimized for data throughput. By matching the Actor workload to the most appropriate architecture, we foresee obtaining significant performance benefits.

One of the strategies of enabling the use of GPUs with the Actor Model is to enforce programmers to provide a companion version of their actor code to be executed on the GPU. The code would need to be written in either CUDA or OpenCL using C/C++. While this approach is simple, it might be problematic to link it with the build system of C/C++, since the GPU code would have to be compiled on the fly during runtime. The second approach of integrating GPU code with the Actor Model is to use domain-specific languages (DSLs) for code generation. As an example, the programmer would develop his application using a provided DSL. Then, at compile time, the system would generate two versions of code, one for the GPU and one for the CPU.

Lastly, in our previous research we successfully used TM for optimizing message processing in the actor model [2]. As the use of TM was beneficial in the context of Actor execution on the CPU, we could investigate how to employ transactional memory for GPUs [5] in the context of the actor model.

References

[1] C. Hewitt, P. Bishop, and R. Steiger, “A universal modular actor formalism for artificial intelligence,” in International Joint Conference on Artificial Intelligence (IJCAI), pp. 235–245, 1973.

[2] Y. Hayduk, A. Sobe, D. Harmanci, P. Marlier, and P. Felber, “Speculative concurrent processing with transactional memory in the actor model,” in International Conference on Principles of Distributed Systems (OPODIS), 2013.

[3] D. Didona, P. Felber, D. Harmanci, P. Romano, and J. Schenker, “Identifying the optimal level of parallelism in transactional memory systems,” in International Conference on Networked Systems (NETYS), pp. 233–247, 2013.

[4] S. M. Imam and V. Sarkar, “Integrating task parallelism with actors,” in Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications, (OOPSLA), pp. 753–772, 2012.

[5] W. W. L. Fung, I. Singh, A. Brownsword, and T. M. Aamodt, “Kilo TM: Hardware Transactional Memory for GPU architectures.,” IEEE Micro, vol. 32, no. 3, pp. 7–16, 2012.

Process-level Power Estimation in Multi-application VM-based Systems

Mascha Kurpicz
Université de Neuchâtel
mascha.kurpicz@unine.ch

Abstract

State-of-the-art power estimation solutions provide coarse-grained support for power estimation in virtualized environments, typically treating virtual machines (VMs) as a black box. Yet, VM-based systems are nowadays commonly used to host multiple applications for cost savings and better use of energy by sharing common resources and assets. To consider such multi-application virtualized environments, we propose a process-level power estimation architecture that exploits the knowledge from the host as well as the knowledge of a process within the VM.

Keywords: power consumption; virtual machines; real-time monitoring.

1 Introduction

When considering the power consumption of cloud computing and comparing it to countries, cloud computing consumed more power than India or Germany in 2007[1]. Hence, there is a great interest in reducing the power consumption in cloud computing. Before solving the problem of power reduction, we need to be able to evaluate current installations in terms of power. In particular, we want to estimate power consumption of a process running as part of multi-application VMs. We consider VMs containing multiple applications, or multiple modules of an application, such as seen in business applications like supply-chain-management utilities.

Existing work in the field can be split into 3 categories:

- Physical power meter
- Power estimation based on hardware
- Power estimation for virtualized environments

Physical power meters as for example Powerspy[2] monitor the entire system and do not provide process-level power values. Another way to make power estimations is to create a model based on the information provided by the hardware. This approach works fine for power estimation on the host, but not in the case of a virtualized environment. From inside the VM, it might not be possible to access the required hardware resources. For example, PowerAPI[1] is based on cpufrequtils[3] which is not available in a VM. Specialized approaches for virtualized environments often consider the virtual machine as a blackbox[2][3], i.e., they estimate the power consumption for an entire VM. We want to identify the power consumption of a single process running in a VM, and propose a tool that forwards the host information to the VM.

[1] Greenpeace's Make IT Green Report, 2010.

[2] http://www.alciom.com/fr/produits/powerspy2.html

[3] http://dev.man-online.org/package/main/cpufrequtils

2 Work accomplished

In our work we provide a process-level power estimation tool for multi-application VMs. Based on a hardware model for each host architecture, the host estimates the power consumption for an entire virtual machine. This power estimation is transmitted to the virtual machine using an efficient VirtioSerial channel. Based on the CPU utilization of a process running in the VM and the power value received from the host, our tool estimates the power consumption of a process within the VM. Figure 10 illustrates the communication of the power values from the host to the VM.

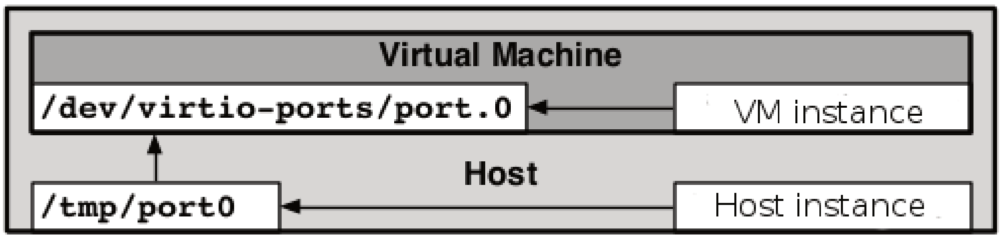


Figure 10: The communication from the host to the VM.

To estimate the power consumption of an application running in the VM $Power_{vm}(app)$, we need to know the consumption of the entire VM $P(vm)$. Additionally, we need to know the CPU utilization of the application $U_{vm}(app)$ and of the other applications running in the VM $U_{vm}(total)$.

$$Power_{vm}(app) = Power_{host}(vm) \cdot \frac{U_{vm}(app)}{U_{vm}(total)}$$

3 Work in progress and future work

We are evaluating our solution with benchmarks and real-world applications. Power estimation of different types of applications shows that the estimation from the host requires corrective measures from within the VM. We noticed that some improvements are possible by using only the CPU utilization of a given process within the VM. More sophisticated methods might be necessary to improve the accuracy of the proposed power estimation.

Most of our experiments were done with Intel CPUs. Since our approach is independent from hardware, we want to get more experiences with other architectures as for example AMD.

Currently, we only consider CPU-intense workloads. Many cloud applications are, however, data intense with disk access. This might change the power consumption model drastically and hence requires some investigation.

References

[1] A. Noureddine, A. Bourdon, R. Rouvoy, and L. Seinturier, “A Preliminary Study of the Impact of Software Engineering on GreenIT,” in *First International Workshop on Green and Sustainable Software (Zurich)*, pp. 21-27, 2012.

[2] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proceedings of the 1st ACM symposium on Cloud computing (ACM)*, pp. 39–50, 2010.

[3] A. Bohra, and V. Chaudhary, "Vmeter: Power modeling for virtualized clouds" in *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW)*, pp. 1-8, 2010.

Cloud-Based Architecture for Environmental Modeling

Andrei Lapin
Université de Neuchâtel
andrei.lapin@unine.ch

Abstract

This document provides an architecture for real time environmental modeling. It consists of a wireless mesh network equipped with sensors for real-time data harvesting and a cloud based infrastructure to perform environmental modeling using real-time data flows. Our initial studies prove that the cloud infrastructure can simultaneously compute a large number of environmental modelers thus allowing for the implementation of environmental Ensemble Kalman Filters in real time.

Keywords: wireless mesh network; cloud computing; environmental modeling.

1 Introduction

Simulating and predicting the behavior of environmental systems is a complex task. Systems that allow us to quantify interactions have to be deployed in order to i) capture data, ii) simulate the system state, and iii) provide the predictions on the behavior of the real system in future. Regions concerned of high interest for environmental scientists are situated in remote locations, therefore capturing system dynamics requires the installation of real-time measurement infrastructures connected to the Internet which provide remote access to harvested data. There exist systems which tackle surface-water, while some of them use Kalman Filter techniques for forecasting. However, modeling a highly dynamic hydrological system of connected surface-water and ground-water profiles is much more computationally demanding, because it requires computing a numerical model that catches the physical processes between these two water profiles. Derived models can also be used to predict the behavior of the system in future. Moreover, a mechanism that automatically, and continuously, assimilates newly available data, updates the model in real-time, and provides predictions based on the best possible description of the current state is required. As such a high number of models have to be computed, such modeling system is highly suited for parallelization.

2 General system architecture

Figure 11 presents an architectural view of our system. It consists of two main parts: i) a Wireless Sensor Network for environmental monitoring and ii) a cloud-based computational service for real-time environmental modeling. Real-time modeling requires a constant flow of data between sensor nodes located somewhere in the field and data-centers. A classical wireless mesh network equipped with sensors is beneficial for environmental researchers, because it is not restricted to the reception of mobile telephony such as GSM, operates over long distances and extremely portable. Also, wireless communication is significantly less expensive than any wired one in terms of installation costs. Our studies reveal great similarities between environmental and system or network monitoring provided by Zabbix, which track the status of system or network components. Monitoring is implemented as a periodical query for the status of a certain unit. Environmental monitoring is similar, because it also requires periodical information about the environmental system state. Generally, running environmental simulations is computationally expensive, but new trends in real-time modeling such as the Ensemble Kalman Filter amplify workload even further by simultaneously running a large number of various environmental models. The models deviate from a real system in time, the simulation requires adjusting predicted system states and recomputing deviating models. This is a huge effort, which requires enormous computational power. We decided to migrate environmental modeling into the cloud environment. In the cloud, the end-user can run a large number of parallel models whereas one model occupies a single working machine (VM). This allows us to compute many models in a reasonable time and use them to provide system state predictions of good quality. In the cloud, we have deployed a leader machine, i.e., Manager, responsible for orchestrating all other elements. The Manager spawns working VMs for newly launched computing instances and provides monitoring functions among VMs. The end-user only has a top-level view of the whole system and communicates with the Manager through the web interface. This general architecture allows us to run generic environmental applications such as the Ensemble Kalman Filter in which workload can be easily distributed among several parallel working instances.

