Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

SCHEDULING WITH PREEMPTION FOR INCIDENT MANAGEMENT: WHEN INTERRUPTING TASKS IS NOT SUCH A BAD IDEA

Marcos D. Assunção
Victor F. Cavalcante
Maira A. de C. Gatti
Marco A. S. Netto
Claudio S. Pinhanez
Cleidson R. B. de Souza

IBM Research Brazil

ABSTRACT

Large IT service providers comprise hundreds or even thousands of system administrators to handle customers' IT infrastructure. As part of the Information Systems that support the decision making of this environment, Incident Management Systems are used and usually provide human resource assignment functionalities. However, the assignment poses several challenges, such as establishing priorities to tasks and defining when and how tasks are allocated to available system administrators. This paper describes a set of incident dispatching policies that can be used, and by using workloads from different departments of an IT service provider, this work evaluates the impact of task preemption on incident resolution and service level agreement attainment.

1 INTRODUCTION

Over the past years numerous companies have outsourced their IT infrastructure to organizations responsible for operating and managing the required IT systems—organizations that are hereafter termed as *IT service provider* or simply *IT provider*. The terms for these arrangements between organizations and IT providers are established in Service Level Agreements (SLAs). Often, failing to abide to the SLAs results in fines and penalties to the IT provider and in low customer satisfaction.

To benefit from economies of scale, providers often consolidate their IT infrastructure and services into large data centers. Although these data centers offer a range of advantages, operating and managing IT infrastructure comprises various processes which often require human intervention with hundreds or even thousands of system administrators handling customers' IT infrastructure. At the heart of the IT provider sits an organization often called *Incident Management*, comprising personnel required for bringing back the IT infrastructure to operate within the terms established in the SLAs after unexpected problems and situations. A single problem or unexpected situation arriving at the incident management system is often referred in the industry as a *ticket*.

Some of the challenges in incident management are establishing priorities to tickets and defining how tickets are allocated to available human resources. Often tickets are classified according to the level of impact on the customer's processes, or *severity* of the ticket (Bartolini and Sallé 2004). Failing to meet SLAs of high priority tickets incurs in high penalties for the IT provider, so minimizing the total number of SLAs not met for high priority tickets is regarded as very important.

For designing ticket dispatching policies for IT providers, one can get inspiration from a vast literature of task scheduling in several areas, such as operations research (Blackstone et al. 1982), distributed systems (Snell et al. 2002), and management of IT changes (Lunardi et al. 2010). While this paper provides an overview of a few well-known dispatching policies, we maintain our focus on policies that aim to prevent

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

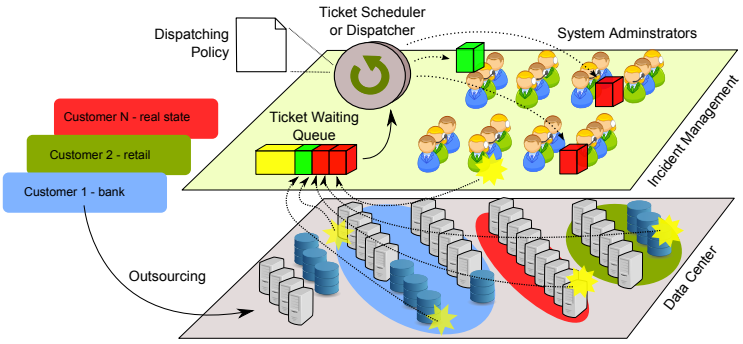


Figure 1: Illustrative view of the incident management system of an IT provider.

high-severity tickets from having their SLAs violated. In particular, we discuss overhead issues for policies with preemption support.

To investigate the effects of different ticket dispatching policies, we have developed a discrete-event simulator and used logs from a real incident management system from different departments of an IT provider as simulation input. We describe the characteristics of the logs collected from the IT provider; we introduce the simulation setup and metrics, and present results where we compare policies with and without preemption and various preemption overheads. The workload characteristics can be leveraged by researchers and practitioners working on scheduling for service providers.

Nevertheless, our main goal is to evaluate the impact on the service level objectives of allowing human resources to stop processing low severity tickets to handle high severity ones. To achieve this goal, we rely on simulations as they provide means to perform repeatable and controlled experiments without affecting the real environment. Therefore, the main contributions of this paper are:

- A detailed analysis of the impact of six ticket dispatching policies, of which two consider preemption, based on a simulation of the ticket assignment process using the real workloads;
- And an analysis, also based on computer simulations, of the impact of the resumption overhead cost on one of the best dispatching policies.

2 BACKGROUND AND PROBLEM DESCRIPTION

The large IT service provider studied in this paper operates and manages IT systems for a group of customers. The customers, who are often large corporations themselves, have stringent requirements on the performance they expect from the IT provider.

Figure 1 depicts the typical structure of the incident management system of an IT provider. As illustrated in the figure, problems can arise in the customers' outsourced IT infrastructure. These incidents consist of, for example, failures in the underlying hardware, alerts generated by management systems, or any other issue that can compromise the proper operation of IT systems. A problem results in the creation of a *ticket*, which is a task describing the symptoms and plausible causes of the problem. Tickets have different *severities*, depending on the impact on the customer's system and operations, and different expected resolution times, agreed upon by provider and customer in the SLA. In this work, we consider that tickets can have either *high* or *low* severities.

