EE-(m,k)-Firm: A Method to Dynamic Service Level Management in Enterprise Environment

Bechir Alaya[1,2]
[1]IResCoMath-Lab, University of Gabes, FSG, Erriadh street 6072 Zrig Gabes, Tunisia
[2]Higher Institute of Technological Studies, University of Gabes, Tunisia

Keywords: Enterprise Management, Operations Analytics, QoS, (m,k)-Firm Model.

Abstract: Due to enterprises environment specificities, the operations management is actually a challenging problem. In this paper, we choose to using a compromise between the available resources and the quality of service (QoS) granularity. We join this compromise to a guaranteed technique in order to reach an intelligent loss of Sub-operations according to the importance of each operation. The resulting approach permits the increase of availability, performance, reliability and system dependability. The aim of our contribution is to ensure the client satisfaction by increasing the QoS while dealing with the enterprises environment characteristics. The effectiveness of what we propose is measured through a simulation study.

1 INTRODUCTION

In the last few years, Enterprise Performance Management (EPM) is became an integrated business intelligence solution that gives companies a comprehensive view of their enterprise. Many of the challenges that companies face today, which greatly influence the enterprise performance and its quality of service (QoS).

These challenges are specified by completely uncontrollable variables, that show the current operation of the system and its performances compared to the optimal performance required by managers. These variables considerably show the current operation of the system and its performances.

Enterprise modeling remains always a challenge, despite the significant advances in modeling technology. The modeling for different points of the company is necessary. Such a modeling is part of the answer to the need for integrating the production functions and specially the maintenance and QoS guarantee. The policy that we propose can be generalized and therefore applied to all the enterprise functions.

Due to the similarities between data management in RTDBS (Hamdaoui and Ramanathan 1995) (Decker, 2014) and in enterprises (Barnes et al., 2015), we propose to adapt some results obtained on the management of QoS in the RTDBS to manage the performance of companies. However, we present a model based on (m, k) -firm model (Davide et al., 2012) (Wang et al., 2004) (Cho et al., 2010) (Goossens, 2008) studded in RTDBS to take into account the congestion of systems workload in firms.

Our main objectives were to design a model that meets the performance requirements of customers and managers and provide QoS guarantees and robustness when customer requests grow rapidly or when company resources are congested.

This paper is organized as follows. Section 2 describes some work related to the management of service quality; Section 3 shows the (m, k) -frim model; Section 4 explains the different characteristics of EE- (m, k) -firm model; Section 5 describes a method for automating the processing of EE- (m, k) -firm constraints; Section 7 analyzes the simulations results and finely, Section 8 presents the Conclusion and some remarks.

2 RELATED WORK

The QoS management in enterprise environment (Li et al., 2006) (Partha et al., 2014) (Arboleda et al., 2016) is a typical problem. Recently, several studies have been based on this topic. In (Arboleda et al., 2015), the authors proposed that to reach a superior performance, it is necessary to suggest (i) the

Alaya, B.
EE-(m,k)-Firm: A Method to Dynamic Service Level Management in Enterprise Environment.
DOI: 10.5220/0006322401140122
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 1*, pages 114-122
ISBN: 978-989-758-247-9

adoption of appropriate strategic behaviors to client, to competitor and to technology and (ii) the targeting of the appropriate market segments, notably early adopters, innovators, early majority, laggards and late majority. In their proposition, the strategic behavior of corporate performance relationship is subject to the company's strategy by examining this relationship on high technology markets and considering further contribution of the appropriate target market selection. This approach provides useful orientation to business managers to the steps that they should take to augment their performances.

The authors of (Barnes et al., 2015) accentuated the effect of interpersonal factors on company's performance through the relationship quality and the intervening roles of intercompany trust. The authors justified that trust plays an instrumental role in enhancing the components of the inter-firm relationship quality. They showed that inter firm relationship quality is positively related to superior financial performance, and most of the associations between each of the interpersonal factors and inter firm trust were moderated by the importer's size and foreign supplier's origin as well as the length of the relationship and which party initiated the relationship.

In (Slater et al., 2007), the authors proposed a technique to improve the risk analysis in Enterprise Resource Planning (ERP) (Li et al., 2006) (Mehrjerdi, 2010). They aimed to obtain a more structured systematic model of the different relationships between the risk factors/effects associated with ERP projects and attain a better understanding. The major objectives of their work were to (i) allow a collaborative approach to risk analysis, (ii) help the administrators in treating and controlling project risk and (iii) help the administrators to comprise the links between the development of a relevant risk analysis strategy and the evaluation of a global risk index for each factor used.