3 Work accomplished

Firstly, we have designed a modular architecture that provides a quick integration of an easily parallelizable environmental computing application with cloud computing environment. Secondly, we have showed that the HydroGeoSphere application which is extensively used in the domain of hydro-geology can be successfully run in distributed way to the benefit of the HydroGeo community. The cloud proves that it can quickly distribute load of many simultaneously computed models, thus it allows for computationally expensive modeling.

4 Work in progress and future work

As a future work, we shall implement on-the-fly integration of dynamic data sources and the environmental Ensemble Kalman Filter (EnKF) in the cloud environment. In such a solution, the main interest is the full integration of cloud infrastructure services (both computing and storage) for a high-throughput hydrological application with the objective of allowing for near real-time processing of sensor data.

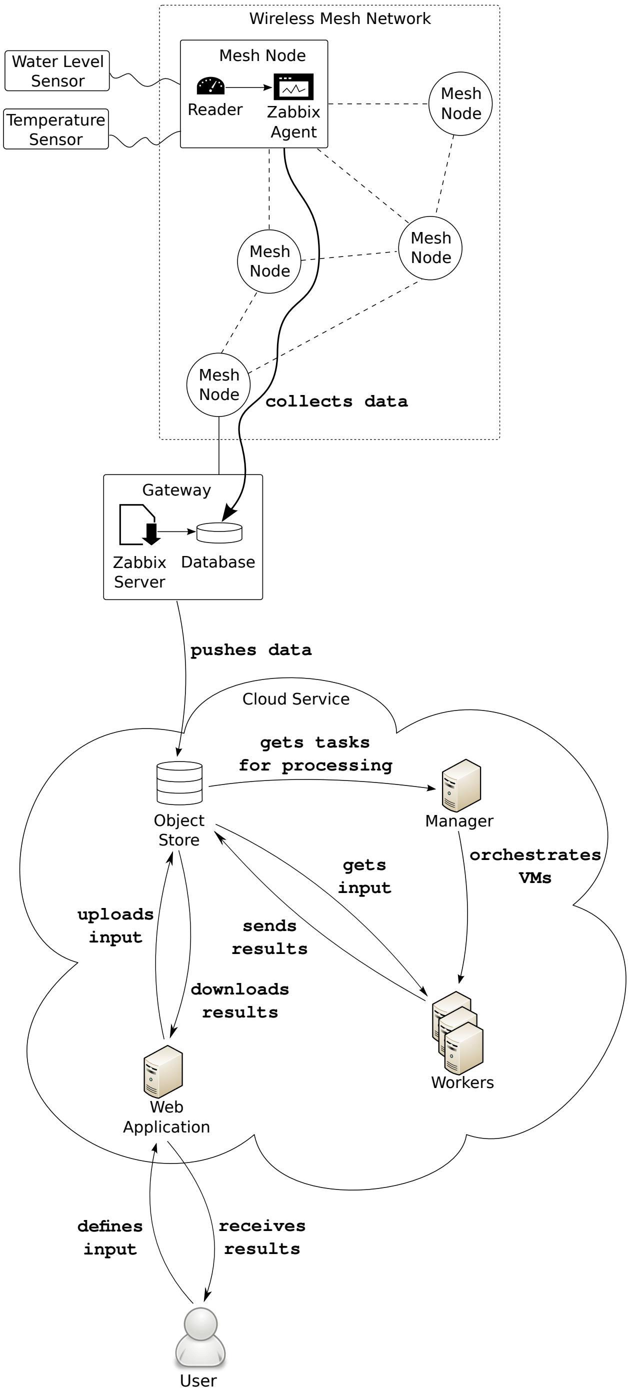


Figure 11: General Architecture

Time-Based Indoor Localization for Narrow-Band System

Zan Li
University of Bern
li@iam.unibe.ch

Abstract

Time-based localization is proposed to achieve sub-meter localization accuracy for wide-band signals. However, for narrow-band signal systems, e.g., GSM and Zigbee, time-based localization is challenged by accurate synchronization, high resolution timestamp and multipath propagation. In our work, we propose three approaches, i.e., DTDOA, sub-sample timestamps, and fingerprinting, to deal with these challenges. We evaluate our proposed methods in a software defined radio based testbed for IEEE 802.15.4 signal. Our results show that time-based fingerprinting is a promising alternative to power-based fingerprinting.

Keywords: Time-based localization; Synchronization; Fingerprinting.

1 Introduction

Time-based localization techniques such as multilateration are favored for positioning to wide-band signals. Applying the same techniques with narrow-band signals such as GSM and Zigbee is not so trivial. The process is challenged by the needs of synchronization accuracy and timestamp resolution both in the nanoseconds range. We propose approaches to deal with both challenges. On the one hand, we introduce a method to eliminate the negative effect of synchronization offset on time measurements. On the other hand, we propose timestamps with nanoseconds accuracy by using timing information from the signal processing chain. Furthermore, the accuracy of timestamps is limited by multipath propagation. Fingerprinting is among the most successful approaches used for indoor localization to deal with multipath propagation. It typically relies on the collection of measurements on Radio Signal Strength (RSS) over the area of interest. We propose an alternative by constructing fingerprints of fine-grained time information of the radio signal. We offer a comprehensive analytic discussion on the feasibility of the approach, which is backed up with evaluations in an IEEE 802.15.4 (Zigbee) testbed. The testbed is implemented using software defined radio systems to collect the fine-grained time information of the Zigbee signal at the physical layer.

2 Time-based Localization

2.1 Synchronization Solution: DTDOA

Due to the high propagation speed of radio signals, time-based localization requires strict synchronization between the Anchor Nodes (ANs) to achieve high accuracy. In order to evaluate GPS synchronization in practice we conducted measurements on the GPS synchronization of ANs and compared them against the requirements of Time Difference Of Arrival (TDOA) [1]. In order to eliminate the synchronization offset, we proposed a system combining GPS synchronization and Differential TDOA (DTDOA), which introduces a Reference

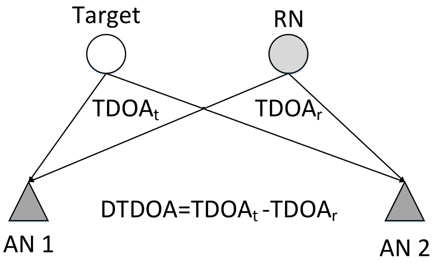


Figure 12: Differential TDOA

Node (RN) to compensate the GPS synchronization offset [2]. DTDOA is defined as the difference of TDOAs for the target and a RN between the same pair of ANs. Figure 12 illustrates the operation of DTDOA.

2.2 High-resolution Timestamp Solution: Time Recovery

Sub-sample timestamps with high resolution are a prerequisite of acquiring accurate time measurements. An advanced method to achieve sub-sample timestamps with nanosecond resolution by using timing information from the signal processing chain is proposed in our previous work [2].

To obtain a high accuracy sub-sample timestamp, the receiver system is designed to include two consecutive phases. First, once the receiver starts to receive packets and generate samples, we can count the generated samples and obtain the sample-based timestamp $T'(k)$ for the kth sample as follows,

$$T'(k) = T'(1) + T_s * (k-1), \tag{3}$$

where $T'(1)$ is the sample-based timestamp for the first sample in the received stream, and T_s is the sampling interval. In Equation (3) the resolution of the sample-based timestamp is limited by T_s.

Second, the sample-based timestamp is further improved by the normalized timing error $\mu(k)$, which can be obtained from the symbol time recovery. The time recovery method [3] is used to correct for shifts in the sampling position during signal recovery. With the $\mu(k) = \frac{\Delta T(k)}{T_s}$, we can improve the resolution of a sample-based timestamp as,

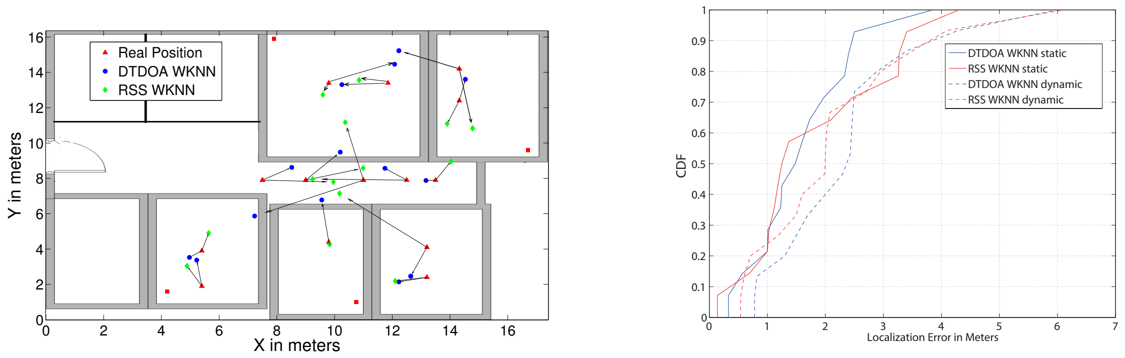
$$T(k) = T'(k) + \mu(k) \cdot T_s, \tag{4}$$

where $T(k)$ is the sub-sample timestamp and $\Delta T(k)$ is the absolute timing error. We propose to apply the sub-sample timestamp $T(k)$ in a narrow-band system for localization.

