

A proposal of an infrastructure for load-balancing transactions on electronic funds transfer systems

Alexandre Luis de Andrade, Tiago André Jost,
Cristiano André da Costa, Rodrigo da Rosa Righi

Universidade do Vale do Rio dos Sinos. Av. Unisinos, 950, Cristo Rei, 93022-000, São Leopoldo, RS, Brazil
alexandreloisandrade@gmail.com, tiagojost@gmail.com, cac@unisinos.br, rrrighi@unisinos.br

Abstract. This article aims to present the first ideas for developing a framework for load-balancing called GetLB. Considering the electronic funds transfer (EFT) context, GetLB offers a new scheduling heuristic that optimizes the selection of Processing Machines to execute transactions in a processing center. Instead of using the Round-Robin typical approach, the proposal combines data from computation, network, memory and disc metrics for producing a unified scheduling approach, denoted LL (i,j). The proposal calculates the load level of executing an i-typed transaction on a j specific Processing Machine. Furthermore, the load-balancing framework also enables notifications triggered by Processing Machines to the Dispatcher for informing it about asynchronous events such as administrative tasks or transactions disposing. Aiming to evaluate GetLB, a simple prototype was developed by using Java RMI. Preliminary tests revealed that the framework is feasible, outperforming the number of queued transactions obtained with the Round-Robin approach.

Keywords: Electronic Funds Transfer, transactions, load balancing, remote method invocation

Introduction

Routing approaches and efficient dispatch of requests are fundamental elements in electronic transactions systems (Araujo *et al.*, 2009; Liu *et al.*, 2010). Usually, an electronic transaction is related to either a purchase or balance requisition, and runs through a round trip path from one terminal up to a processing center. POS (Point of Sale), EFT (Electronic Funds Transfer), ATM (Automatic Teller Machine) and mobile devices are examples of the most used terminal devices.

A typical transaction-processing center supports different types of transactions. It is possible to quote those related to credit and debit cards, prepaid telephony, deposit and withdrawal. Each type has its own CPU and IO (memory and network) requirements, as well as access time to different databases. In addition, each one may use specific subsystems within the processing center. Figure 1 illustrates a common organization of a processing center for incoming transactions. The architecture presents a switch that acts as a central

point for receiving transactions. Its function concerns the dispatching of transactions to be executed on processing machines, or PM. The output of the system in Figure 1 refers to target companies for each transaction.

Each electronic transaction comprises the steps of requesting, replying and confirming (Araujo *et al.*, 2009; VISA, 2012). The main objectives in the management of requests can be summarized as follows: (i) high-performance on transactions processing with lower computational costs, and (ii) high availability to avoid loss of transactions. Both objectives go through an efficient scheduling of transactions made by the switching element as well as the analysis of the network scalability for data processing. However, the most common scheduling approach on processing centers comprises the use of the so-called Round-Robin technique (Rojas-Cessa and Lin, 2004). It operates with a circular list for mapping consumers to producers. It is the best choice when working with static homogeneous systems in both levels of transactions and resources. Simplicity to implement and quickly decision making are the

main reasons for adopting Round-Robin.

Since the transactions are heterogeneous, Round-Robin algorithm can distribute them for processing on highly loaded machines, leaving others with moderate load or idle. In addition, this strategy restricts the use of computational resources with specialized features, such as those that have hardware assisted cryptography or decoding image capabilities. Besides the use of Round-Robin, we can observe that a common organization of a processing center includes all processing machines and internal subsystems in the same local area network. However, an analysis of a company's growth could lead to use of regional subsystems, each one with potentially heterogeneous processing machines. Moreover, some countries have security rules that impose that the transaction processing system must be located in national territory. Therefore, a company can act to decentralize the processing machines across multiple domains. Such action represents a way to join different countries in a global transaction electronic system.

Considering the context aforementioned, this article presents a proposal for load balancing framework called GetLB. It acts as an alternative to use the RR method and presents the following features: (i) efficient interaction between the switch (Dispatcher) and MP machines for collecting scheduling data; (ii) scheduling algorithm that takes into considera-

tion transactions and Processing Machine data. Furthermore, our framework allows the employment of heterogeneous machines spread in different Internet domains efficiently. Finally, a prototype was implemented in Java RMI and experimental tests showed encouraging results. Especially, GetLB can improve the distribution of transactions if compared to Round-Robin when the environment is composed by highly and moderated-loaded machines.