Demand Response Management (DRM) which is a key component of the future smart gridDemand Response Management (DRM) was the subject of (Chai and Chen, 2014). In fact, the author studies DRM with different public service companies.

Depending on the requirement of enterprise, several types of information systems have been improved for various goals (Krell et al., 2016). A study in (Al-Mamary et al., 2014) attempted to demonstrate the role of each type of information systems in firms' organizations. According to O'Brien & Marakas (Brien and Marakas, 2010), the applications of information systems that are implemented in today's business world can be classified in several different ways. In enterprises world, there are varieties of information systems such as, Office Automation Systems (OAS), Expert System (ES), Transaction Processing Systems (TPS), Management Information Systems (MIS), Executive Information Systems (EIS), Decision Support System (DSS), etc. Each type of information system has a specific objective in management operations and in organizational hierarchy (Alam et al., 2015).

Other researches were proposed in (Kadiri et al., 2016) (Atkinson et al., 2015) to present customized views of enterprise systems to various stakeholders according to their competencies and requirements. For a better QoS, they were interested in developing and improving the services and languages offered by such tools on a continuous basis. They discuss the weaknesses and strengths of different approaches (Nikolow et al., 2013) interested in language development and proposed a modeling framework more able to support the main extension scenarios currently found in practice.

3 ENTERPRISE AND QoS MANAGEMENT

3.1 (m,k)-Firm Model

The recurrence of tasks in real-time systems allow to ignore some invocations (or jobs) using (m,k)-firm constraints. These constraints specify that in a window of k invocations, at least m tasks ($0 \leq m \leq k$) must respect these deadlines (West and Zhang, 2004) (Hamdaoui and Ramanathan 1995) (Cho et al., 2010). Otherwise, for k tasks, m tasks are required and (k-m) tasks are optional. In (Bernat, 1998), Bernat showed through an example why it is best to use two parameters to define this type of constraints. Furthermore, it has been shown that the concept of (m,k)-firm constraints is appropriate for specification (management) of QoS of real-time application (Wang et al., 2002). To effectively manage the tasks under (m,k)-firm constraints, new scheduling algorithms have been proposed (Hamdaoui and Ramanathan 1995) (Ramanathan, 1999). They are divided into two main groups units, (1) Dynamic algorithms (Hamdaoui and Ramanathan 1995) and (2) Static algorithms (Dixon and Verma, 2013). Briefly, the static algorithms provide a deterministic vision of the system, while the dynamic algorithms rather provide a probabilistic vision. The dynamic algorithms take into account any system modification.

3.2 (m,k)–Firm Constraints Application

The enterprises management aims to meet the client requests by integrating many constraints: the costs, products quality, deadlines, customer demand, necessary personnel, infrastructure, supply of raw materials, etc.

The enterprise management must take into account four main types of constraints (Dixon and Verma, 2013).

The main objective in enterprise environment is both quantitative and qualitative. The QoS degradation implies the degradation of system performance in such a way that the system continues to function but with a disequilibrated level of QoS. In an overload situation, the production and QoS degradation is inevitable since clients' demands will always be dropped or delayed although many clients' demands can tolerate some delay if they arrive with a permitted mode. Moreover, the effect on QoS in enterprise environment depends on how and when the degradation is present.

The proposed method can be described as follows: a task in process of industrial production is constrained by (m,k)-firm requirements if at least m task instances within a range of k consecutive tasks respect their intended deadlines. If more than (k−m) deadline of tasks fail in k consecutive tasks at that moment, we can mention that the tasks will fall in a dynamic failure state. Consequently, the QoS constraints will not be satisfactory for the customers. For each enterprise branch, the values of m and k vary according to the criticality of tasks and system load. In practice, the values indicated by the industrial systems are not all of the same importance.

3.3 Tasks Management and Adaptation

The tasks of an enterprise system are decomposed into several classes according to their tolerance to tasks loss.

We consider three classes of tasks in the industrial environment: critical task, hard non-critical task and optional task. With this technique, which we called (m,k)-firm in Enterprise Environment (EE-(m,k)-firm), we can realize a compromise between the available resources and the QoS granularity in the same type of task.