2.3 Multipath Solution: Time-based Fingerprinting

Fingerprinting relies on an offline phase in which the mobile device moves through interesting environment in different training positions and records the radio parameters, e.g., DTDOA in our work, to form a radio map (database) with DTDOA vectors ($\mathbf{DTDOA_i}$, where i indicates the ith training position). Our work focuses on network-based localization and thus the radio parameters are measured and collected at ANs.

Once the offline training phase is complete, the position of a target can be estimated by performing a radio scan and feeding the measured **DTDOA** vector to a localization algorithm. The easiest localization algorithm of fingerprinting is Weighted KNN (WKNN).



(a) Positioning in Static Environment

(b) CDF of Localization Errors for DTDOA and RSS-based Fingerprinting

With WKNN algorithm, the Euclidean distances between **DTDOA** in current position and all the training positions are first calculated as,

$$e_i = \|\mathbf{DTDOA} - \mathbf{DTDOA_i}\|, \tag{5}$$

where $\|\cdot\|$ indicates the norm value of a vector. Second, we set $K = 3$, which means that the three positions in the database with the minimum e_i are selected. Finally, the position (x, y) of the target is calculated as

$$(x, y) = \sum_{i=1}^{3} \frac{w_i}{\sum_{j=1}^{3} w_j}(x_i, y_i), \tag{6}$$

where w_i is the inverse of e_i, $w_i = \frac{1}{e_i}$, and (x_i, y_i) are the coordinates of the ith training position.

3 Measurement Results for Time-based Fingerprinting

To test the performance of the proposed time-based fingerprinting localization algorithm, we have designed a Zigbee testbed, which is comprised of several software defined radio receivers. The receivers are able to passively overhear the packets from IEEE 802.15.4 signal emitters (TelosB node in our work) and accurately timestamp the packets.

In order to analyze the performance in different scenarios, our measurements are conducted in both static and dynamic environments. In the static environment, both offline and online phases were conducted during the weekend when there was no change of the layout and no people movement. In the dynamic environment, the online phase was conducted during the working time, which was 5 days later than the creation of the radio database.

3.1 Static Environment

Figure 13(a) indicates the measurement results of DTDOA-based and RSS-based fingerprinting algorithms at fourteen testing positions. Among them, seven positions are located in the areas, where the target has Line Of Sight (LOS) connection to one of the ANs, and are thus referred to LOS area. Seven positions are in the areas, where the target has no LOS connection to any AN, and are referred to Non-LOS (NLOS) area. Figure 13(b) summarizes the Cumulative Distribution Functions (CDF) of the localization errors for the DTDOA and RSS-based fingerprinting algorithms.

In the static environment, DTDOA fingerprinting achieves 90% localization errors below $2.5m$, which outperforms RSS-based fingerprinting by about $0.8m$. For median accuracy, DTDOA fingerprinting achieves $1.5m$ accuracy, which is $0.2m$ worse than RSS-based fingerprinting. Furthermore, for the small localization errors (smaller than $1.5m$), DTDOA and RSS-based fingerprinting achieve quite similar performance, but for large localization errors, DTDOA-based fingerprinting generally performs better.

3.2 Dynamic Environment

We conducted measurements in a dynamic environment with the same database. The dashed lines in Figure 13(b) show the CDF of the localization errors.

The performance of both DTDOA and RSS-based fingerprinting in the dynamic environment generally get worse. Both DTDOA and RSS-based fingerprinting achieve 90% localization error below $3.8m$. The reason is that during the working time the people move in the offices and the layout of the surrounding environments also changes, e.g., the door are open or closed. Therefore, these factors influence the accuracy for matching algorithms, i.e., WKNN, to find the correct neighbors based on the original database, and thus the performances of both DTDOA and RSS-based fingerprinting deteriorate. Generally, DTDOA is more sensitive to the outdated database, where the median accuracy decreases by about 60% from $1.5m$ to $2.4m$. For RSS-based fingerprinting, the median accuracy decreases by about 54% from $1.3m$ to $2m$.

4 Conclusion and Future Work

Through experiments, we have demonstrated that DTDOA in the narrow-band system is feasible for fingerprinting and achieves 90% localization errors below $2.5m$ in static environments and $3.8m$ in dynamic environments.

For fingerprinting, in future work, RSS and DTDOA parameters can be fused to further improve localization accuracy. Furthermore, in dynamic environments, the layout changes in the surrounding environments and people movement make the performance of DTDOA-based fingerprinting deteriorate. Therefore, we propose to analyze how to keep the database updating and investigate multipath mitigation algorithms to reduce the dynamic influence.

Since both RSS and DTDOA-based fingerprinting are very time-consuming to construct the radio map and sensitive to the layout change, we are currently working on passively WiFi range-based localization. WiFi is with wider bandwidth, i.e., $20MHz$, than Zigbee, and thus should achieve higher accuracy in theory. We propose to consider channel information to mitigate the multipath influence for WiFi timestamp.

References

[1] Z. Li, D.C. Dimitrova, T. Braun, and D. Rosario, “Highly Accurate Evaluation of GPS Synchronization for TDOA Localization,” *Wireless Days (WD), 2013 IFIP*, pp. 1-3, 2013.

[2] Z. Li, D.C. Dimitrova, D. Hawes and T. Braun, “TDOA for Narrow-band Signal with Low Sampling Rate and Imperfect Synchronization,” in *7th IFIP Wireless and Mobile Networking Conference (WMNC)*, 2014.

[3] M. Heinrich, M. Moeneclaey, and S.A. Fechtel, “Digital Communications Receivers: Synchronization, Channel Estimation and Signal Processing,” in *John Wiley and Sons*, 1998.

Swarm Intelligence Based Construction Strategy for Multi-Objective Data Aggregation Tree in Wireless Sensor Networks

Yao Lu
École d'ingénieurs et d'architectes de Fribourg
Yao.Lu@edu.hefr.ch

Abstract

Data aggregation tree has been widely used as an efficient method to reduce high data redundancy and communication load in wireless sensor network. However, the tree construction normally considers only energy consumption and neglects other practical requirements, such as network lifetime, aggregation latency and communication interference. How to find the optimal structure of data aggregation tree subjected to multi-objectives becomes a crucial task, which can be abstracted as multi-objective Steiner tree problem. In order to address this issue, a multi-objective optimization framework is proposed, and two heuristic strategies based on swarm intelligence are introduced to discover Pareto optimal solution. Static strategy can obtain the solution with higher quality, but the performance becomes inferior when the topology and data interest dynamically change. Dynamic strategy has better adaptability and decentralized implementation, but the convergence time and quality of solutions may not be as good as static strategy. According to the different characteristics, the selection of static or dynamic strategy can be determined by the application demands.

Keywords: Wireless Sensor Networks; Data Aggregation; Steiner Tree; Swarm Intelligence

1 Introduction

One of the most significant functions of wireless sensor networks (WSNs) is gathering data from the environment. Since sensors are intentionally and densely deployed, the data gathering events are possible to concurrently trigger the responding actions from a portion of sensors. In the normal case, direct data transmission from source nodes to the sink node leads to high data redundancy and communication load. Therefore, data aggregation is developed to address this problem [1]. Tree aggregation as a typical technique outperforms others on long-term and static aggregation events. Its general principle is gathering data based on the tree structure, source nodes transmit original data to relaying nodes, which have the aggregation function and are responsible for eliminating the redundant data, and afterwards the aggregated result is transmitted to the higher capable relaying nodes until the sink node is reached.

Nevertheless, there is a significant issues that has to be considered: which sensors are selected as the relaying nodes. It can be abstracted to an NP-complete combinatorial optimization problems, known as Steiner Tree Problem (STP). Given a weighted graph in which a subset of nodes are identified as terminals (sink and source nodes), find a minimum-weight connected subgraph that includes all the terminals. For the purpose of discovering the efficient implementation of tree aggregation, there are multiple performance metrics to evaluate the structure. The selection of metrics is depending on the concrete system requirements. For instance, energy consumption, convergence time, network lifetime, and communication interference are the most conventional performance criterion. Supposing that multiple metrics are concerned simultaneously, constructing the aggregation tree becomes an multi-objectives optimization problem.

The combinatorial issue of multi-objective optimization and STP is called MOSTP. In order to find the near-optimal solution in polynomial time for MOSTP, two kinds of strategies with different characteristics are proposed. The selection of static or dynamic strategy can be determined by the specific application demands.

2 Work accomplished

The static strategy based on jump particle swarm optimization(JPSO) is developed to discover and construct the optimal aggregation tree [2], afterwards the data packages are aggregated on the intermediate nodes of tree structure. Each particle is the solution to represent a tree structure, and the primary difficulty derives from designing efficient encoding scheme and evolutionary operator.