Once created, tickets are placed in a *queue* and assigned by the scheduler—a person also called *dispatcher*—to available *system administrators*, a.k.a. sysadmins. The dispatcher sorts the queue and assigns tickets to sysadmins following respectively a *sorting policy* and a *scheduling policy*, the union of which is termed as a *dispatching policy*. Currently, the IT provider under study has no integrated system that supports the implementation of the sorting and scheduling policies, due to compatibility issues among customers. However, one of the aims of our future research is to provide as automated scheduling system as possible for this investigated IT provider.

As tickets have a resolution deadline dictated by SLAs, the main goal of the dispatcher is to assign tickets to sysadmins in a way they are handled before their deadlines expire, hence ensuring that the service level promised to customers is achieved. According to the terminology used by the sysadmins, tickets should be "closed" before the SLA deadline. Although high severity tickets have a greater impact on service levels (even sometimes incurring in fines), we observed in the investigated provider that once tickets are assigned to a sysadmin, she will work on it to completion, until the ticket is "closed" even if the current ticket is a low-severity one and high-severity tickets arrive.

Based on this observation, it is important to explore possible performance improvements with regard to the number of SLA violations and preemption overhead, especially for high severity tickets, by simulating different dispatching policies using a discrete event simulator in order to compare dispatching policies in which preemption is used. In the context of the incident management system, that would mean that a sysadmin would stop working on a low severity ticket once a higher severity ticket is assigned to her.

3 DISPATCHING POLICIES

3.1 Classical Policies

First Come First Served (FCFS) is a well-known scheduling policy aimed at maintaining fairness by handling tickets as they arrive, ignoring both their severity and their deadlines. This policy assigns tickets to sysadmins in order of arrival; an approach that can be effective if deadlines, resolution times, and severities are similar for all tickets. Earliest Deadline First (EDF), on the other hand, sorts tickets in the waiting queue by deadlines. Although this policy neglects severity, it seems to be more suitable for our scenario than FCFS, and would be a strong candidate if tickets did not have different severities. High Severity First (HSF) is a policy that prioritizes tickets with high severity, disregarding any time-related information, such as arrival time or deadlines. It can be effective under workloads whose tickets have similar resolution times and deadlines, but different severities.

A straightforward improvement to EDF and HSF is the addition of preemption support, which consists in stopping a low severity ticket to make room for a high severity ticket that arrives. For EDF, every time an urgent ticket arrives, the dispatcher moves the most relaxed ticket (*i.e.*, the ticket that has the most time before it violates its SLA) to the waiting queue and assigns the urgent ticket to the now available sysadmin. The same principle could be followed in HSF, but using ticket severities as preemption criterion instead of deadlines.

3.2 Algorithm for Saving High Severity Tickets

An alternative to the classical policies mentioned above is called High Severity Earliest Deadline First (HSEDF), a policy that blends HSF and EDF. Under this policy, the waiting queue is ordered by severity, and time to reach the deadline is used as a criterion to break even in case two tickets being compared have the same severity.

Algorithm 1: Pseudo-code of the HSEDF-P policy.

```
1 Add new ticket to Waiting Queue;
2 sortWaitingQueue() /* WQ */ ;
3 sortRunningQueue() /* RQ */ ;
4 hRQ ← get head RQ;
5 hWQ ← get head WQ;
6 if possiblePreempt(hRW,hWQ) then
7     preemptTicket(hRW,hWQ);
```

Preemption being allowed, when an urgent ticket arrives a dispatcher has to decide a ticket to preempt to make room for the urgent ticket. Algorithm 1 presents the pseudo-code of the HSEDF policy with preemption, which we call HSEDF-P. As a new ticket arrives, it is added to the waiting queue (Line 1). Then the policy sorts the waiting queue (Line 2) by severity. Tickets with the same severity are sorted by the ascending order of their deadlines. The running queue is sorted in the reverse order of the waiting queue to decide which ticket is the candidate for preemption (Line 3), hence tickets with lower severity and more relaxed deadlines are at the beginning of the queue. Once the queues are sorted, both head queues are collected (Lines 4 and 5) and compared to see whether the severity of the ticket at the head of the running queue is lower than that of the ticket at the head of the waiting queue (Line 6). That being the case, preemption occurs (Line 7), *i.e.* hRQ moves to the waiting queue and hWQ is assigned to the sysadmin.

3.3 Preemption Overhead

As the presence of SLAs might have affected the processing of the tickets in the environment from which we collected our simulation data described in the next sections, we decided not to model "preemption overheads" as a function of the duration of tickets. We evaluate various scenarios where the overhead is modeled based on the work already performed by a sysadmin on a ticket. We defined HSEDF-P and HSEDF-PR as policies that support preemption with zero and maximum overhead cost respectively. Therefore, in total, six policies were evaluated, as follows in Table 1:

Table 1: Summary of the dispatching policies.

Policy	Description
FCFS	Sort tickets by their arrival time;
HSF	Sort tickets by their severity;
HSEDF	Sort tickets by their severity and deadline;
EDF	Sort tickets by their deadline;
HSEDF-P	HSEDF with preemption without overhead; and
HSEDF-PR	HSEDF restarting tickets from scratch after preemption

4 WORKLOAD CHARACTERISTICS AND EXPERIMENTAL RESULTS

4.1 Workload Characteristics

To feed our simulations, we used a workload log spanning 4 months with 17,486 tickets obtained from 2 departments (D0 and D1) of the incident management system of a large IT provider. In addition to other information, this log contains, for each ticket, an ID, its arrival time, deadline, severity, and the time a sysadmin spent to solve it. Table 2 presents ticket information per month and per department. We observe that high severity tickets account to 62% of the total number of tickets, and that the number of tickets per month/department remain similar, except for D0–Mar, which has an increase in the number of tickets processed by the system of about 30%.