In this work, we focused on the adaptation of the number of tasks to the system load state. We assumed that measures of the system capacity were available on the one hand and that we had a significant number of client demands on the other hand.

We also assumed a system situation in a production enterprise, whose actual performance is N, was overloaded. We supposed that Optimal QoS (Opt-QoS) was the quality of the client demand necessitating M tasks. In order to be coherent with the system performance, it was requisite to throw (M−N) tasks. Consequently, we had to reduce the quality of the client demand and if necessary, we could remove some tasks. However, the removal without applying a control method would be arbitrary.

The removed tasks are lost from the system, causing QoS degradation, notably if some critical tasks (Goossens, 2008) are removed. In this work, we adapted the EE-(m,k)-firm constraints, that serve to discard some tasks but intelligently.

The three classes of tasks were proposed to adjust the QoS requested by the clients based on real system capacity. We proposed that constraints for each task category were fixed as follows: EE-(mc,kc)-firm for critical tasks, EE-(mh,kh)-firm for hard non-critical tasks and EE-(mo,ko)-firm for optional tasks. Notably, mc tasks must be executed among kc tasks.

The system capacity was calculated using the formula: mc + mh + mo, where mc and kc present the constraints of critical tasks .The constraints of different task classes are organized as follows: mc>mh>mo. In the enterprise environment, we usually propose that mc=kc, given that these types of tasks are critical and that it is not recommended to lose them.

With the application of our EE-(m,k)-firm policy, we suppose that the required capacity necessary to respond to an enterprise transaction is M. With:

M=kc + kh + ko,

N=mc + mh + mo.

We proposed how to equilibrate the QoS at the tasks level in a production enterprise according to the available system capacity. We began by calculating the required capacity by all the current clients. Then, we calculated the rate that presents the ratio between the available system capacity (N) and the required capacity.

$$Rate = \frac{N}{\sum_{i=1}^{k} DR_i} \qquad (1)$$

Given that:

- k present the number of available tasks in the system.
- DRi present the demanded resource by task i.

4 INTELLIGENT TASKS LOSS

In order to guarantee an intelligent loss of tasks according to their importance, we defined a method which describes how a client demand is composed of k tasks. These tasks follow a well-defined organization when the system resources are not available to respond to all necessary tasks to have the requested QoS.

Given the differentiation between tasks, we already proposed that the critical tasks are two categories. In one part, the mandatory and critical tasks (C1) without which the client demand will not be realizable. In the other part, the critical tasks (C2) without which, the client demand can be realizable, but the QoS will be extremely poor.

Similarly to the hard and non-critical tasks, we proposed a classification into two categories, namely H1 and H2. Finally, for optional tasks that usually reflect the QoS degree we have two classes (O1, O2) according their importance for QoS level.

Consequently, The k tasks of a client demand based on EE-(m,k)-firm constraints are represented by a succession of k elements from {C1,C2,H1,H2,O1,O2} as described previously.

In enterprise environment, it is difficult to have an approach that guarantees an optimal QoS for all clients arriving to the system. Using this intelligent loss specification, each client can show these EE-(m,k)-firm constraint according to requested QoS and the available system resources. A minimum QoS is guaranteed if at least all critical and mandatory tasks are executed. Notably, if some optional tasks are missed by the system, the degradation will be only to EE-(mo, ko)-firm constraints, but not to EE-(mc, kc)-firm constraints.

These constraints are extremely appropriate in order to extract all requirement of a client demand. In all cases, a client demand is represented by a succession of critical tasks, hard tasks and optional tasks.

The loss of tasks in a type of critical tasks and/or in a type of hard tasks necessary for a client demand will cause some degradation in the following tasks until a new demand occurs, although the optional tasks loss has no effect.

4.1 Dynamicity of EE-(m,k)–Firm Constraints

To improve performance, availability, reliability, and system dependability, we applied a method of dynamic treatment of tasks. The objective was to automate the treatment of EE-(m,k)-firm constraints.

The need of responding to critical and hard tasks is most crucial when it comes to sensitive systems where an error can be humanly costly other than financially. This is the case, for example, for nuclear reactors, chemical factory, aircraft systems and many others. Following to importance of client demand and the importance of meeting their QoS requirements, we associated a dynamic analysis modules to EE-(m,k)-firm constraints, to optimize the gap between the rate of received QoS and the rate of desired QoS. In some situations, the system stops the operation of any other tasks (critical, hard or optional) to respond to tasks that are humanly more critical.