There are two requirements for encoding scheme. First, the encoding can be evolved on incomplete graph, which means the links are included in graph, only if their length are shorter than wireless transmission distance. Second, the involved nodes of encoding are variable in STP, the encoding can be evolved on not only the complete set of all nodes, but also the different subsets of these nodes. The previous schemes potentially generate unfeasible solutions. To enable efficient evolution, the double layer encoding scheme is developed to guarantee the particle flying inside feasible solution space. Correspondingly, specific evolutionary operations are utilized to ensure the particles to ceaselessly fly in feasible solution space. This operator can inherit partial tree structure of attractor (current best particle), and explore the optimal position. Through the simulation results, our approach can generate the approximate optimal tree structure for MOSTP, and the performance is better than other methods.

The good convergence time and quality of solutions are guaranteed in this strategy, but the performance becomes inferior when the topology and data interest dynamically change, furthermore, the global knowledge is required.

3 Work in progress and future work

Dynamic strategy based on ant colony optimization lets transmission node dynamically decide the next hop node, and updates the pheromone depending on the evaluation of routes [3]. The route of each ant is a path from one source node to sink, and a tree structure is the union of the paths from all source nodes to sink. To explore the routes of optimal aggregation tree, the forward ants are piggyback on the data packages from source nodes to sink, and the backward ants are utilized to deposit or evaporate pheromone. The nodes on the better routes should have more pheromone deposited after a while, therefore, these nodes can be selected by higher probability. When multiple packages concurrently arrive at same intermediate node, the data is automatically aggregated.

For this strategy, better adaptability and decentralization can be achieved, but the convergence time and quality of solutions may not be as good as static strategy. In addition, the execution of data aggregation functions are probabilistic, and a specific method to improve the probability of aggregation is expected.

References

[1] Fasolo, E.; Rossi, M.; Widmer, J.; Zorzi, M., “In-network aggregation techniques for wireless sensor networks: a survey,” *Wireless Communications, IEEE*, vol.14, no. 2, pp.70-87, 2007.

[2] S. Consoli, J. A. Moreno Perez, K. Darby-Dowman, N. Mladenovic, “Discrete Particle Swarm Optimization for the Minimum Labelling Steiner Tree Problem,” in *Nature Inspired Cooperative Strategies for Optimization*, vol.129, pp.313-322, 2008.

[3] Dorigo, M.; Gambardella, L.M., “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol.1, no.1, pp.53-66, 1997.

DynamiK: Lightweight Key Management for Privacy-Preserving Pub/Sub

Emanuel Onica
Université de Neuchâtel
emanuel.onica@unine.ch

Abstract

Content-based publish/subscribe (pub/sub) is an appealing communication paradigm for distributed systems. Consumers of data subscribe to a pub/sub service, typically offered through a distributed broker overlay, and indicate their interests as constraints over the information content. Publishers generate the information flow, which the brokers filter and route to the interested subscribers. Protecting the privacy of subscribers and publishers is an important concern when the brokers are located in untrusted domains (e.g., public clouds). Encrypted matching techniques allow untrusted brokers to match encrypted subscriptions against encrypted publications without accessing their content. These solutions require appropriate key management support, taking into account the decoupled nature of pub/sub communication. Due to the use of encrypted subscriptions stored in untrusted domains, a key update may require all subscribers to re-encrypt and resubmit their subscriptions before publishers may use the new key. This is a costly and long operation. We present DynamiK, a lightweight key management architecture that takes into account the decoupled nature of pub/sub and allows updating encrypted subscriptions directly at the brokers. We perform a security analysis and implement DynamiK for the ASPE encryption scheme. Finally, we evaluate the performance of key updates and their impact on the pub/sub service performance.

Keywords: key management; publish/subscribe; encrypted matching.

1 Introduction

Content-based publish/subscribe (pub/sub) [1] is an information dissemination model used in distributed systems for delivering data produced by different sources (*the publishers*) to subsets of interested clients (*the subscribers*). Subscribers register subscriptions to a pub/sub service. These subscriptions are composed of constraints over a set of attributes. The pub/sub service is typically provided by a set of distributed *brokers*. Publishers disseminate information by sending publications to any of the brokers. A publication is composed of a header including a set of attributes that characterize the published information, and an optional payload. Brokers filter the publication flow by matching the publication headers against the constraints in stored subscriptions. Matching publications are then routed to the interested subscribers. An illustration of the generic flow of information in a pub/sub system is depicted in Figure 13.

The confidentiality of subscriptions and published information is a major hurdle for the deployment of pub/sub applications [2]. The broker service is typically located in a publicly accessible domain such as a public cloud infrastructure. Although the brokers are expected to behave according to the system specifications, this does not guarantee that the data they operate on remains confidential. Due to such liabilities, the typical assumption is to consider brokers as *honest-but-curious*, and to encrypt the pub/sub data passing through the broker overlay. Using normal encryption techniques unfortunately impairs the brokers from

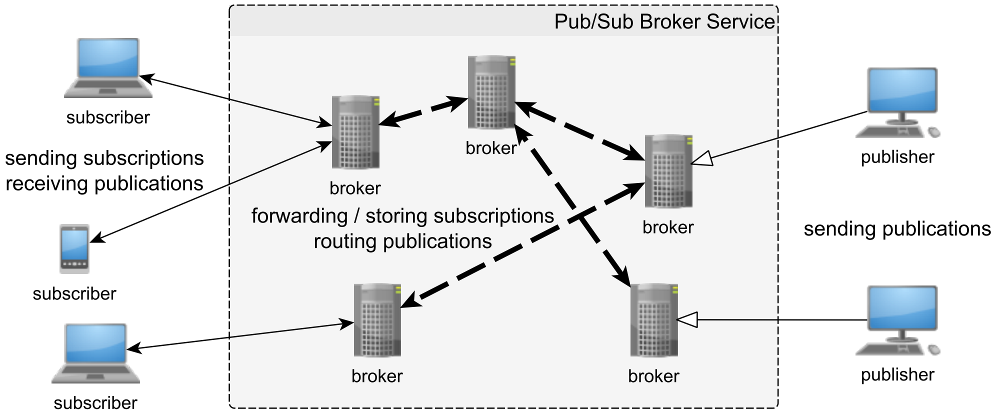


Figure 13: Generic broker-based publish/subscribe system.

matching messages based on their content. However, recently proposed specific encryption techniques allow the brokers to match the encrypted data without disclosing the actual content of the messages. We call such an operation *encrypted matching.*

One of the most interesting encrypted matching schemes in terms of supported functionality is Asymmetric Scalar Product-preserving Encryption (ASPE) [3]. This, as also most of other encrypted matching schemes, requires a solution for key exchange between communicating parties. Key management is an issue that is considered orthogonal in most of the existing work on secure pub/sub, and generally accepted to be a cumbersome issue impairing deployment of practical solutions.

A difficulty lies in the fact that publishers have no a-priori knowledge of the subscribers which will receive a given publication. Besides the initial key exchange, updating a key introduces even more serious challenges. All subscriptions stored by the brokers and encrypted with the old key can no longer be matched with publications encrypted with the new key. A naive solution requires that the clients resubmit all their subscriptions, re-encrypted with the new key. This forces subscribers to keep track and store their set of previous subscriptions. Also, the key update process becomes prohibitively long (depending on the constraints of the network layer) and resource consuming, as a large amount of incoming subscriptions and un-subscriptions have to be handled by the brokers. Consequently, the impact on the quality of service, e.g., in terms of throughput and delays, can become significant.

2 Work accomplished

To address the issues described we have developed DynamiK, a key management architecture maintaining a partial decoupling between the individual publisher and subscriber nodes. Our solution eliminates the need for subscription resubmission upon a key update by introducing a particular requirement for the encrypted matching scheme: the ability to perform *in-broker subscriptions re-encryption.* Untrusted brokers storing subscriptions can be given a specific token K_R directly related to the encryption key that allows them to update encrypted subscriptions *in place.* Obviously, this token should not permit obtaining the original encryption key.

The components of our architecture, displayed in Figure 14 can be summarized as:

- A *grouping* of the subscribers, publishers and brokers in established security domains, which allows disseminating a common key to communicating hosts.

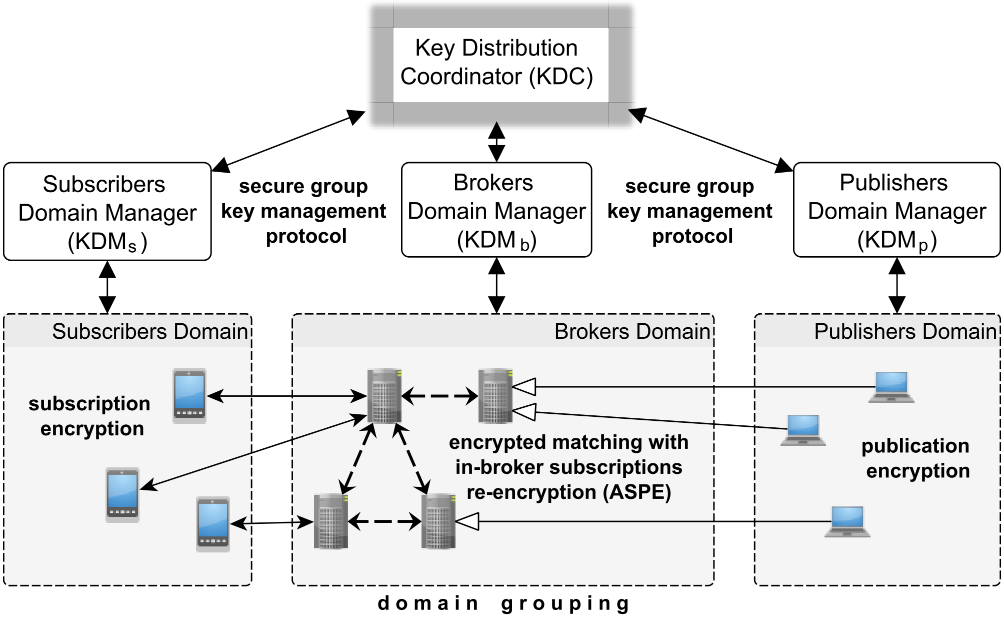

Key Distribution Coordinator (KDC)
Subscribers Domain Manager (KDMs)
secure group key management protocol
Brokers Domain Manager (KDM b)
secure group key management protocol
Publishers Domain Manager (KDMp)
Subscribers Domain
subscription encryption
Brokers Domain
encrypted matching with in-broker subscriptions re-encryption (ASPE)
Publishers Domain
publication encryption
domain grouping

Figure 14: The DynamiK architecture.