To enable reproducibility of results, Figure 2 presents the main characteristics of the workload. The top histograms (Figures 2(a), 2(b), 2(c)) in this figure show the durations; deadlines according to SLAs (*i.e.*, amount of time the ticket needs to be solved before violating the SLA); and interarrival time of high severity tickets. The bottom histograms (Figures 2(d), 2(e), 2(f)) show the same information, but for low severity tickets. The histograms show that there is often a large number of short tickets, and that tickets arrive fairly frequently, which would at a first glance indicate that preemption would not be beneficial. However, if a long ticket is served before a group of short tickets, the group may be penalized by the long ticket; one of the factors that motivated us to measure the impact of preemption.

Another important characteristic of the workload is that it has many situations where tickets do not meet their deadline for resolution not because it took too long to solve the underlying problem, but because

Table 2: Summary of ticket information by month, department, and severity.

Month	Department 0						Department 1					
	Tickets	Severity				Violations	Tickets	Severity				Violations
	#	Low		High		%	#	Low		High		%
		#	%	#	%			#	%	#	%	
Dec	1535	663	43	872	57	12	2696	905	34	1791	66	5
Jan	1503	707	47	796	53	15	2838	1030	36	1808	64	4
Feb	1429	506	35	923	65	31	2691	1096	41	1595	59	5
Mar	1957	623	32	1334	68	19	2837	1171	41	1666	59	3
Total	6424	2499	39	3925	61	–	11062	4202	38	6860	62	–

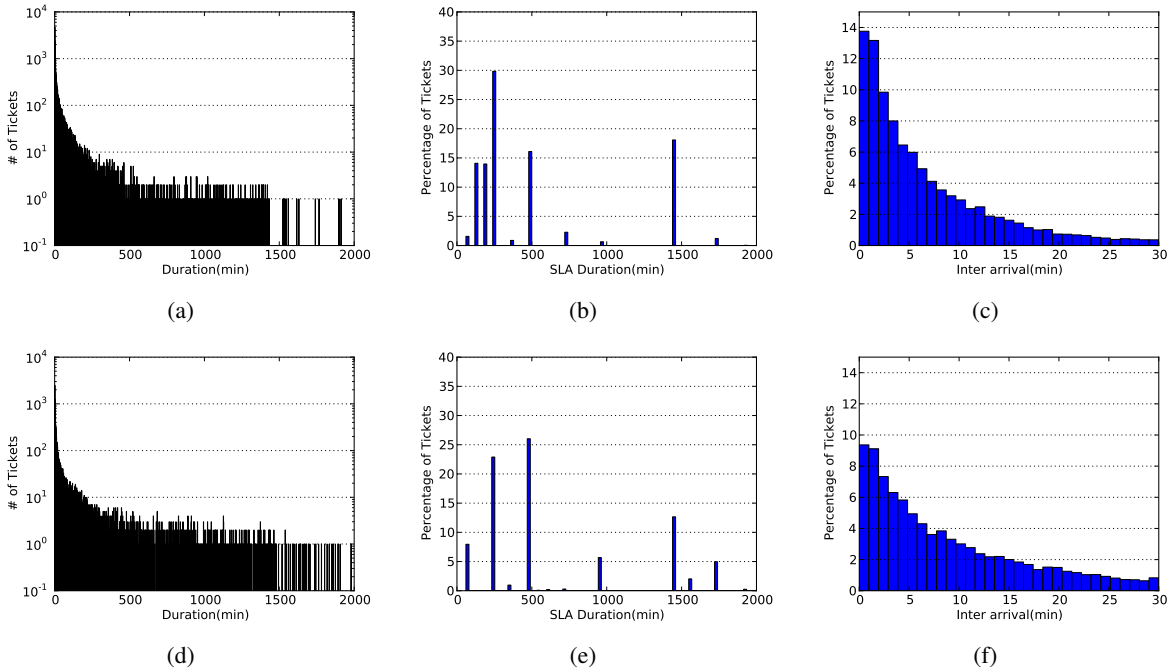


Figure 2: (a), (b), (c) – High severity tickets; (d), (e), (f) – Low severity tickets.

it took too long for a sysadmin to start working on them. This is clearly shown in Figure 3 which is a graph showing for D1 – Dec the percentage of the SLA time spent by a ticket in the queue before a sysadmin works on it, here simply called (*i.e.* Assignment Time) and by sysadmins on their processing (*i.e.* Resolution Time), normalized by the total allowable solution time. Ideally, tickets should lie well below the 100% line, but instead, various tickets are either very close to this line or well above it. The shadowed area shows the tickets that could have finished before their deadlines if the period between a ticket's arrival and its assignment to a sysadmin were minimized (*i.e.*, having more efficient dispatching). Upon investigation, we observed that long dispatching times relate to the fact that once the processing of a ticket starts, it is handled to completion, which can delay both dispatching and processing of other tickets that arrive. One would expect that by preempting low severity tickets to process high severity ones, for example, this scenario could be improved. This is another factor that instigated us to evaluate the impact of ticket preemption.