A deviation detecting module between the provided QoS by the system and the desired QoS by the client has become necessary. Then, according to task type, a consultation of gap impact must be carried out. Finally, the system decides the necessary values of EE-(m,k)-firm constraints.

4.2 Detection and Localization Gap

The detection procedure aims to determine the occurrence of a gap between the values m and k of each task type that has a specific EE-(m,k)-firm constraints fixed by the system. Indeed, because the properties of different tasks according to their types, the difference between MC and KC is more important than the difference between MH and KH and also between MO and KO. However, this detection procedure will be applied to all possible types of constraints. Generally, for proper operation of an enterprise, these differences are usually of zero mean, which represents an optimum QoS to clients.

A means to auto-observe the gap between different EE-(m,k)-firm constraints is to estimate the needed values for each constraints type (KC, KH and KO). The estimated values of MC, MH and MO are then respectively subtracted from maximum constraints KC, KH and KO to form the gaps E(C), E(H) and E(O) as follows:

$$\begin{cases} E(C) = K_C - M_C \\ E(H) = K_H - M_H \\ E(O) = K_O - M_O \end{cases}$$

Given that KC> MC, KH> MH and KO> MO.

At production times in an enterprise, the gap E(·) will significantly deviate according to the increase of system load, it will be equal to zero except when the system operates normally. In real applications, the differences are not exactly a zero value for the systems absence that accurately reflects the actual state of resources. Besides, the assigned measurements that aim to reflect the available resources are often marked by measurement noises.

The optimal QoS for clients varies according to values of measurement noises. With the proposed treatment of EE- (m,k)-firm constraints (see figure 1),

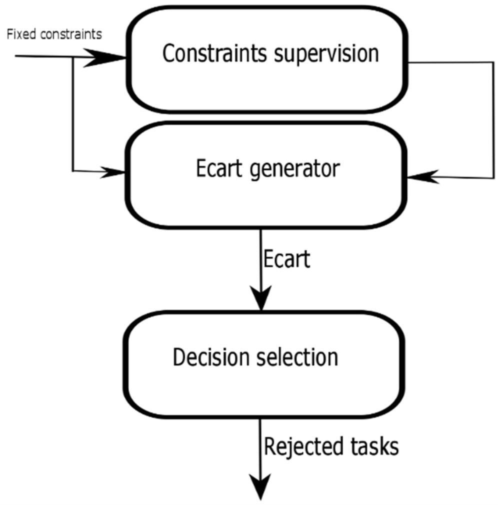

Figure 1: Treatment structure.

and depending on the criticality of supplied products (chemical, nuclear ...), the values of KC and KH must be accurately measured. The gaps will then be written as:

$$\begin{cases} E(C) = K_{mC} - M_C \\ E(H) = K_{mH} - M_H \\ E(O) = K_{mO} - M_O \end{cases}$$

Where Km (·) is the value measured by the system and which is composed by the real value K(·) and the various types of noises relating to the calculation uncertainties .

To guarantee the application of EE-(m,k)-firm constraints, we propose a comparison method of each gap E (.) at an optimal predefined threshold for each type of task: Threshold ε for critical tasks, ε' for hard tasks and ε'' for optional tasks, respectively. At every crossing of threshold, an alert is sent to the system for a new QoS management, we will then have:

$$\begin{cases} E(C) \le \varepsilon \leftrightarrow \text{Alert} = 0 \\ E(C) > \varepsilon \leftrightarrow \text{Alert} = 1 \end{cases}$$

$$\begin{cases} E(H) \le \varepsilon' \leftrightarrow \text{Alert} = 0 \\ E(H) > \varepsilon' \leftrightarrow \text{Alert} = 1 \end{cases}$$

$$\begin{cases} E(O) \le \varepsilon'' \leftrightarrow \text{Alert} = 0 \\ E(O) > \varepsilon'' \leftrightarrow \text{Alert} = 1 \end{cases}$$

After detecting the presence of a gap between M and K, it is necessary to locate the task type affected by this gap. This is nominated by the gap localization.

At the realization, we proceed at a structuring of all generated gaps during the system function. Generally, we constructed a first set of gaps Ei (·) that depend on the tasks types. From these basic gaps, we form two types of gaps: hard gap and soft gap.