This article is organized as follows. The next section presents related work. The section entitled "GetLB" describes the proposed framework. In addition, this section presents the scheduling method for optimizing the distribution of transactions. In the sequence, next sections present a prototype and a discussion about the tests and the results. Finally, the conclusion is described in the final section. It addresses the most important issues relating to technical and scientific contributions of the work.

Related work

Nowadays, we can observe that the use of electronic medias for payment is increasingly adopted, instead of employing money in currency paper or check directly (Vines *et al.*, 2011; Xiaojing *et al.*, 2012). Besides the convenience for consumers, the use of electronic cards benefits trade institutions and makes

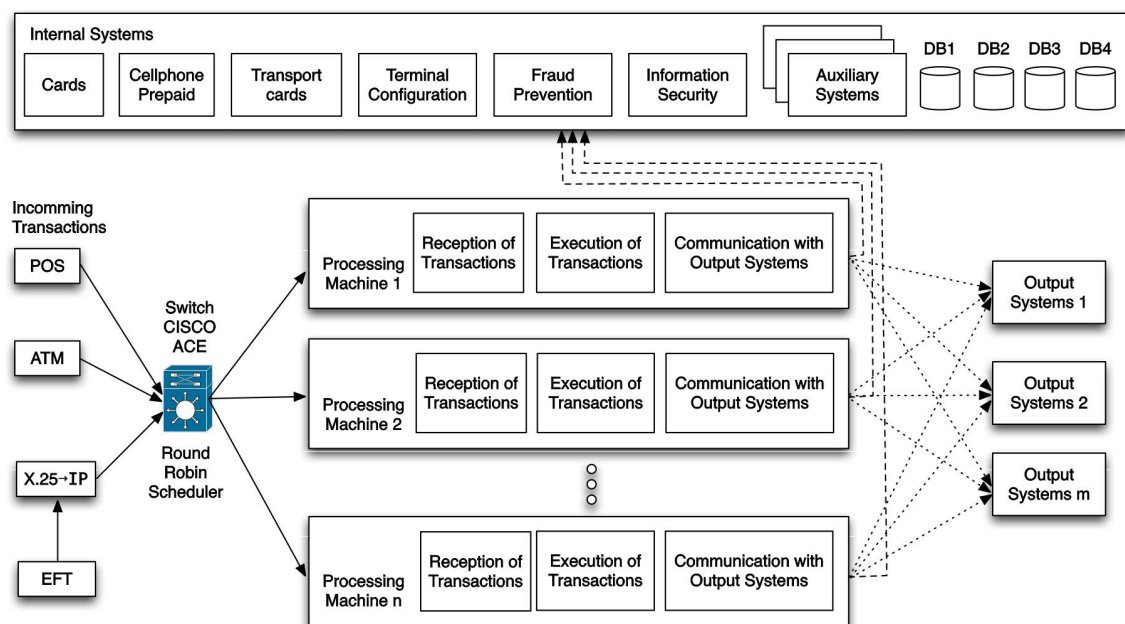


Figure 1. Typical infrastructure for processing electronic funds transfer.

the access to applications and services on the Internet easier. Virnes *et al.* (2011) claim that this transition appears both in banks and e-commerce systems, and also in e-governance, entertainment, healthcare systems and mobile devices areas. One of the most studied topics in electronic transactions systems considers the security of information (Sastre *et al.*, 2006; Seltsikas *et al.*, 2011; Vishik *et al.*, 2011). Vishik *et al.* (2011) states that both secure data transmission and trust relation should be reviewed as smartphones and embedded systems are more and more representative for generating transactions. In particular, Sastre *et al.* (2006) discuss security algorithms optimized to meet different modes of transmission, such as ADSL and GPRS.

Sousa *et al.* (2009) present a stochastic model for performance evaluation and resource planning systems for Electronic Funds Transfer (EFT). These authors developed a study of the performance by considering characteristics of dependability such as availability, reliability, scalability and security. They state that an analysis of an EFT system without these criteria may lead to inaccurate results. Furthermore, Sousa *et al.* (2009) report that the criteria shown before, should guide the efficient use of resources in order to maintain the Service Level Agreement (SLA) with customers. Araujo *et al.* (2009) claim that the analysis of performance should always observe the worst-case volume transactions arrival to be credible with the business reality of processing centers. For that, the authors adopted Petri Nets and use the access time and disk storage data, besides the arriving transactional volume.