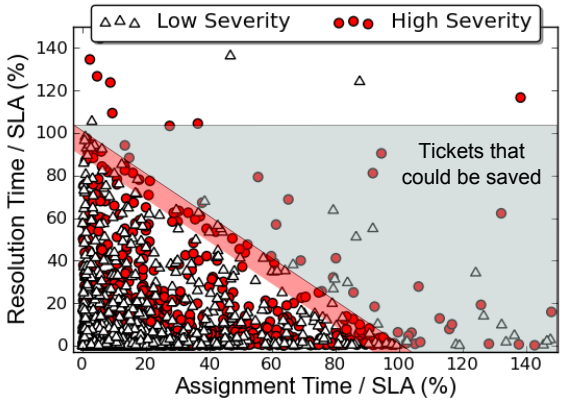


Figure 3: Percentage of the SLA period consumed on dispatching and processing (D1 – Dec).

4.2 Simulation Setup and Metrics

To evaluate different dispatching policies and the impact of preemption in IT providers, we developed a discrete-event simulation tool in Java termed as *ServeSim*. As this work is part of a larger project, we decided to implement our own simulator as existing ones do not address all aspects of such project. The simulator accepts as input: a log with the information about the tickets; and a configuration file specifying the policies to be evaluated, the departments that are modeled, how the preemption overhead is computed, and how the availability of sysadmins varies over time.

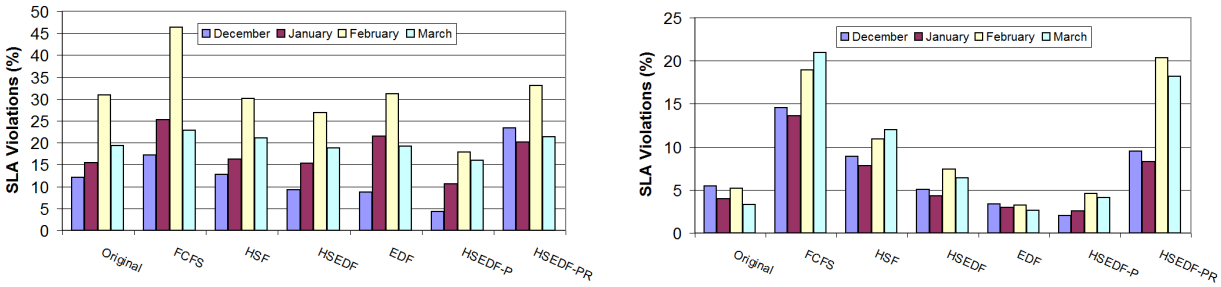
As information on the number of sysadmins working in each department and how this number varies over time was not available in the original logs, the following approach was used compute this information. For each department and month, we performed experiments with HSEDF and EDF without preemption; the two policies we believed were closest to the reality of the investigated IT provider according to our interviews with dispatchers. Then, we varied the number of sysadmins from 1 to 20 and analyzed the points where the percentage of SLA violations was similar to the original data, and took the average of the two policies as the number of available sysadmins.

In addition, analysis of work-shift information and interviews with dispatchers helped us define two availability factors σ_{week} and $\sigma_{weekend}$ that represent respectively the availability of sysadmins during week days and during weekends based on the maximum number of sysadmins working on a given department. The availability during weekdays is: 1.00, between 8am and 4pm; 0.66, between 4pm and 12am; 0.91, between 12am and 8am; whereas over weekends, the values for the same periods are 0.80, 0.52 and 0.73.

Table 3: Number of sysadmins and utilization per month.

Month	Department 0		Department 1	
	# System Administrators	Sysadmin Utilization	# System Administrators	Sysadmin Utilization
Dec	5	65%	13	46%
Jan	8	40%	14	27%
Feb	6	65%	9	51%
Mar	12	65%	9	51%

Table 3 presents the maximum number of sysadmins working in each department over the considered months. It also shows the sysadmins' utilization, which is percentage of sysadmins/hours consumed by ticket processing of the total of hours made available using the computed number of sysadmins and availability. The first set of experiments evaluates the SLA violation related metrics for all policies described in the



(a) Department 0. (b) Department 1.

Figure 4: Percentage of SLA violation under different scheduling policies (original added for comparison).

Table 4: Percentage of SLA violations for each policy and ticket severity.

Department – Month	Low Severity Tickets							High Severity Tickets						
	Ori	FCFS	HSF	HSFEDF	EDF	HSEDF-P	HSEDF-PR	Ori	FCFS	HSF	HSFEDF	EDF	HSEDF-P	HSEDF-PR
D0 – Dec	*8.6*	11.6	16.1	10.1	6.3	**4.4**	48.4	*14.7*	21.6	10.1	8.7	10.6	**4.4**	4.4
D0 – Jan	*15.7*	22.5	23.6	22.8	21.1	**18.1**	38.3	*15.3*	27.8	9.7	8.7	22.0	**4.0**	4.0
D0 – Feb	*26.3*	42.1	48.0	45.3	**32.0**	40.1	83.0	*33.5*	48.8	20.4	16.9	30.7	**5.7**	5.7
D0 – Mar	*16.1*	15.7	20.7	20.2	**14.8**	19.6	36.1	*20.8*	26.1	21.3	18.1	21.2	**14.**4	14.4
D1 – Dec	*5.7*	11.6	14.0	8.8	4.0	**3.4**	25.7	5.4	*16.1*	6.3	3.1	3.0	**1.3**	1.3
D1 – Jan	*3.8*	10.2	11.6	7.2	**2.7**	4.5	20.3	4.0	*15.6*	5.7	2.7	3.2	**1.4**	1.4
D1 – Feb	*4.5*	13.0	18.0	10.9	**2.1**	7.2	45.9	5.7	*22.9*	6.1	5.0	4.0	**2.8**	2.8
D1 – Mar	*3.8*	18.6	21.1	12.4	**2.9**	7.3	41.4	2.9	*22.6*	5.6	2.2	2.5	**1.9**	1.9