In case of hard gap, after receiving an alert, the system immediately acts even by an intelligent violation of allocated resources to other clients' demands. This gives a dynamicity of resources allocation and EE-(m,k)-firm constraints. However, in case of soft gap, the system does not immediately act, but waits for the availability of resources to respond to this task type. During system function, the EE-(m,k)-firm constraints dynamically vary according to priority of client demands, system load and gap type.

We will have a decrease in optional tasks number and an increase in critical tasks number. For hard tasks, the number varies depending on the decrease and increase of critical and optional tasks.

5 SIMULATIONS AND RESULTS

We now detail the implementation of the EE-(m,k)-firm policy. Four types of decisions should be taken by our policy. We first describe the necessary data structures, and then we consider each of these decisions separately.

5.1 Description of Data Structures

Table.1 shows the data structure for each client demand. In a table noted table of demands in which each line contains the tasks number of a demand, and the class of popularity (EE-(m,k)-firm constraints), indeed, three classes are present.

The first refers to the C tasks (Critical) which are the most requested tasks by the system. The second regroups tasks of average importance H (Hard). The third contains optional L tasks (Low), i.e. least required by the system. Tasks table (table 2) records various information about the demanded tasks. Note that, the demands may not have the same number of tasks. Each entry in the units table (table 3) corresponds to a unit and maintains several counters that keep track of free and served resources.

Table 1: Demands table.

Demand-id	Requested tasks	EE-(m,k)-firm constraints		
		C	H	L
Demand_1	14	4	4	6
Demand_2	7	3	2	2
Demand_3	9	4	3	2
Demand_4	4	3	1	0

First, for load balancing on the units, the choice of the tasks will be on a lightly loaded unit that is selected for execution.

Table 2: Tasks table.

	Number of task on execution	Number of unit tasks	Unit-id			
Task_1	60	4	U1	U3	U2	U4
Task _2	70	2	U2	U3		
Task _3	50	3	U3	U1	U4	
Task _4	20	4	U2	U3	U1	U4

This is achieved by traversing the entrance of the tasks table to find all the units that contain the type of requested task (including tasks in progress) and then looking into the set of the corresponding units to find the least loaded unit. Whereas unit Ui has completed execution of a task Tj, the data structures must be updated, to indicate a resource liberation on Ui. This is done by resetting the counter in the entry of tasks table.

Table 3: Units table.

Id-Unit	Total resource (Rs)	Free Rs	Served Rs	Rs in free admission
U1	1000	100	900	10
U2	3500	700	2800	5
U3	2000	300	1700	7
U4	1500	200	1300	20

5.2 Unit and Task-type Selection

After having taken the decision to get the process of responding to a client demand, the EE-(m,k) -firm policy must select the tasks using the different constraints.

The execution begins when the system completes the selection of different tasks types of a demand. Note that the EE-(m,k)-firm policy does not change the task type simply because a resource is released by this task or because it has caused a unit overload.

This avoids the problem of changing the task type which slightly affects the QoS requested by the client. The estimated profit of Pti to execute a task i of a demand is a measure of future load that can be reduced from the current unit. This is calculated as follow:

$$Pt_i = \left(\frac{1}{t_i} - \frac{1}{t_i + 1}\right) \sum_{j=0}^{i-1} n_j \, w^{i-j-1} \qquad (2)$$

Where W represents the weighting factor. The motivation for using this formula is to change the task type where the advantage in terms of load is expected to be higher in the future.

The load that can be changed in the near future (execution task time) is given by the load on the previous task. However, the load on the previous tasks represents the load that can be changed gradually in the future. To further improve the performance of the immediate load transfer, the profit to execute a task was calculated by weighting exponentially.

Algorithm 1.

```
Ti = number of tasks of i type
Pti = Execution profit of task i
Si = unit number that can execute the
task i
Sth = threshold of tasks number
Pi = Popularity of i task
Pmin = min (Pi)
Pmax = max (Pi)
Pmoy = (Σi = 1..N Pi) / N
Class L = [Pmin (Pmin + Pmoy) / 2]
Class      H       =[(Pmin+Pmoy)/2,
(Pmax+Pmoy)/2]
Class C = [( max + Pmoy) / 2, max]
V (V1 ... ..Vj ...... .VN)
For (j = 1 to N)
If (Vi∈ H Class)
For (i = 2 to tasks number)
If (Si> Sth)
R = round ((Si-sth)/quota)
If  (Pti  superior  to  all  benefit  of
another tasks)then
Execute  the  task  i  in  the  first
selected unit
```

The load on the precedent tasks can be found from entries matching tasks in the tasks table. Also, if there are ti tasks of the current demand i, creating a modification of task type result (1/(ri -1)/(ri + 1)) of profit in terms of load movement. The current number of tasks ri will be also available in the current task entry.