A formal performance analysis is applied on different segments of parallel and distributed computing. Desnoyers *et al.* (2012) developed a system called Modellus, which allows modeling the use of datacenters around the Internet automatically. Modellus uses queuing theory to derive predictive models of resource usage by applications. Its differential approach is focused on combining data from several applications to infer the state of a data-center. Along the same lines, the queuing theory is also applied to wireless sensor networks in Samiullah *et al.* (2012). Other work includes the evaluation of strategies for scheduling in computational grids using traces of real workloads (Mehmood Shah *et al.*, 2010). Formal analysis allows observing the efficiency of the algorithms and the average waiting time for the completion of each job.

The analysis of performance prediction of an environment comprises the transactions' arrival rate definition. For this purpose, samples are collected at t intervals and will guide the rate calculation above. The article written by Tchrakian *et al.* (2012) presents a prediction of the flow of vehicular traffic based on time series. For that, they use a data collection interval of 15 minutes. The same step cannot be applied to an EFT system, since it can disregard a given peak of arrival of transactions.

Figure 1 illustrated a common framework for processing different types of transactions (Liu *et al.*, 2010). Besides this organization, there are especial systems for dealing with transactions for a single target such as VISA or Mastercard (Araujo *et al.*, 2009). They work with two superscalar computers as a bidirectional processing environment. Processing is performed on both resources, with database changes replicated to both sites. The databases are kept in-sync with each other so either site can take over processing if the other site fails. This means that in the event a computer is disabled, only a portion of the network is momentarily down.

Scheduling is a so-called problem in distributed systems. Besides being employed on transactions processing centers, in cloud computing providers we must use scheduling for mapping virtual machines to physical nodes. Scalability, energy saving and elasticity depend on an efficient scheduling policy on cloud environments (Sotomayor *et al.*, 2009). In this context, cloud providers, such as Amazon EC2, Nimbus, Eucalyptus and OpenNebula, present the Round-Robin scheduler for assigning virtual machines (Li, 2009). Particularly, the last three systems work with this technique as a default for their cloud environments.

In Liu *et al.* (2010), the authors affirm that there are three major trends when dealing with workload management: (i) use of Big Data systems such as Apache Hadoop; (ii) cloud computing for high performance computing and; (iii) knowledge of the application in order to get better decisions by the scheduling brokers. Especially, we can analyze the impact of this last sentence on the electronic funds transfer context. Instead of using a plain Round-Robin approach, we plan to combine I/O (network, memory and disc) and CPU data to offer a heuristic scheduling that maps transactions on heterogeneous platforms.

GetLB: Load balancing framework for EFT systems

The traditional infrastructure of EFT processing centers consists in a single scheduler that dispatches transactions in accordance with the Round-Robin approach. Furthermore, this architecture comprises a unique local area network in which Processing Machines are homogeneous among themselves. Concerning this scenario, we developed a new infrastructure for processing centers in which applies a different treatment over both the network and Processing Machines. The main idea is to support a heterogeneous platform in an efficient way by providing an optimized scheduler instead of a Round-Robin one. Basically, the proposed infrastructure was developed with the following design decisions:

- The dispatching module must work with up to date information regarding the Processing Machines for scheduling calculus;
- The scheduling of transactions must combine relevant data in order to compose a unified metric for the notion of load;
- Processing Machines must be capable to notify the switch when occurring asynchronous events in which will impact on scheduling decisions;

- The framework must deal with heterogeneous resources. This issue considers both communication and computing capabilities of the system;
- The communication comprises the message passing between the switch and Processing Machines as well as between Processing Machines and the internal systems. In addition, each Processing Machine can present its own CPU clock and latency for I/O (Input/Output) operations (such as memory, disc and network).

The first aforementioned item refers to the interaction between the nodes inside a processing center. Different from the traditional approach, Processing Machines update their own data by passing messages to the switching module periodically. Concurrently, this last entity can receive transactions and use the most recent data from machines for mapping decisions. This data refers to the transactions queue state as well as information about the CPU clock, CPU load and latencies for I/O operations on Processing Machines.