previous section, whereas the second set of experiments focuses on the overhead threshold metric for HSEDF and HSEDF-R policies. The analysis shows the results of the original data for reference purposes.

4.3 SLA Violation Analysis

Figure 4 summarizes the results on the percentage of SLA violations for the two departments (Department 0 in Figure 4(a) and Department 1 in Figure 4(b)) over four months. In these first experiments HSEDF-P uses an overhead of 0%. Taking the results of HSEDF as a baseline, one can observe that HSEDF-P leads to improvements in ticket dispatching when preemption is considered to have zero overhead. Although this may not reflect reality, the results are encouraging as preemption leads to substantial performance improvements.

SLA violation results for the various dispatching policies broken by month, department, and ticket severity are detailed in Table 4. Although EDF and HSEDF, with and without preemption, present performance improvements compared to the other policies in various scenarios, two characteristics are worth noting: the processing of low severity tickets pays a price for preemption (although not always!) and, if tickets have to be restarted from scratch (*i.e.* HSEDF-PR), preemption can have disastrous consequences in terms of SLA violation. Hence, it is important to identify the threshold for preemption overhead above which preemption stops being beneficial.

Comparing EDF and HSEDF-P, for all scenarios HSEDF-P outperforms EDF for high severity tickets, and for most scenarios where EDF outperforms HSEDF-P for low severity tickets, the benefits are not significant. This happens because although HSEDF-P prioritizes high severity tickets, by using preemption, it is also able to reorganize low priority tickets. This reorganization allows tickets with more relaxed deadlines to move back to the waiting queue and give chance to tickets with tighter deadline to be served, thus avoiding the violation of their SLAs.

Table 5 presents results on the SLA violated time for the different policies, departments, and months. Similar to SLA violations metric, EDF and HSEDF, with and without preemption, outperform the other

Table 5: Average SLA violated time, in minutes, for each policy, month, department, and ticket severity.

Department – Month	Low Severity Tickets							High Severity Tickets						
	Ori	FCFS	HSF	HSFEDF	EDF	HSEDF-P	HSEDF-PR	Ori	FCFS	HSF	HSFEDF	EDF	HSEDF-P	HSEDF-PR
D0 – Dec	*71*	29	65	23	13	**12**	1103	*128*	55	26	24	27	**17**	17
D0 – Jan	*207*	198	334	308	**173**	220	967	*179*	260	25	24	198	**18**	18
D0 – Feb	*329*	582	951	499	**275**	477	6506	*529*	594	68	60	207	**27**	27
D0 – Mar	*211*	47	112	119	**43**	105	334	*225*	98	71	62	80	**54**	54
D1 – Dec	*74*	55	108	24	**10**	15	326	*92*	70	10	6	6	**4**	4
D1 – Jan	*21*	35	50	25	**8**	18	117	*24*	47	9	5	**6**	**4**	4
D1 – Feb	*40*	39	82	35	**5**	27	662	*36*	69	8	7	7	**6**	6
D1 – Mar	*118*	147	197	46	**6**	32	562	*26*	161	10	**5**	6	**5**	5

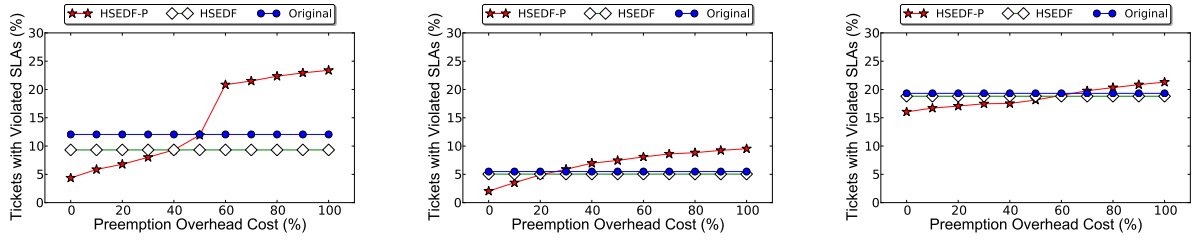
policies. However, although HSEDF-P has better performance than EDF, HSEDF-PR can lead to considerable performance degradation of processing of low severity tickets if they have to be restarted from scratch after preemption. This information corroborates the previous results that considered the percentage of tickets with SLA violations, and makes identifying the preemption overhead extremely important.

4.4 Preemption Overhead Analysis

The overhead incurred when resuming a ticket that has been preempted is computed as a percentage of the amount of time already spent on processing the ticket. We run experiments varying this overhead from 0% to 100%. An overhead of 0% means that preemption does not pose any cost whereas an overhead of 100% means that the ticket is restarted from scratch every time it is resumed after preemption.