5.3 Simulation Results

First, we discuss the impact of the tasks number on the system response time (access delay) with "fixed EE-(m,k)-firm constraints" and "dynamic EE-(m,k)-firm constraints". Figure 1 describes the representative results for different values of demands, such as 10, 20, 30, 40 and 50.

Figure 2 shows that the response time for all curves decreases with the increase of tasks number. The time considerably decreases between a low tasks number and an important tasks number. Indeed, the load balancing between different tasks types is significantly reduced. This is due to dynamics of tasks treatment that affects several factors. In particular, the access delay, in case of a unit overload improved with a dynamic EE-(m,k)-firm constraints, since it depends on tasks criticality that will be dynamically treated by the system. Thus, the system, which will be able to answer, has several tasks with these dynamic treatments, leading to improve management of the storage space and the QoS. This means that the EE-(m,k)-firm policy brings several benefits other than reducing the access delay.

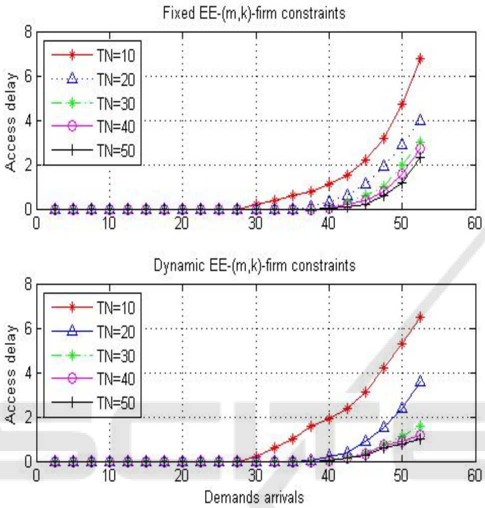

Figure 2: Access delay.

At the tasks execution, the system begins the tasks dispersion between the necessary units. The second application of EE-(m,k)-firm policy results in the correct application of dynamicity of the policy showing that mC and kC have the highest priority. The graph in figure 2 shows the behavior we expected. We can equally notice from the same figure that EE-(m,k)-firm policy, with fixed or dynamic constraints in all loads requirement, gives a shorter response time.

But, we note that when increasing the tasks number, the difference between results decreases. Consequently, we can predict that if the number of tasks attains a certain threshold, there will be no difference between the different algorithms.

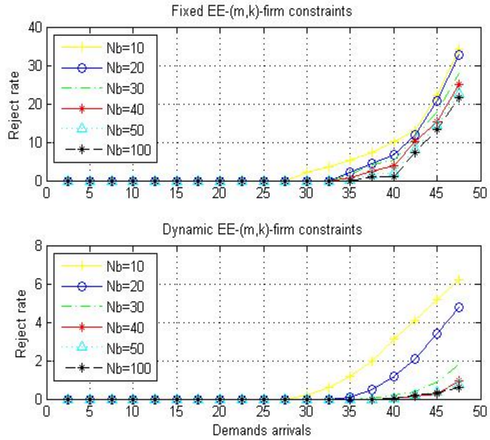

Figure 3: Reject rate.

Figure 3 shows that our policy with a dynamic treatment of EE-(m,k)-firm constraints significantly reduces the rejection rate. The difference between the curves, using fixed and dynamic constraints, shows the improvement of tasks acceptance rate. The gap between sub-curves of EE-(m,k)-firm policy with a dynamic constraints m and k on different numbers of demands, shows the effectiveness of this dynamicity on the rejection rate. The ratio between the decrease of rejection rate with the increase of demands number shows that when the tasks number increases, the curves of our policy will be confused. We can conclude from these comparisons that our proposed policy achieved the desired results, even with a large tasks number.

The served tasks rate present the ratio among the number of received and executed tasks and all requested tasks.