Besides the fact of considering heterogeneous Processing Machines, they can be located in different Internet domains in which are accessible by the single switching module. Thus, machines linked directly to the switch will present a lower latency if compared to those

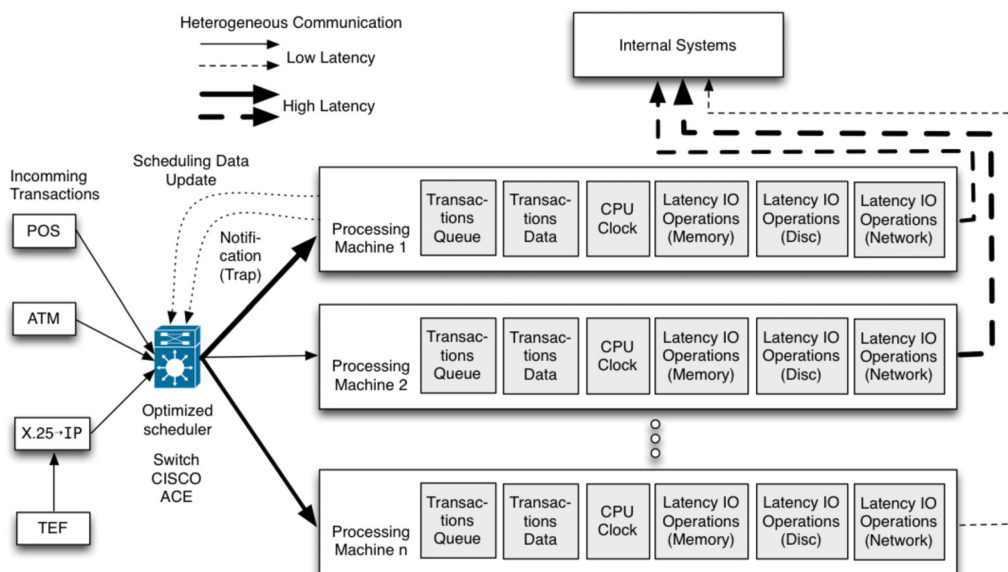


Figure 2. GetLB infrastructure in which Processing Machines can be heterogeneous among themselves. Moreover, they may be located in different Internet domains, resulting in different network latencies for contacting the scheduling machine (Switch Cisco).

installed in different sites. In this last case, a penalty to access Internet resource will be paid and this time will be susceptible to network congestion. This organization enables companies to extend their processing center to different cities or countries. As mentioned earlier, some countries such as Chile (South America) have especial rules in which all electronic transactions must be processed by machines located in national territory.

Considering that both Processing Machines and the transactions may be modeled as a heterogeneous system, we developed a scheduling heuristic called LL (*Load Level*). LL can be viewed as a decision function $LL(i, j)$ with two input parameters: i means a specific type of transaction while j denotes a target MP machine for receiving the transaction i . Considering this, the switch will calculate n equations $LL(i, j)$ for each new transaction i , where n means the number of Processing Machines. In this way, the lowest result will inform the machine that will receive a specific transaction. $LL(i, j)$ can be obtained by calculating Equation (1).

$LL(i, j)$ can be obtained by calculating the time required for receiving and processing of a given transaction i in a target machine j . The term $Receive(i, j)$ considers the time required to transmit all bytes of transaction i from the switch module up to the processing machine. Considering this, Equation (2) takes into consideration the number of bytes to be transmitted through the network as well as the time to send 1 byte between the switch and the target machine. Naturally, Equation (2) can be the most onerous part of the scheduling calculus, since the latency on WAN networks is much larger than LAN ones.

$Processing(i, j)$ corresponds to processing time for all transactions mapped to machine j , including the candidate transaction i . In this way, Equation (3) can be divided in two sub-elements: (i) a prediction of computing transaction i on Processing Machine j and; (ii) the processing time for computing all previously mapped transactions present on j transactions' queue.