- A *secure group communication key management protocol* used to distribute the key in the established domains.
- A *in-broker subscriptions re-encryption* function that must be supported by the chosen encrypted matching scheme.

For the hosts grouping we have used, without loss of generality, a simple partitioning in three domains: subscribers, publishers and brokers. The protocol we used for distributing the key is based on a simple hierarchy using public key encryption in which a central coordinator (KDC) sends the key to domain managers (KDMs), who further disseminate it to individual hosts in their domain. The protocol takes advantage of the ZooKeeper coordination service [4] for the key distribution and synchronization, inheriting its dependability.

Most of our work was concentrated on the last item of the architecture, the in-broker subscriptions re-encryption. We implemented support for this operation on top of the ASPE encrypted matching scheme [3]. In this purpose we modified the scheme, by substantial additions to its original version. We proved that our additions do not allow deriving the actual encryption keys. We also added changes that increase the security of the scheme in its basic form.

Our architecture implementation integrates with the StreamHub [5] content-based pub-/sub engine. The evaluation results we obtained demonstrate that the key update can be conducted with a negligible impact on the quality of service.

3 Work in progress and future work

We assume that key management protocols dedicated for secure group communication in distributed systems can achieve better performance in disseminating the key. We study adapting such protocols for secure pub/sub scenarios. Also, we investigate the possibility of obtaining support for in-broker subscription re-encryption for other encrypted matching schemes besides ASPE.

References

[1] P. T. Eugster, P. Felber, R. Guerraoui and A. M. Kermarrec, “The many faces of publish/subscribe”, *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, 2003.

[2] C. Wang, A. Carzaniga, D. Evans and A. Wolf, “Security Issues and Requirements for Internet-Scale Publish-Subscribe Systems”, in *Proc. of the 35th Annual Hawaii International Conference on System Sciences*, 2002, p. 303.

[3] S. Choi, G. Ghinita, and E. Bertino, “A Privacy-Enhancing Content-Based Publish/-Subscribe System Using Scalar Product Preserving Transformations”, *Lecture Notes in Computer Science*, vol. 6261, pp. 368–384, 2010.

[4] P. Hunt, M. Konar, F. Junqueira, and B. Reed, “ZooKeeper: Wait-free Coordination for Internet-scale Systems”, in *Proc. of the 2010 USENIX Conference on USENIX Annual Technical Conference*, 2010, p. 11.

[5] R. Barazzutti, P. Felber, C. Fetzer, E. Onica, M. Pasin, J. F. Pineau, E. Rivière and S. Weigert, “StreamHub: A Massively Parallel Architecture for High-Performance Content-Based Publish/Subscribe”, in *Proc. of the 7th ACM International Conference on Distributed Event-Based Systems*, 2013.

Network Coding in Content Centric Networks

Jonnahtan Saltarin
Universität Bern
saltarin@iam.unibe.ch

Abstract

We are studying how the network coding concept can be used in content centric networks (CCN). In a first scenario, a source generates content and places it in one or multiple repositories. The content object packets generated by the source should contain a network coding (NC) header, and can be mixed or non-mixed packets. A client interested in this named data should send Interest messages with its name, without including any NC header in the name. In this way, the sender can forward any packet available or re-encoded packets. The Interest message should include a field notifying that the receiver is Interested in network coded content. Interests would be sent by the client through its faces, based on the strategy layer of CCN. An intermediate node receiving that Interest should forward it to upstream nodes if it doesn't have any data that match the Interest. Instead, if the intermediate node has data in its content store that match the Interest, it should decide whether (a) a content object or a combination of two or more objects should be sent in response to the Interest, or (b) the Interest should be forwarded to its upstream nodes, and wait for fresh content objects to arrive before sending the response to the Interest. Currently, we are developing and implementing the base algorithms that will allow us to test a first solution to integrate NC and CCN.

Keywords: Network Coding; Content Centric Networking; CCN, CCNx.

1 Introduction

In this project we will evaluate the benefits that network coding could bring to content centric networks, and develop a protocol based on CCNx, an implementation of CCN developed by the Palo Alto Research Center (PARC) [1]. In, we will propose a protocol that enables network coding into the current version of CCNx. This document presents the first version of this protocol. Before we will introduce some basic concepts of content centric networking, and also some concepts of network coding.

Content centric networking (CCN) is a new network model in which the communication is driven by the name given to the data, instead of the IP addresses of the nodes that are communicating. In CCN, when a given node is Interested in certain information, it pushes an Interest message into the network, specifying the content it is Interested in. Nodes receiving this Interest message should reply with the requested information, if they have it, or forward the Interest message otherwise. In particular, for this project we will use the CCNx implementation [1].

Network Coding [2] is a technique in which the nodes of a network not only forward packets towards its destination, but also encode the received packets, before forwarding them. This has shown to improve the network's throughput for some topologies, especially the ones containing bottlenecks. As an example, consider the network shown in Figure 1. In a scenario without network coding, if both nodes A and B send data to both Receivers 1 and 2, two timeslots will be needed to complete the communication over link CD. Instead if

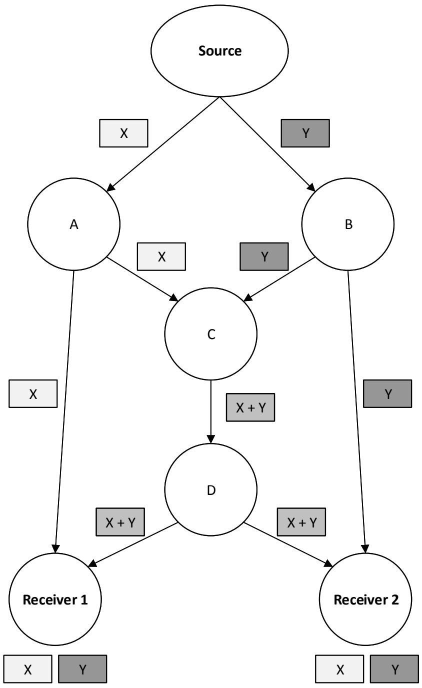


Figure 15: Butterfly network

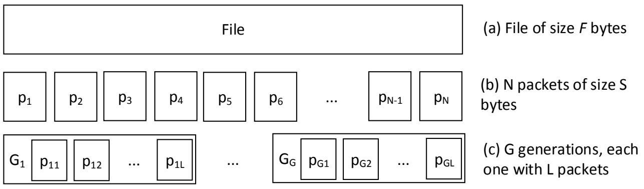

File
(a) File of size F bytes
p_1 p_2 p_3 p_4 p_5 p_6 ... p_{N-1} p_N
(b) N packets of size S bytes
G_1 p_{11} p_{12} ... p_{1L} ... G_G p_{G1} p_{G2} ... p_{GL}
(c) G generations, each one with L packets

Figure 16: Data Segmentation

network coding is allowed, node C can mix packets from node A and B, and communication over link CD will be completed in only 1 timeslot. A practical approach for implementing network coding has been presented in [3].

This document continues as follows: In section 2 we present the proposed protocol, explaining the behavior of each type of node in our network model. Then we present the next steps of our research, in section 3.

2 Work Accomplished

Network coding and content centric networking are the two main topics addressed in this project, where we have developed a first version of a protocol to use network coding in content centric networks. Below, we describe this protocol.

Our protocol is based on Content Centric Networking (CCN) [1], and Practical Network Coding [3]. We consider a network in which nodes could be (a) Sources that have all the content in their memory, (b) Receivers that are the nodes interested in receiving the information, and (c) Intermediate Nodes that connect the sources and the receivers.

2.1 Source Nodes

In our scheme we consider nodes that have certain data stored in its CCN repository as sources. This data may be generated by the same node or uploaded by other nodes.

This information could be sent without any coding, or could be network coded before sending, since the source has all the original segments. However, coded and not coded information should follow the same data segmentation and naming scheme.

2.1.1 Data Segmentation and Tagging

Our segmentation scheme is similar to the one presented in [3]. In it, data is partitioned into packets of the same size. Then, these packets are grouped into generations of size h packets. Network coding operations are applied to packets of the same generation. The partitioning process is performed only in by sources, before making the content available to other nodes. A graphical description of data segmentation in sources is shown in Figure 1.