We describe the case of little workload arriving to the case of the best workload. With dynamic constraints of EE-(m,k)-firm policy and in case of weighty system workload, our policy substantially affects the rate of served tasks. Consequently, at different workloads, EE-(m,k)-firm policy is powerful to overcome the system congestion problems (figure 4). From this study, we notice that the EE-(m,k)-firm policy provides satisfactory results.

Outstandingly, EE-(m,k)-firm policy leads to an important number of served tasks in the case of high workload, up to 98% with dynamic constraints and about 49% with fixed constraints.

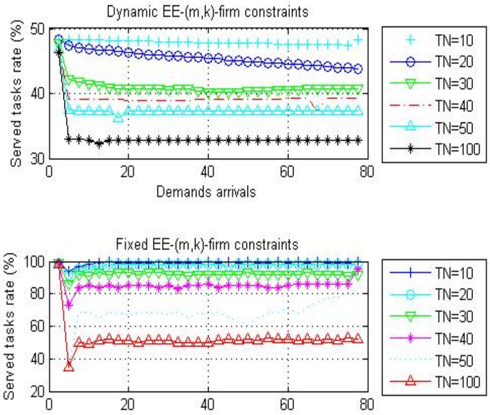

Figure 4: Served tasks rate.

6 CONCLUSIONS

The main purpose of this study was to present a novel policy of a specific treatment technique of tasks in enterprises environment. We proposed the EE-(m,k)-firm policy to indicate the necessary tasks and calculate their ranking, using a compromise between the available resources and the QoS granularity in the same task type. Based on an in-depth review of the relevant literature, three categories of tasks are possible in enterprise environment, namely critical tasks, hard tasks and optional tasks.

Afterwards, a guaranteed technique was applied to losses tasks intelligently according to importance of each task. A dynamicity of EE-(m,k)-firm constraints is then used to attain an increase of availability, performance, reliability and system dependability.

The results obtained from the proposed policy reveal that a ‘‘lack of awareness regarding the benefits of dynamic treatment of tasks in an intelligent and dynamic manner in enterprise environment is the most important reason behind the implementation of EE-(m,k)-firm policy. This type of policy can be extremely valuable for companies that wish to focus their efforts and resources to guarantee a satisfactory QoS for end-users and challenges toward the successful implementation of tasks management.

REFERENCES

Hamdaoui, M., Ramanathan, A., 1995. *"A Dynamic Priority Assignement Technique for Streams with (m,k)-firm Deadlines*", IEEE Trans. Computers, vol: 44, No: 12, pp: 1443-1451.

Decker, A., 2014. *"Factors Influencing The Quality of Accounting Information System And Its Implications on The Quality of Accounting Information"*, Research Journal of Finance and Accounting, Vol:5, No:2.

Barnes, B, R., Leonidou, L. C., Siu, N. Y.M, and Leonidou, C. N., 2015. *"Interpersonal Factors as Drivers of Quality and Performance in Western–Hong Kong Interorganizational Business Relationships"*, Journal of International Marketing, Vol: 23, No: 1, pp: 23-49.

Davide, A., Riccardo, D., and Valeria, M., 2012. *"Risk assessment in ERP projects"*, Information. Systems. Vol: 37, pp: 183-199.

Wang, Z. , Chen, J., Sun, Y., *"An integrated DBP for streams with (m,k)-firm real-time garantee"*, J. Zhejiang Univ. Sci, Vol: 5, No: 7, 2004.

Cho, H., Chung, Y., Park, D., 2010. *"Guaranteed dynamic priority assignment schemes for real-time tasks with (m,k)-firm deadlines"*, ETRI Journal, Vol: 32, No: 3, pp: 422-429.

Goossens, J., 2008. *"(m k)-firm constraints and DBP scheduling: impact of the initial k-sequence and exact schedulability test"*, In16th International Conference on Real-Time and Network Systems. Rennes, pp: 61-65.

Li, J., Song, Y., Simonot-Lion F., 2006. *"Providing real-time applications with graceful degradation of QoS and fault tolerance according to (m,k)-firm model"*, IEEE Transactions on Industrial Informatics, Vol: 2, No: 2, pp: 112-119.

Partha, P., Atighetchi, M. , Soule, N., Ishakian, V., Loyall J., Grant, R., Sinclair, A., 2014. *"Secure and QoS-Managed Information Exchange Between Enterprise and Constrained Environments"*,17th IEEE International Symposium on Object /Component/ Service-Oriented Real-Time Distributed Computing (ISORC), pp: 141-149.