Equation (4) represents a prediction of computing transaction i over the Processing Machine j . Each transaction presents the following data: (i) number of instructions, in which can be captured by using $n_inst()$ function; (ii) a vector that informs the internal systems that will be needed for transaction processing; (iii) the numbers of I/O operations considering

$$LL(i, j) = Receive(i, j) + Processing(i, j) \quad (1)$$

$$Receive(i, j) = n_bytes(i) \cdot time_to_transfer_byte(j) \quad (2)$$

$$Processing(i, j) = transaction_time(i, j) + \sum_{z=0}^m transaction_time(z, j) \quad (3)$$

$$\begin{aligned} & transaction_time(i, j) \\ &= \frac{n_inst(i)}{freq_clock(j)} + \sum_{z=0}^m (t_aces(z, j) \\ &+ t_srv(z, j) + HDio(i) \cdot t_operHD(j) \\ &+ RAMio(i) \cdot t_operRAM(j)) \end{aligned} \quad (4)$$

both disc (HD) and main memory (RAM) devices. In the same way, each Processing Machine know its own time to access internal systems and the service time of each sub-module inside the internal systems. Considering that transaction i must access at least one module inside the internal systems, the second element of Equation (4) considers both the time to access and the service time of each module. Finally, a time estimation of I/O operations completes the $transaction_time(i, j)$ function.

Implementation

GetLB consists of a switch that receives the transactions, which are originated by terminals. A common device for this role concerns a Cisco ACE 65000 switch (CISCO, 2012), which concentrates and distributes transactions between MP machines. Aiming to create a GetLB prototype system, we are employing a machine to perform the role of the ACE switch and five MP machines.

The idea of the framework concerns the creation of remote objects in both ACE and MP machines. Figure 3 illustrates the GetLB implementation with RMI (Remote Method Invocation). The remote objects located in the ACE machine hold information about each MP machine. The idea of this approach is centered on efficiency, since the scheduling calculation does not need to capture up to date information across the network. In the other side, each MP machine creates a remote object to handle the queuing system for transaction processing. In this sense, ACE has proxies for each remote object in the MP machines. ACE decides which of them should receive a new transaction.

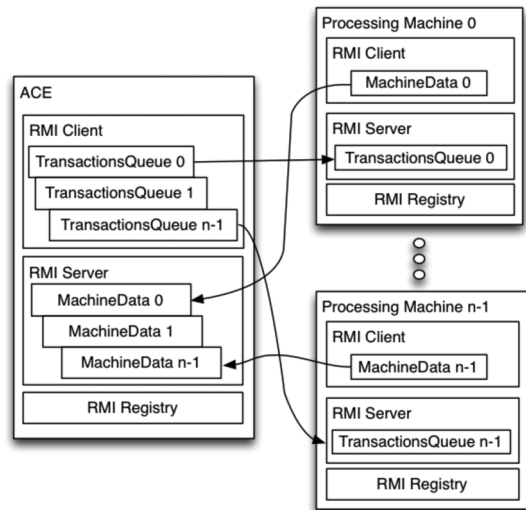


Figure 3. Prototype implementation with Java RMI.

ACE has a collection of remote objects for storing information such as CPU, disk, memory, and network time to access a particular Processing Machine. ACE machine also has a vector of proxies (stubs) for putting transactions in a specific Processing Machine queue. The number of elements in this vector is equal to the amount of Processing Machines.

Each Processing Machine must create a new thread called MachineThread. This thread is launched in the constructor of the class that represents a Processing Machine entity. The thread has the role of both collecting updated data periodically and calling the update methods over the remote object on ACE machine.

Preliminary results

Besides the proposed framework prototype with RMI, we implemented another system, which uses the Round-Robin approach in Java as well. Both GetLB and traditional approaches are illustrated in Figure 4 and 5, respectively. GetLB uses LL (Load Level) optimized scheduler rather than Round-Robin one. The tests topology in both systems consists in a switching entity (dispatcher) that receives transactions and sends them to homogeneous Processing Machines. All Processing Machines were placed in the same datacenter.

As we intended to test the GetLB efficiency and not its performance, the incoming transaction's frequency adopted for this test was 1 TPS (transactions per second). Credit card transactions were chosen for the test. This

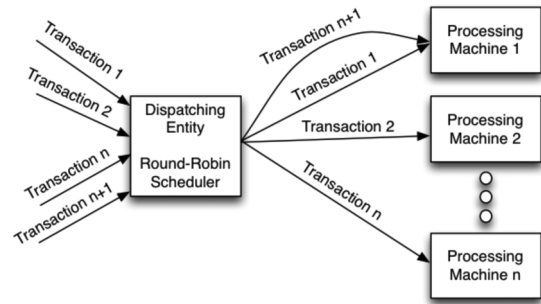


Figure 4. Topology for balancing transactions with the Round-Robin approach.