Each generated packet can be identified by the ID of the generation to which it belongs and its position inside the given generation. In [3] these identifiers are appended as a header to each source packet. Generation ID is an increasing number (e.g. an integer), while the position of the packet inside the generation is represented as a vector with the same size as the generation, with 0's in all positions, except the one referring to the packet.

In contrast to [3], our proposed scheme does not append the network coding header at the beginning of each packet, but instead appends this information to the CCN name. This will be further discussed in the next section.

2.1.2 Naming scheme

Our proposal follows the naming scheme of the CCN protocol specification. However, we extend the prefix with two parameters: the generation ID and the network coding header of the packet. Also, in the first version of our scheme, we do not use the version and segment components, even if by default they are appended to the prefix. In a later version of our scheme we might consider the way to use the version and segment number. Consider a data object with the name "the/name.file". After segmenting the video, it came out to be g generations of h packets each one. The first component to extend the prefix would be the generation ID. The form of the generation ID can be defined by each application. For simplicity, here we consider it as an increasing number starting from 1. Thus, in our example the prefixes extended with the generation ID would range from "the/name.file/1" to "the/name.file/10". After the generation ID, the network coding header would be appended. The network coding header is composed of h symbols containing the coefficients used to compose the payload of the packet.The size of these symbols is determined by the size of the Galois field in which the operations are performed. This could be passed as a parameter inside the content object. This network coding header is an array of bits, and will be appended to the prefix as that.

2.2 Receiver nodes

Receiver nodes, or clients, should decide if they would like to receive network coded data or non-network coded data. If a client decides not to receive any network coded data, it can send a normal CCN Interest, and should receive non-network coded Content Objects.

Any node interested in receiving network coded data should send an Interest with the flag "NetworkCoding" activated. When content is received, clients should store it in their memory, to be decoded as soon as they receive enough data segments (that means that the global encoding matrix reaches the full rank). In this first version of our protocol, the nodes use the same storage used in CCN as the place to store the network coded data. However, this might change in future versions. A more efficient storage may be used to support the matrix operations needed by network coding.

2.3 Intermediate Nodes

Intermediate nodes should distinguish among Interests for network coded data and Interests for normal data. Interests for non network coded data should be treated as defined in the CCN protocol. Interests for network coded data should be treated in the following way:

If no matching data is available in the content store of an intermediate node, it should simply forward the Interest according to the strategy layer, as in the current version of CCN.

If matching data is available in the buffer, the node should decide how to react to the Interest. The node might keep track of the Content Objects sent through a given face, and avoid sending them again over the same face. A better case would be to avoid sending any Content Object that can be expressed as a linear combination of any sub-set of Content Objects already sent through this face. However, verifying if a Content Object is a linear combination of other Content Objects will add too much complexity to the system. For that reason, in this version of our protocol we will stick to the first option: intermediate nodes will store the face over which each Content Object has been sent. Then, every time an Interest for network coded data arrives over a certain face, the node will only reply with content that has not been previously sent over that face. If all the content available in the node's content store has been already sent over the given face, the Interest will be forwarded to other nodes. In addition, a receiver driven approach is also proposed. In it, the receiver should fill the 'Exclude' field of the Interest with the name of the packets that are already in its memory. Intermediate or source nodes should avoid replying to such an Interest with any Content Object which name matches any of the names in the 'Exclude' field. We plan to evaluate both approaches separately, as well as a combined version.

3 Work in progress and future work

We are currently finishing the implementation of our protocol. We plan to evaluate our protocol using the NS-3 Direct Code Execution (DCE). We will compare our protocol with an unchanged version of CCNx and evaluate the delay, the amount of transmissions needed to download the same data, the processing needed in each node, and other metrics. With the first tests, we should be able to adjust our protocol in order to improve the results. We should also evaluate in which cases network coding has advantages and in which not. This will lead to the next step, which is the development of a scheme in which nodes decide to implement network coding or not.

In the future we plan to test our protocol with multimedia applications, making the necessary changes to the current implementation and evaluating new metrics, more appropriate to video scenarios, like the Peak Signal-to-Noise Ratio (PSNR). We should also study how communications can be secured and signed. The current CCN signature scheme does not work properly with the proposed network coding protocol, since intermediate nodes will change (network encode) the content and then the signatures are not valid anymore. Thus, we would need to design a new signature scheme.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York,

NY, USA: ACM, 2009, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658941

[2] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[3] P. Chou and Y. Wu, "Network coding for the internet and wireless networks," *Signal Processing Magazine, IEEE*, vol. 24, no. 5, pp. 77–85, Sept 2007.

Enterprise Integration of Smart Objects using Semantic Service Descriptions

Matthias Thoma
University of Bern / SAP (Switzerland) Inc.
thoma@iam.unibe.ch

Abstract

We present a research project aiming for investigating the opportunities that lie in the usage of an semantic overlay for Internet of Things application layer protocols. The projects aims for enabling interoperability by introducing a semantic overlay based on Linked USDL. We introduce an extension called Linked USDL for IoT (Internet of Things) taking the special need of IoT application layer protocols and typical communication patterns into account. Furthermore, everything is embedded into an semantic framework enabling automatic creation of code, which is integrated into existing application layer protocols. We furthermore research the properties of application-layer protocols in an semantically enhanced environment in terms of, among others, energy consumption, CPU and memory resource consumption or service access times.

The whole system is tested in a close to real world setting. We implemented a novel enterprise integration platform, which has been implemented for evaluation and experimentation purposes. Additionally, empiric research has been conducted to determine the properties such a system should have and to derive recommendations for the further development of the system.

Keywords: Semantics, Internet of Things, Application Layer Protocols, Wireless Sensor Networks, Cyber-Physical Systems

1 Introduction

The aim of the project is to investigate new methodologies to enable interoperability between wireless sensor networks (in general various heterogeneous Internet of Thing (IoT) devices) and enterprise IT systems. The project assumes that the lower layers of a typical IoT/WSN protocol stack is mature and concentrates on application layer protocols, service-based integration of devices and (semantic) data content abstraction.

To overcome the gap between the specialized knowledge needed to run IoT-devices on one hand and the usage of service-based architectures and enterprise service bus architectures on the other hand we propose semantic service overlays in the context of IoT application-layer protocols. We are trying to research answers to the following questions:

1. Despite the millions of Euros and man years invested into that technology why does semantics not have the widespread adoption that was predicted? What is the current real-world adoption of Internet-of-Things services and protocols.

2. Given the heterogeneity in application layer protocols and the need for custom protocols: How would such a semantic overlay need to be designed? How could it be integrated into an enterprise environment? How can legacy protocols be supported? Can semantic service descriptions be used for automatic creation of adapters?

3. How do different application-layer protocols perform (in terms of energy consumption, round trip time etc.) when semantic workload is added to enable interoperability? The main focus here is also the memory and CPU resource consumption, as business processes are anticipated to run on such devices.

4. Exploration of the possibility of automatic code generation based on (semantic) knowledge repositories and their integration into unicast/multicast systems and sensor grouping: Can the knowledge about entities and sensors be utilized to take advantage of sleepy node states and improved communication schemes to save energy and increase the lifetime of the sensor network?

5. Can existing enterprise-level protocols designed for interoperability (i. e. OData) be downscaled for the use on constrained devices? What are the prospective opportunities of such an approach and what are the drawbacks?

6. In case of very constrained devices, where application-layer protocols still might be too heavy-weight, can supernodes be used to make application-layer protocols feasible?

To gain a better insight into the domain and its specific problems and possible solutions we base our work on empirical research, as well as experimentation. For evaluation and experimentation in a close to real world setting we implemented a novel enterprise integration platform, based on a semantic service description language (Linked USDL). It supports modeling IoT/WSN specific details, including technical interface descriptions, data representation (input/output) as well as different communication patterns. For enterprise IT systems that do not support a specific application layer protocol the semantic descriptions enable algorithmic solutions for automatic conversion between technical interfaces and automatic creation of further technical interfaces. The semantic representation of services and things support seamless integration of various heterogeneous devices and abstracts the things monitored by a wireless sensor network away from the actual sensing devices, allowing a domain expert to model a business process or business rules easily without the need of having specific technical knowledge about the sensing devices. First evaluation results show that the performance of the platform is very promising and the overhead imposed by the semantic layer is reasonable compared to alternatives such as WSDL.

2 Work accomplished

We surveyed the status of semantics and Internet of Things application layer protocols in academic and industrial use [NOMS2014]. It became obvious, that the IoT domain remains to be highly heterogeneous. While semantics are expected to play a role in future IoT systems, there is still a way to go. Most participants see some benefit in the semantic management of things, devices and services. Nonetheless, when looking into the actual situation and the planned usage of semantics in IoT, these benefits seem not to be strong enough to stimulate large scale usage. One possible reason could be a lack of training in semantics and the more "bit and byte"-oriented skillset current embedded developers have. Even if there seems to be the expectation that IPv6/6LoWPAN will play a crucial role and finally make the IoT vision a reality. CoAP, while currently not used at a large scale, is expected to be for IoT what was HTTP for the WWW. Nonetheless, the number of people using or expecting the use of custom protocols in the future is quite high. Gateways or proxies will still be widely used. It seems as if a convergence towards an Internet standard might not happen as soon as expected.