Table 6: Overall threshold (%) for preemption overhead.

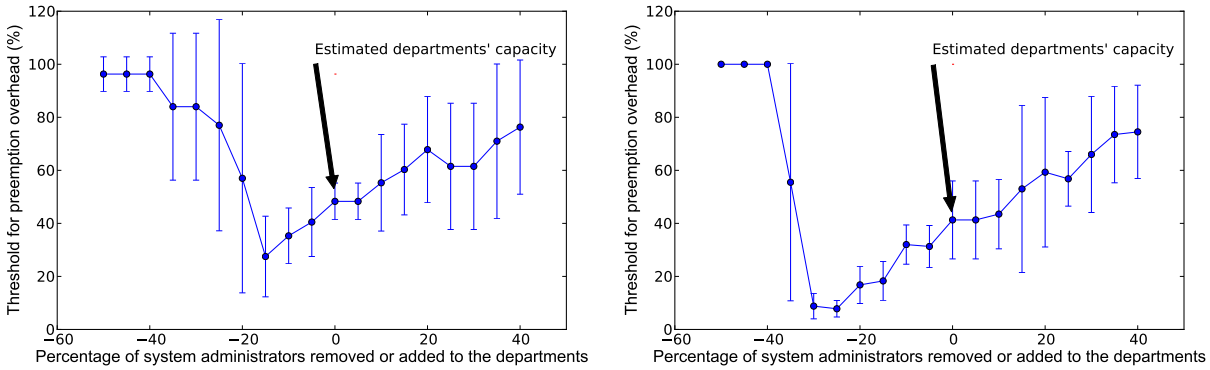
Month	Department 0			Department 1		
	All Tickets	Severity		All Tickets	Severity	
		Low	High		Low	High
Dec	40	21	100	21	12	100
Jan	44	18	100	62	41	100
Feb	51	6	100	38	31	100
Mar	58	3	100	44	41	100
Average	48 ± 07	12 ± 08	100 ± 0	41 ± 15	31 ± 12	100 ± 0



(a) Dec – D0. (b) Dec – D1. (c) Mar – D0.

Figure 5: Examples of different impact on preemption overhead.

That said, the second set of experiments evaluates the impact of different preemption overhead values on SLA violations. Table 6 shows that on average the benefits reaped from using preemption stop when the preemption overhead exceeds around 45%. This finding is illustrated in Figure 5, which shows the percentage of tickets with SLA violation under three different scenarios. The threshold here is the point where the HSEDF-P line intersects HSEDF, *i.e.* where having or not preemption provide similar results.



(a) Department 0.

(b) Department 1.

Figure 6: Sensitivity of preemption when varying the number of sysadmins.

Note that although the threshold for certain scenarios is higher, this threshold may not directly reflect the benefits the policies derive from preemption as such benefits depend on the environment conditions, such as system utilization.

4.5 Sensitivity Analysis

The next experiment has two main goals: (i) determine how sensible are the results we obtained if the number of sysadmins in the departments were different from those we estimated and (ii) how the results look like for extreme environment conditions, *i.e.* when removing or adding several sysadmins to the departments. Therefore, for each department and month, we varied the number of sysadmins available to handle tickets and identified at which point of preemption overhead using preemption stops being an advantage. Figures 6 (a) and 6 (b) show the average results for department D0 and D1, respectively. The X axis of this graph shows the percentage of sysadmins that were added to or subtracted from the number of sysadmins originally estimated for each department as described in the previous sub-section. The arrows point to the overheads with the original estimated number of sysadmins.

Figure 6 (a) shows that the average point above which preemption stops from having advantages decreases as we decrease the number of sysadmins. This happens because the number of preemptions tends to increase and so the overhead cost has a higher impact. However, if the number of sysadmins is too small to handle the load posed by ticket processing, the threshold is likely to increase. This is due to the fact that when there are few sysadmins in the system, low severity tickets are preempted at the beginning of their execution. Therefore, even if the overhead cost were very high, such overhead would be computed on short ticket executions, thus generating low impact on SLA violations due to preemption overhead. Another reason is that preemption is rare as tickets are sorted inside the waiting queue, thus sysadmins receive high severity tickets before the low severity ones. When we add sysadmins to each department/month, there tends to be an excess of personnel to handle tickets, and thus preempting tickets is less necessary. When a preemption occurs in this case, the overhead cost has a higher impact as tickets tend to spend more time with sysadmins before preemption. This also explains the high deviations when increasing the number of sysadmins. Figure 6 (b) shows a similar behavior of that described in Figure 6 (a). As a final remark, these graphs show that it pays off to have around 35-45% of preemption overhead close to the estimated number of sysadmins.

5 DISCUSSION AND RELATED WORK

Our results are aligned with prior research in Human-Computer Interaction (HCI) and Psychology that explores the effects of interruptions in the work of information knowledge workers like system administrators (Bailey et al. 2007). Findings from an observational study on interruptions in the workplace reinforce

that, in some cases, workers benefit from interruptions (O'Conaill and Frohlich 1995). Rouncefield et al. (1994) suggest that issues of cooperation and sociality may lead to interruptions that in turn improve workers' performance even in small offices. Zijlstra et al. (1999), and Adamczyk and Bailey (2004) evaluated the effect on interruptions on on task performance; and workers' emotional, psychological state, and social attribution.