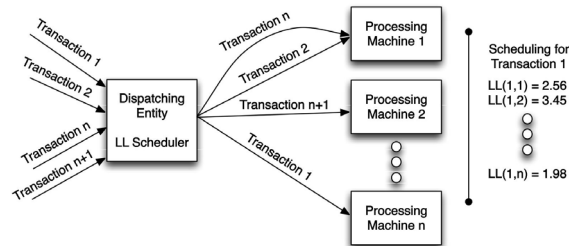


Figure 5. Topology for GetLB balancing transactions with the LL scheduler.

kind of transaction occupies, in average, 200 bytes in memory. As can be seen in Figure 6, the numbers of transactions being serviced in three Processing Machines are in equilibrium until a given instant at which occurs a situation of abnormality in one of them. The RR method does not consider the problems over a specific machine and continues to distribute transactions for it. Clearly, the affected machine will act as a bottleneck. Transactions on its queue could not be completed inside the timeout for transaction processing. The system follows this trend until this machine suffers a crash and becomes unavailable. A failure in a machine can result in a higher demand over the other machines. This process can lead to a cascade effect, causing the system to crash, becoming totally unavailable.

Figure 7 presents the results for the same situation seen above, but now applying the GetLB infrastructure and LL scheduling method. With this configuration, the same problem of balancing machines on ACE is identified by MachineData structure through the asynchronous notifications from Processing Machines to the scheduler. At this time, the distribution problem is stopped, as the MachineData structure was updated, so only viable machines are used for transaction scheduling. By this proce-

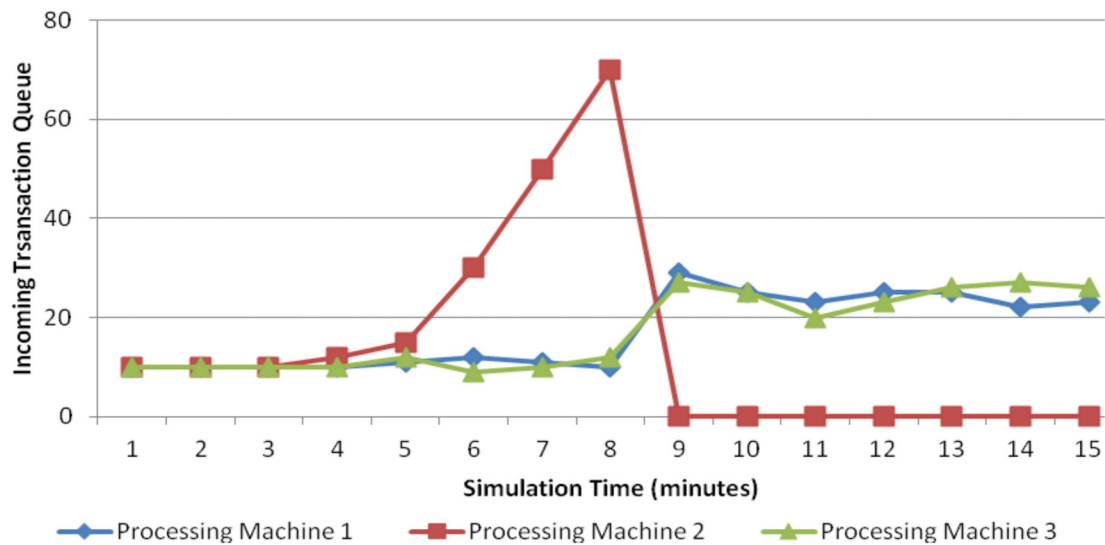


Figure 6. Round-Robin load balance allows switch to send transactions to a machine with problem until it crashes. Then, the amount of transactions in other machines is increased.