The aforementioned survey, together with an earlier study on services in the Internet of Things [iThings2012], served as a basis for design and empirical evaluation of Linked USDL for IoT. Linked USDL for IoT [LUSDL4IOT] is an extension of Linked USDL [LUSDL2014] featuring the ability to model several application-layer protocols and encoding schemes. It allows to model many IoT specific quality of information and quality of actuation parameters. It is designed to not only work with high-level application layer protocols, but does also support custom protocols with its data format specified in ASN.1. Furthermore, it supports modeling communication patterns (such as Publish/Subscribe) and an automatic conversion (adapter generation) between endpoints [WCNC2014]. We have shown that semantic endpoint descriptions in conjunction with smart endpoint generation can be used to efficiently integrate IoT-based smart objects into enterprise platforms. Compared to existing solutions we integrated not only ways of establishing technical interoperability [IWEI2013], like pure RDF or WSDL does, but support also legacy protocols or protocols with special requirements as often found in sensor network environments. For example, in settings where energy consumption is a major issue, the semantic service descriptions with ASN.1 combined with smart endpoints can be used to ensure interoperability. Our evaluation has shown that endpoint descriptions in RDF do not need considerable more memory than SOAP or XML REST based services and evaluated the price in terms of energy for application level protocols and latency for introducing another layer of indirection.

We introduced an abstraction for (semantic) business entities in an Internet of Things context, called semantic physical business entity (SPBE) [WD2013]. That abstraction serves as a theoretical foundation of our integration platform. It decouples the entities from the actual sensing devices and enables writing applications without any knowledge about the underlying hardware. Interoperability and machine-readability is achieved through ontologies and common vocabularies. Data is gathered and aggregated by an integration platform using a query language. The platform compiles the services depending on the query which is specified in a custom language called SPBEQL. SPBEQL triggers the code generation and the deployment on the mote. The actual deployment is based on information stored in semantic domain specific knowledge repositories.

Furthermore, we evaluated the use of enterprise-level protocols [WONS2014], like OData, as part of an IoT platform. We were able to show that the approach of implementing a high-level protocol on small motes is feasible and that the extra costs in terms of energy consumption or round-trip time are small. Nonetheless, on very limited motes (like IRIS motes with only 8kb of RAM) resource consumption can become an issue. In such cases we evaluated the use of a supernode (assistant node). We demonstrated that a supernode is able to solve the resource constraints for a small additional cost in terms of energy and round-trip time. In some cases, though, especially with long running tasks involving flash-paging the use of a supernode can lead to lower energy consumption and faster response times.

3 Work in progress and future work

Linked USDL has recently received a major updated. Currently we are working on adapting Linked USDL4Iot to be compatible to the most recent version. Furthermore, we are preparing an comprehensive evaluation of our CoAP implementation for IBMs Moterunner system. To our knowledge, this is the first and only implementation on a JAVA-based embedded system. As most of our research was done on a Java based platform we are also preparing an empirical evaluation of Java based development for Wireless Sensor Nodes.

The main research area currently is how the semantic overlay can be used in unicast or multicast based CoAP systems, including the use of sleepy nodes. Here we are investigating how the information in the semantic repositories can be used to reach a optimal distribution of sensing nodes, the building of groups of sensors (one-to-many relationship between CoAP endpoints) and maximizing the sleeping times of the sensor nodes to increase the lifetime of the wireless sensor network.

References

[NOMS2014] Matthias Thoma, Torsten Braun, Carsten Magerkurth and Alexandru-Florian Antonescu, “Managing Things and Services with Semantics: A Survey”, *IEEE/IFIP Network Operations and Management Symposium (NOMS 2014)*, 2014, Krakow, Poland

[WONS2014] Matthias Thoma, Theofilos Kakantousis and Torsten Braun, “REST-based sensor networks with OData”, *Wireless On-demand Network Systems and Services (WONS), 2014 11th Annual Conference on*, 2014, Oberurgl, Austria

[WCNC2014] Matthias Thoma, Torsten Braun, Carsten Magerkurth, “Enterprise Integration of Smart Objects using Semantic Service Descriptions”, *IEEE Wireless Communication and Networking Conference (WCNC 2014)*, 2014, Istanbul, Turkey

[WD2013] Matthias Thoma, Klaus Sperner, Torsten Braun and Carsten Magerkurth, “Integration of WSNs into enterprise systems based on semantic physical business entities”, *Wireless Days (WD)*, 2013 IFIP

[IWEI2013] Matthias Thoma, Alexandru-Florian Antonescu, Theano Mintsi and Torsten Braun, “Linked Services for the Sensing Enterprise”, *Enterprise Interoperability*, Lecture Notes in Business Information Processing, Volume 144, Springer, 2013

[iThings2012] Matthias Thoma, Sonja Meyer, Klaus Sperner, Stefan Meissner and Torsten Braun, “On IoT-services: Survey, Classification and Enterprise Integration”, *2012 IEEE International Conference on the Internet of Things*, 2012, Besancon, France

[LUSDL4IOT] Matthias Thoma, Torsten Braun, Klaus Sperner, Carsten Magerkurth, “Linked USDL for IoT”, Technical Report, 2014

[LUSDL2014] Carlos Pedrinaci, Jose Cardoso and Torsten Leidig, “Linked USDL: a Vocabulary for Web-scale Service Trading”, *11th Extended Semantic Web Conference (ESWC 2014)*, 2014, Springer.

Sensor Context-aware Adaptive Duty-cycled Opportunistic Routing

Zhongliang Zhao
University of Bern
zhao@iam.unibe.ch

Abstract

Energy is of primary concern in wireless sensor networks (WSNs). Low power transmission makes the wireless links unreliable, which leads to frequent topology changes. Consequent packet retransmissions aggravate the energy consumption. Beaconless routing approaches, such as opportunistic routing (OR) choose packet forwarders after data transmissions, and are promising to support dynamic feature of WSNs. This paper proposes SCAD - *Sensor Context-aware Adaptive Duty-cycled* beaconless OR for WSNs. SCAD is a cross-layer routing solution and it brings the concept of beaconless OR into WSNs. SCAD selects packet forwarders based on multiple types of network contexts. To achieve a balance between performance and energy efficiency, SCAD adapts duty-cycles of sensors based on real-time traffic loads and energy drain rates. We implemented SCAD in the TinyOS sensor operating system running on top of Tmote Sky sensor motes. Real-world evaluations show that SCAD outperforms other protocols in terms of both throughput and network lifetime.

Keywords: Wireless sensor networks; beaconless opportunistic routing; context awareness; adaptive duty cycle.

1 Introduction

WSN low power radio transmission makes wireless links unreliable. Traditional WSN routing protocols aim to find the shortest path between a source and a destination before data transmission. However, the selected route will be invalid if topology changes. Opportunistic routing (OR) copes with the uncertainty of wireless links by postponing the forwarder selection to the receiver side. In OR, a source node does not determine its forwarder before data transmission. Instead it selects a set of nodes, called *candidates*, and broadcasts the packet. Multiple receivers of this transmission coordinate to select one forwarder. Most of the existing OR protocols statically generate a candidate list according to certain routing metrics prior to data transmission, which is not suitable for the dynamic nature of low power WSNs.

Context-aware communication has increased attention recently since it allows automatic adaptation of protocol behavior to user's changing context. The context is the information characterizing the situation of an entity and providing knowledge about other entities in the environment. Multiple types of context should be exploited efficiently to improve system performance. Context-aware routing enables nodes to learn network states, automatically adjust their behaviors, and thus make optimized routing decisions.

In this paper [1], we integrate beaconless OR with WSNs. We consider the particular requirements of WSNs by taking energy efficiency as the main concern and tailoring OR to duty-cycled sensor nodes. The proposed Sensor Context-aware Adaptive Duty-cycled beaconless OR protocol (SCAD) selects forwarders by jointly considering multiple types of cross-layer context, such as link quality, progress, residual energy, and energy draining rate.

An adaptive duty-cycling scheme has been designed and implemented to tune the sleep intervals of sensor nodes according to traffic load and energy drain rate. Experiment results show that SCAD improves throughput, and prolongs network lifetime compared to other approaches.

2 Work accomplished

This section introduces the mechanism of SCAD, which has been implemented in TinyOS on top of Tmote Sky motes. SCAD is a cross-layer routing approach that utilizes multiple types of context information to forward packets. It includes an adaptive duty-cycle mechanism to control the sleeping intervals of sensors to achieve a balance between energy efficiency and routing performance. Key implementation components of SCAD in TinyOS includes an on-line energy profiling scheme, and an accurate estimation of traffic load at run-time.

2.1 Packet Forwarding Scheme in SCAD

We assume that each node is aware of the locations of itself and the destination. SCAD does not use any beacons to maintain network topology. Instead, the packet forwarding decision is made in a hop-by-hop fashion. Therefore, it conserves the scarce energy for packet transmissions. In SCAD, packet transmission is triggered by broadcasting a data packet. Whenever a node has data to send, it adds the locations of itself and the destination into the packet header, broadcasts it and waits for responses. The neighbors that successfully receive this packet first check their relative closeness to the destination by comparing their distances to the destination with that of the packet sender. If they do not provide any distance improvement, they just drop the packet. Otherwise, they will start a timer, called *Dynamic Forwarding Delay (DFD)* and wait for the expiration of the timer. When this happens, the node will rebroadcast the packet and repeat the same process until the packet reaches the destination. During the count-down of the timer, a neighbor might overhear the re-transmission of the same packet from another node. This indicates that the timer of another neighbor has expired already, and the neighbor should cancel its timer and drop the packet.