Most of results of interruption in work activities for several reasons (Mark et al. 2005; Cutrell et al. 2000; Czerwinski et al. 2000; Jackson et al. 2001; Jambon 1996) focus on short-term tasks. To the best of our knowledge, there is no conclusive research on the effects of interruptions in long duration tasks.

Despite its advantages in certain aspects, task preemption has also some drawbacks. It is recognized in the literature that one of the effects of interruptions is that people do need some time to "get back on track" of the task they were performing before they were interrupted. This time, called the resumption lag, might slow down the task being performed (Bailey et al. 2000), increase the stress and anxiety (Mark et al. 2008) of the person performing the task, as well as increase the chances of error (Bailey et al. 2000).

Czerwinski et al. (2004) report a diary study of information workers' activities, and seek to characterize how people interleave multiple tasks amidst interruptions. Results show that task complexity, task duration, length of absence, number of interruptions, and task type, influence the perceived difficulty of switching tasks. Bailey et al. (2000) presented three results: (i) a user performs slower on an interrupted task than a non-interrupted task, (ii) the disruptive effect of an interruption differs as a function of task being performed, and (iii) different interruption tasks cause similar disruptive effects on task performance. Bailey presented that the degree to which the interrupted tasks performed slower when compared to tasks performed without interruption ranges from 5% to 40% of the total time of the task, which confirms our results.

From the simulation point of view for incident management context, Sheopuri et al. (2008) tackle the problem of assigning multiple severity level service requests to resources in a pool. Each severity level is associated with a due date and a penalty, which is incurred if the service request is not resolved by the due date. They proposed an Index-based policy which is a combination of there policies: First-Come-First-Serve, Weighted Shortest Expected Processing Time and Generalized Longest Queue policy. They implemented three preemption rules: (i) no preemption; (ii) partial preemption: preempt the lowest index service request that is being served that has been served for less than the mean of the service time of its severity level if a higher index service request is waiting; and (iii) full preemption: preempt the lowest index service request that is being served if a higher index service request is waiting. They do not assume there is a preemption cost.

Parvin et al. (2009) present an heuristic algorithm that assigns an allocation index to each service request that has arrived. The index incorporates factors such as variability in individuals skills, deadlines and the variability in service time. The proposed dispatching algorithm assigns a priority-based allocation index to each service request in the queue based on that index. The index is dynamically updated upon each service termination in the system. They assume non-preemptive service and also, no idling is allowed, i.e, a request can be assigned to an idle agent even though he has no skill in that service request. They compared the SLA violation penalty against the FCFS policy.

While both Sheopuri et al. (2008) and Parvin et al. (2009) have proposed a new policy for the dispatching assignment, they only compared it with regard to the Priority FCFS which is a variation of the Earliest Deadline First policy that we also evaluated in our work. We, on the other hand, performed experiments using six dispatching policies and service requests logs spanning four months collected from two IT company departments.

6 CONCLUSIONS

This work evaluated the impact of task preemption when managing incidents in IT service providers. We performed experiments using six dispatching policies and showed that sorting tickets by only their deadlines (*i.e.* EDF) affects negatively the processing of high severity tickets. Preempting low priority tickets in the presence of high priority ones and sorting them by their severity and deadlines (*i.e.* HSEDF-P), on the other

hand, reduces considerably the number of high severity tickets that miss their deadlines, without having a negative impact on the low severity tickets. This happens because, by using preemption, low severity tickets can also be rearranged by their deadlines, thus saving more tickets than a simple EDF policy. While examining the sensitivity of the preemption overhead we noticed that, up to a preemption overhead cost of about 45%, preemption is beneficial. The workloads and results presented in this paper can assist the design and implementation of scheduling policies for incident management as they provide the basis for deciding when and with what cost preemption improves service quality and client satisfaction. As ongoing work, we are deploying a preemption pilot in one of the departments of the investigated IT provider to analyze the benefits of preemption in practice.

REFERENCES

Adamczyk, P. D., and B. P. Bailey. 2004. "If not now, when?: the effects of interruption at different moments within task execution". In *SIGCHI Conference on Human factors in Computing Systems*, edited by E. Dykstra-Erickson and M. Tscheligi, CHI '04, 271–278. New York, USA: ACM.

Bailey, B., J. Konstan, and J. Carlis. 2000. "Measuring the effects of interruptions on task performance in the user interface". In *IEEE International Conference on Systems, Man, and Cybernetics*, Volume 2, 757–762 vol.2: IEEE.

Bailey, J. H., E. Kandogan, E. M. Haber, and P. P. Maglio. 2007. "Activity-based management of IT service delivery". In *1st ACM Symposium on Computer Human Interaction for Management of Information Technology (CHIMIT'07)*, edited by E. Kandogan and P. M. Jones: IEEE.

Bartolini, C., and M. Sallé. 2004. "Business Driven Prioritization of Service Incidents". In *Utility Computing*, edited by A. Sahai and F. Wu, Volume 3278 of *Lecture Notes in Computer Science*, 64–75. Springer Berlin Heidelberg.

Blackstone, J., D. Phillips, and G. Hogg. 1982. "A state-of-the-art survey of dispatching rules for manufacturing job shop operations". *International Journal of Production Research* 20 (1): 27–45.