Arboleda, H., Paz A., Jiménez, Tamura G., 2016. *"Development and Instrumentation of a Framework for the Generation and Management of Self-Adaptive Enterprise Applications"*, Ingeniería y Universidad, Vol: 20, num: 2, pp: 303-334.

Arboleda, H., Paz, A., Jiménez, M. and Tamura, G., 2015. *"A framework for the generation and management of self-adaptive enterprise applications"*, 10th Computing Colombian Conference (10CCC).

Slater F., Tomas G., Hult, M., Eric M., 2007. *"On the importance of matching strategic behavior and target market selection to business strategy in high-tech markets Stanley"*, Journal Academy of Marketing Science. Vol: 35, pp: 5-17.

Li, J., Song, Y., Simonot-Lion F., 2006. *"Providing real-time applications with graceful degradation of QoS and fault tolerance according to (m,k)-firm model"*, IEEE Transactions on Industrial Informatics, Vol: 2, No: 2, pp: 112-119.

Mehrjerdi, Y. Z., 2010. *"Enterprise resource planning: risk and benefit analysis"*, Business Strategy Series, Vol: 11, No: 5, pp: 308 – 324.

Chai, B., Chen, J., Y. Zhang, 2014. *"Interpersonal Factors as Drivers of Quality and Performance in Western–Hong Kong Interorganizational Business*

Relationships", Smart Grid, IEEE Transactions on, Vol:5, No: 2, pp: 722 - 731.

Krell, K., Matook, S., Rohde, F., 2016. *"The impact of legitimacy-based motives on IS adoption success: An institutional theory perspective"*, Information & Management, Vol: 53, No: 6, pp: 683–697.

Al-Mamary, Y. H. S., Shamsuddin A., and Aziati, A. H. N. 2014. *"The Role of Different Types of Information Systems In Business Organizations: A Review"*, International Journal of Research (IJR), Vol: 1, No: 7.

Brien, J.A., Marakas, G.M., 2010. *"Management information systems"*, 10th ed., byMcGraw-Hill/Irwin, a business unit of The McGraw-Hill Companies.

Alam, K. A., Ahmad, R., Akhunzada, A., Nasir, M. H. N, Khan, S. U., 2015. *"Impact analysis and change propagation in service-oriented enterprises: A systematic review"* Information Systems journal, Vol: 54, pp: 43-73.

Kadiri, S. E., Grabot, B., Thoben, K., Hribernik, K., Emmanouilidis, C., Cieminski, G. V., Kiritsis, D., 2016. *"Current trends on ICT technologies for enterprise information systems",* Computers in Industry, Vol: 79, pp: 14-33.

Atkinson, C., Gerbig, R., Fritzsche, M., 2015. *"A multi-level approach to modeling language extension in the Enterprise Systems Domain"*, Information. Systems. Vol: 54 , pp: 289-307.

Nikolow, D., Slota, R., Polak, S., Mitera, D., Pogoda, M., Winiarczyk P., Kitowski, J., 2013 *"Model of QoS Management in a Distributed Data Sharing and Archiving System"*, Procedia Computer Science, International Conference on Computational Science 'ICCS', Elsevier, pp: 100-109.

West, R., Zhang Y., 2004. *Dynamic window-constrained scheduling of real-time streams in media servers.* IEEE Trans. Comput, Vol: 53, No: 6, pp:744-759.

Bernat, G., 1998. *"Specification and Analysis of Weakly Hard Real-Time Systems"*, PhD thesis, Université de les Illes Balears.

Wang, Z., Song, Y., Poggi, E.M., Sun Y., 2002. *"Survey of Weakly-Hard Real-Time Schedule Theory and its Application"*, Distributed Computing and Applications to Business, Engineering and Science (DCABES), pp: 429-437.

Ramanathan,, P., 1999. *"Overload Management in Real-Time Control Applications Using (m,k)- Firm Guarantee"*, IEEE Trans. Parallel Distrib. Syst, vol: 10, No: 6, pp: 549-559.

Dixon, M., Verma, R., 2013. *"Sequence effects in service bundles: Implications for service design and scheduling"*, Journal of Operations Management, Vol: 31, No: 3, pp: 138-152.