2.2 Dynamic Forwarding Delay (DFD)

Whenever receiving a broadcast packet, neighbors that provide distance improvements will start a timer before re-broadcasting this packet. The goal of this timer is to select a forwarding node while avoiding collisions caused by the concurrent re-transmissions from multiple neighbors. In SCAD, we calculate this timer, called *Dynamic Forwarding Delay (DFD)*, by integrating multiple cross layer context information, such as *geographic progress*, *link quality*, *residual energy* and *energy drain rate.*

2.3 Run-time Energy Profiling of Tmote Sky

Sensor energy consumption can be calculated easily in a simulator. However, in a real-world implementation, one challenge of energy-aware protocols is to accurately measure sensors' energy at run-time. The hardware platform such as Tmote Sky nodes and standard WSN operating systems such as TinyOS, does not provide any energy measurement functionality. Hardware-based energy measurements are typically difficult and expensive to add to existing hardware platforms, since they require a significant amount of modifications. Therefore, the energy consumption estimation can only be done using software methods in a real-world deployment.

In this work, we implement on-line energy profiling of the Tmote Sky node running TinyOS by logging the time spent by the main energy-consuming components in different states. Due to the fact that radio transceivers dominate the energy consumption, and also because our protocol does not include high computation overhead, we ignore energy consumption of MCU and memory access. We only consider the energy consumption caused by wireless radio transceivers.

2.4 Adaptive Duty Cycling

Energy is of primary concern. Sensor sleeping intervals should be controlled according to real-time network conditions. An identical sleep interval leads to heterogeneous energy consumption such that nodes located in heavy traffic regions are prone to suffer from frequent data transmission, which leads to fast energy depletion and short network lifetime.

Merely increasing the sleep interval makes sensor nodes sleep longer and save energy, but with a worse performance. To achieve a balance between energy efficiency and performance, SCAD adapts sensor duty cycles according to energy drain rate and traffic load. SCAD integrates duty-cycling features of MaxMAC and IDEA. MaxMAC aims to maximally adapt to changes in the network traffic load. It improves system throughput by assigning additional wake-ups when the rate of incoming packets reaches certain threshold values. However, MaxMAC does not consider the energy drain rate, such that after a highly intensive wake-up period, nodes do not rest longer to compensate their fast energy depletion. On the other hand, IDEA increases the sleep interval of a sensor node if its energy drain rate during a certain interval increases.

In SCAD, nodes start from a state with default wake-up interval of *Base Sleep Interval (T_B)*. Each node persistently estimates the incoming traffic rate within the *sliding window*. When nodes observe an increasing traffic load, they will reduce the sleep interval by assigning additional wake-ups to increase the throughput. In addition to estimating the traffic rates, SCAD nodes also measure the energy consumption within the sliding window to estimate the energy drain rate. When an intensive traffic load passes away (the timespan that nodes promise to stay in the new state, referred to as *LEASE*, expires), a node checks the energy drain rate. If it is above a certain threshold, the node will increase its sleep interval to rest longer and compensate its previous vast energy consumption.

Figure 17 illustrates the state diagram of the duty-cycling adaptation scheme of SCAD. The state transition is triggered by either reaching a traffic rate threshold or an energy consumption rate threshold. SCAD defines six states, and each state is with a different wake-up interval value. More states can be introduced if more precise thresholds of traffic load and energy drain rate are defined. Nodes switch from default state S_0 to state S_1, S_2, and S_3 when the estimated traffic rate reaches the predefined threshold values T_1, T_2, and T_3. When switching to higher states (states with bigger state number), SCAD nodes schedule extra wake-ups and apply a wake-up interval of $\frac{1}{2}T_B$ (at state S_1), $\frac{1}{4}T_B$ (at state S_2), and $\frac{1}{8}T_B$ (at state S_3). SCAD sacrifices performance, if there is a conflict between energy-efficiency and performance. This is illustrated by the introduction of states S_4 and S_5. For example, at state S_2, when the LEASE of S_2 expires, a node might transit into state S_1, S_3, S_4 or S_5, depending on the measured values of energy consumption (EC) and traffic rate (TR) within the last sliding window. If EC is above threshold ET_2, it means too much energy has been depleted in this sliding window and the node should sleep longer to compensate its fast

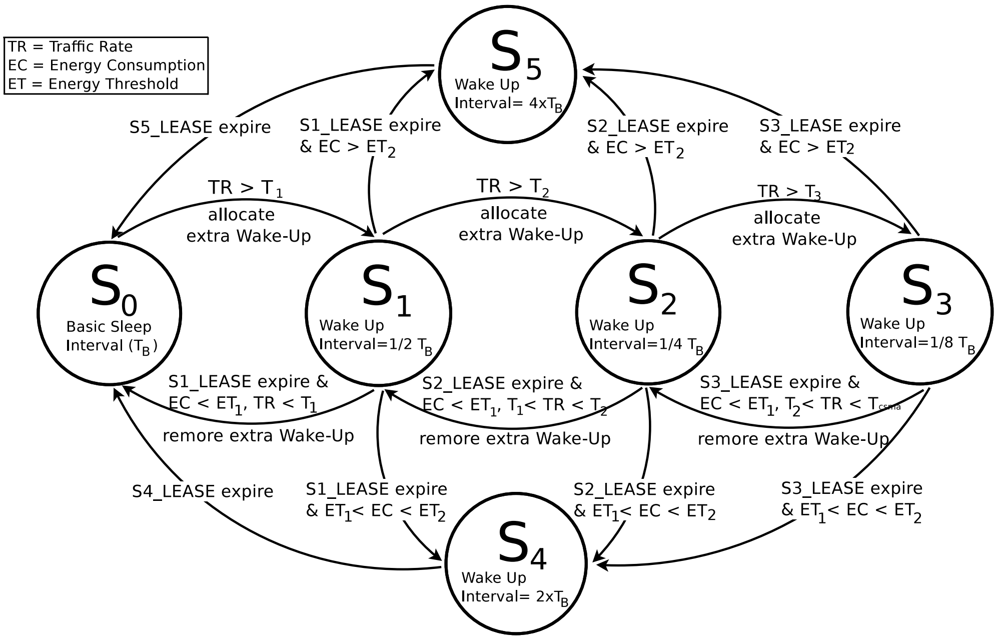

TR = Traffic Rate
EC = Energy Consumption
ET = Energy Threshold
S5
Wake Up Interval= 4xT_B
S0
Basic Sleep Interval (T_B)
S1
Wake Up Interval=1/2 T_B
S2
Wake Up Interval=1/4 T_B
S3
Wake Up Interval=1/8 T_B
S4
Wake Up Interval= 2xT_B
S5_LEASE expire
S1_LEASE expire & EC > ET2
S2_LEASE expire & EC > ET2
S3_LEASE expire & EC > ET2
TR > T1 allocate extra Wake-Up
TR > T2 allocate extra Wake-Up
TR > T3 allocate extra Wake-Up
S1_LEASE expire & EC < ET1, TR < T1 remore extra Wake-Up
S2_LEASE expire & EC < ET1, T1< TR < T2 remore extra Wake-Up
S3_LEASE expire & EC < ET1, T2< TR < Tcsma remore extra Wake-Up
S4_LEASE expire
S1_LEASE expire & ET1< EC < ET2
S2_LEASE expire & ET1< EC < ET2
S3_LEASE expire & ET1< EC < ET2

Figure 17: State diagram of the duty-cycling adaptation scheme

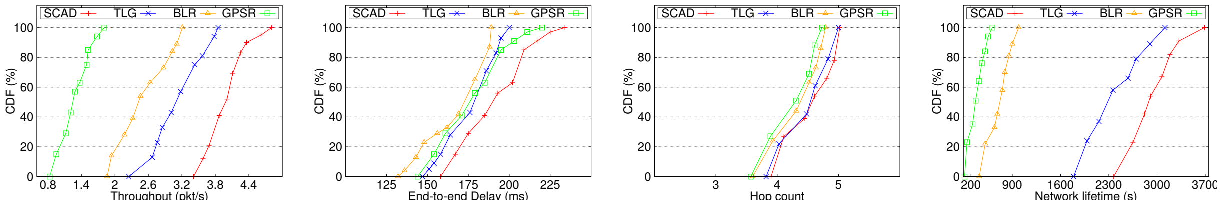


Figure 18: CDF of (a) Throughput; (b) End-to-end Delay; (c) Hop-count; (d) Network lifetime in the ***static topology***

energy consumption. Therefore, the node transits to state S_5, where the wake-up interval is extended to four times of T_B.

3 Real-world Experiments and Evaluation

To evaluate SCAD, we conducted experiments using our indoor testbed. We used five source nodes, 14 intermediate nodes, and one destination node. We choose the geographic routing protocol GPSR, the beaconless protocol BLR, and our work of TLG as the baseline protocols. The results of cumulative distribution functions (CDF) of throughput, end-to-end delay, network lifetime, and hop-count are collected as performance metrics. Experiment results show that our protocol produces the best results in terms of network lifetime and system throughput, which are due to the integration of context-aware opportunistic routing with adaptive duty-cycle operations of sensor nodes.

References

[1] Z. Zhao, and T. Braun, "Real-World Evaluation of Sensor Context-aware Adaptive Duty-cycled Opportunistic Routing," *Proceeding of 39th Annual IEEE Conference on Local Computer Networks (LCN 2014)*, September, Edmonton, Canada.