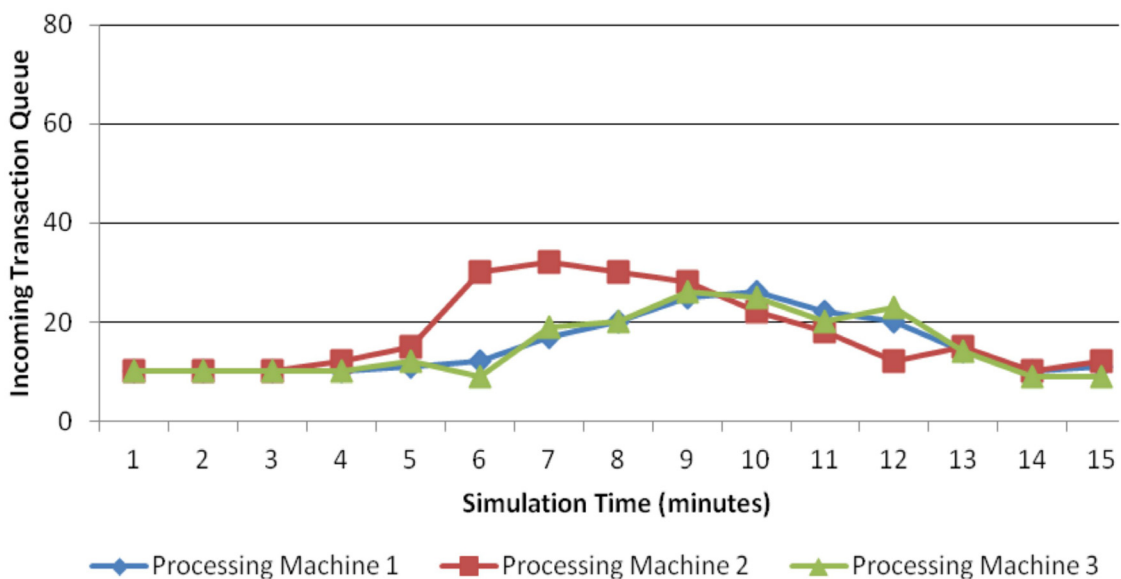


Figure 7. The scheduler algorithm with LL prevents the crash, because the switch is notified about the processing problem in Processing Machine 2, reducing the load until conditions get to normal.

ture, the scheduler avoids the crash problem as seen at the RR simulation. The scheduler continues not sending transactions to the problematic processing machine until it returns to his normal conditions. As observed in Figure 4, when the Processing Machine 2 is able to process transactions again, the load returns being equally distributed among the three machines. Indeed, by using GetLB not only a single machine is prevented of crash, but also all the system is preserved from a possible collapse.

Conclusion

This article presented the preliminary ideas of the GetLB infrastructure for transactions processing. The experimental assessments showed satisfactory results, since the load was better distributed among the processing machines. GetLB offers a new scheduling algorithm called LL (Load Level). LL can deal with both transactions and Processing Machine data for assigning transactions in a better way. It outper-

forms the Round Robin action since LL can map transactions in a arbitrary order in accordance with $LL(i,j)$ index, where i means a particular kind of transaction (with its own requirements) and j represents a target processing machine.

A prototype was developed in Java with the RMI middleware. RMI is known as one of the most onerous systems in JVM owing to reflection and serializations costly actions. Although RMI could not be the best solution for electronic transactions systems, it was useful to test the GetLB ideas feasibility. Future work comprises the development of new prototypes with UDP Sockets directly or by using SNMP.

Acknowledgments

This work was partially supported by the following Brazilian agencies: CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) and FAPERGS (*Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul*).