Cutrell, E. B., M. Czerwinski, and E. Horvitz. 2000. "Effects of instant messaging interruptions on computing tasks". In *CHI '00 Extended Abstracts on Human factors in Computing Systems*, edited by T. Turner and G. Szwillus, CHI EA '00, 99–100. New York, USA: ACM.

Czerwinski, M., E. Cutrell, and E. Horvitz. 2000. "Instant Messaging and Interruption: Influence of Task Type on Performance". In *OZCHI 2000 Conference*, edited by C. Paris, N. Ozkan, S. Howard, and S. Lu, 356–361. Sydney, Australia: ACM.

Czerwinski, M., E. Horvitz, and S. Wilhite. 2004. "A Diary Study of Task Switching and Interruptions". In *SIGCHI Conference on Human factors in Computing Systems*, edited by E. Dykstra-Erickson and M. Tscheligi, CHI '04, 175–182. New York, USA: ACM.

Jackson, T., R. Dawson, and D. Wilson. 2001. "The Cost of Email Interruption". *Journal of Systems and Information Technology* 5:81–92.

Jambon, F. 1996. "Formal Modelling of Task Interruptions". In *CHI '96, Human Factors in Computing Systems*, edited by M. J. Tauber, 45–46.

Lunardi, R. C., F. G. Andreis, W. L. da Costa Cordeiro, J. A. Wickboldt, B. L. Dalmazo, R. L. dos Santos, L. A. Bianchin, L. P. Gaspary, L. Z. Granville, and C. Bartolini. 2010, Apr.. "On Strategies for Planning the Assignment of Human Resources to IT Change Activities". In *IEEE NOMS 2010*, edited by Y. Kiriha, L. Granville, and D. Medhi, 248–255. Osaka, Japan: IEEE.

Mark, G., V. M. Gonzalez, and J. Harris. 2005. "No task left behind: Examining the nature of fragmented work". In *ACM CHI 2005*, edited by A. SIGCHI, 321–330: ACM.

Mark, G., D. Gudith, and U. Klocke. 2008. "The cost of interrupted work: more speed and stress". In *26th annual SIGCHI conference on Human factors in computing systems (CHI'08)*, edited by A. SIGCHI, 107–110: ACM.

O'Conaill, B., and D. Frohlich. 1995. "Timespace in the workplace: dealing with interruptions". In *Conference companion on Human factors in computing systems*, edited by I. Katz, R. Mack, and L. Marks, CHI '95, 262–263. New York, USA: ACM.

Parvin, H., A. Bose, and M. P. V. Oyen. 2009, December. "Priority-based routing with strict deadlines and server flexibility under uncertainty". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 3181–3188. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rouncefield, M., J. A. Hughes, T. Rodden, and S. Viller. 1994. "Working with "constant interruption": CSCW and the small office". In *ACM Conference on Computer Supported Cooperative Work*, edited by J. B. Smith, F. D. Smith, and T. W. Malone, CSCW '94, 275–286. New York, USA: ACM.

Sheopuri, A., S. Zeng, and C. Dorai. 2008, December. "A new policy for the service request assignment problem with multiple severity level, due date and SLA penalty service requests". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 1661–1668. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Snell, Q., M. J. Clement, and D. B. Jackson. 2002. "Preemption Based Backfill". In *8th International Workshop on Job Scheduling Strategies for Parallel Processing*, edited by D. G. Feitelson, L. Rudolph, and U. Schwiegelshohn, JSSPP '02, 24–37. London, UK, UK: Springer-Verlag.

Zijlstra, F. R., R. A. Roe, A. B. Leonora, and I. Krediet. 1999. "Temporal factors in mental work: Effects of interrupted activities". Open access publications from maastricht university, Maastricht University.

AUTHOR BIOGRAPHIES

MARCOS DIAS DE ASSUNCAO obtained a Ph.D. in Computer Science and Software Engineering (2009) from the University of Melbourne, Australia, and a M.Sc. (2004) from the Federal University of Santa Catarina in Florianopolis, Brazil. His current topics of interest include Cloud computing, workload migration to Clouds and analytics services. His email address is marcosda@br.ibm.com.

VICTOR F CAVALCANTE did his Ph.D. in Computer Science at the State Univesity of Campinas (UNICAMP), Brazil. His main research interests include Operations Research, Combinatorial Optimization and Algorithms. His email address is victorfc@br.ibm.com.

MAIRA A. DE C. GATTI obtained her Ph.D. (2009) and M.Sc. (2006) in Software Engineering at the PUC-Rio, Pontifical Catholic University of Rio de Janeiro, Brazil. Her main area of expertise is in Computer Science and specific areas ranges from Distributed Computing to Multi-Agent-based Simulation. Her email address is mairacg@br.ibm.com.

MARCO A. S. NETTO obtained his Ph.D. at the University of Melbourne, Australia. His main research interests are Cloud Computing, scientific computing, and computer simulations. His email address is mstelmar@br.ibm.com.

CLAUDIO S. PINHANEZ is the leader of the Service Systems group of IBM Research - Brazil. He got his PhD. in 1999 from the MIT Media Laboratory and soon after joined IBM Research. His main research areas are Ubiquitous Computing, Human-Computer Interfaces, and Service Science. His email address is csantosp@br.ibm.com.

CLEIDSON R. B. DE SOUZA received his Ph.D. in Information and Computer Science in 2005 from University of California, Irvine. His interested in the intersection between software engineering and computer-supported cooperative work. His email address is cleidson@cdesouza.net.