References

- ARAUJO, C.; SOUSA, E.; MACIEL, P.; CHICOUT, F.; ANDRADE, E. 2009. Performance modeling for evaluation and planning of electronic funds transfer systems with bursty arrival traffic. *In: INTERNATIONAL CONFERENCE ON INTENSIVE APPLICATIONS AND SERVICES*, 1, Valencia, 2009. *Proceedings...* p. 65-70. <http://dx.doi.org/10.1109/INTENSIVE.2009.17>
- CISCO. 2012. ACE Client and Servers Hitting the Same VIP. [S.l.], Cisco Systems Inc. Available at: http://www.cisco.com/en/US/prod/collateral/modules/ps2797/ps5138/product_data_sheet09186a00800ff916_ps708_Products_Data_Sheet.htm. Accessed on: 04/12/2012.
- DESNOYERS, P.; WOOD, T.; SHENOY, P.; SINGH, R.; PATIL, S.; VIN, H. 2012. Modelling: automated modeling of complex internet data center applications. *ACM Transactions on the Web*, 6(2):1-29. <http://dx.doi.org/10.1145/2180861.2180865>
- LIU, L.; SONG, M.; LUO, X.; BAI, H.; WANG, S.; SONG, J. 2010. An implementation of the on-line-payment platform based on SaaS. *In: SYMPOSIUM ON WEB SOCIETY (SWS)*, 2, Beijing, 2010. *Proceedings...* p. 658-662.
- LI, L. 2009. An optimistic differentiated service job scheduling system for cloud computing service users and providers. *In: THIRD INTERNATIONAL CONFERENCE ON MULTIMEDIA AND UBIQUITOUS ENGINEERING*, Qingdao, 2009. *Proceedings...* p. 295-299. <http://dx.doi.org/10.1109/MUE.2009.58>
- MEHMOOD SHAH, S.N.; MAHMOOD, A.K.B.; OXLEY, A. 2010. Analysis and evaluation of grid scheduling algorithms using real workload traces. *In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF EMERGENT DIGITAL ECOSYSTEMS*, New York, 2010. *Proceedings...* New York, ACM, p. 234-239.
- ROJAS-CESSA, R.; LIN, C. 2004. Frame occupancy-based round-robin matching scheme for input-queued packet switches. *In: GLOBAL TELECOMMUNICATIONS CONFERENCE*, 4, Dallas, 2004. *Proceedings...* IEEE, 3:1845-1849. <http://dx.doi.org/10.1109/GLOCOM.2004.1378307>
- SAMIULLAH, M.; ABDULLAH, S.; IMAMUL HOQ BAPPI, A.; ANWAR, S. 2012. Queue management based congestion control in wireless body sensor network. *In: INTERNATIONAL CONFERENCE ON INFORMATICS, ELECTRONICS VISION (ICIEV)*, 4, Dallas, 2012. *Proceedings...* p. 493-496.
- SASTRE, R.; BASCON, S.; HERRERO, F. 2006. New electronic funds transfer services over IP. *In: ELECTROTECHNICAL CONFERENCE*, 6, Malaga, 2006. *Proceedings...* IEEE, p. 733-736.
- SELTSIKAS, P.; MARSH, G.; FRAZIER-MCELVEEN, M.; SMEDINGHOFF, T.J. Secure government in cyberspace? *In: ANNUAL INTERNATIONAL DIGITAL GOVERNMENT RESEARCH CONFERENCE: DIGITAL GOVERNMENT INNOVATION IN CHALLENGING TIMES*, 12, New York, 2011. *Proceedings...* New York, ACM, p. 359-361.
- SOTOMAYOR, R.S.; MONTERO, I.M.; FOSTER, I. 2009. Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing*, 13(5):14-22. <http://dx.doi.org/10.1109/MIC.2009.119>
- SOUSA, E.; MACIEL, P.; ARAUJO, C.; CHICOUT, F. 2009. Performability evaluation of EFT systems for SLA assurance. *In: IEEE INTERNATIONAL SYMPOSIUM ON PARALLEL DISTRIBUTED PROCESSING*, 09, Rome, 2009. *Proceedings...* p. 1-58
- TCHRAKIAN, T.; BASU, B.; O'MAHONY, M. 2012. Real-Time Traffic Flow Forecasting Using Spectral Analysis. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):519-526. <http://dx.doi.org/10.1109/TITS.2011.2174634>
- VIRNES, J.; BLYTHE, M.; DUNPHY, P.; MONK, A. 2011. Eighty something: banking for the older old. *In: BCS CONFERENCE ON HUMAN-COMPUTER INTERACTION*, 25, Swinton, 2011. *Proceedings...* Swinton, British Computer Society, p. 64-73.
- VISA. 2012. Visa International Operating Regulations Core Principles. [S.l.], Visa Inc. Available at: <http://usa.visa.com/download/merchants/visa-core-principles.pdf>. Accessed on: 05/07/2012.
- VISHIK, C.; RAJAN, A.; RAMMING, C.; GRAWROCK, D.; WALKER, J. 2011. Defining trust evidence: research directions. *In: ANNUAL WORKSHOP ON CYBER SECURITY AND INFORMATION INTELLIGENCE RESEARCH*, 7, New York, 2011. *Proceedings...* New York, ACM, p. 1-4.
- XIAOJING, L.; LIWEI, Z.; WEIQING, W. 2012. The mechanism analysis of the impact of e-commerce to the changing of economic growth mode. *In: IEEE SYMPOSIUM ON ROBOTICS AND APPLICATIONS (ISRA)*, 07, Kuala Lumpur, 2012. *Proceedings...* p. 698-700.

Submitted on November 5, 2012

Accepted on March 7, 2013