Efficient Management of Idleness in Systems[*]

Ningfang Mi
College of William and Mary
ningfang@cs.wm.edu

Alma Riska
Seagate Research
alma.riska@seagate.com

Qi Zhang
Microsoft
qizha@microsoft.com

Evgenia Smirni
College of William and Mary
esmirni@cs.wm.edu

Erik Riedel
Seagate Research
erik.riedel@seagate.com

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Design Studies

General Terms

Performance, Reliability

Keywords

Foreground/background scheduling, storage systems

1. INTRODUCTION

As computer systems operate 24 hours a day, 7 days a week, it is becoming common to schedule maintenance jobs during idle times. These jobs are considered *background jobs*. Background jobs are used for improving system reliability, availability, and consistency, to enhancing system performance, Completion of background jobs is critical to system operation, yet their priority is not as high as that of foreground jobs, i.e., the regular jobs of the system users. In addition, scheduling of background activities should not compromise the performance of foreground jobs.

We focus on the general problem of how to bin-pack non-preemptive background jobs during system idle times. Idleness is considered as an additional system resource that needs to be effectively managed but there is a trade-off between maximizing the completions of non-preemptive background jobs while minimizing their impact on foreground performance. Previous work focuses on monitoring the performance of foreground and background jobs [1] to base background scheduling decisions. [1] proposes to idle wait, i.e., to delay scheduling of a background job during an idle interval. This delay is fixed and equal to the average demand of a background job.

Here, in addition to monitoring the performance of foreground and background jobs, measurements of the empirical distribution of idle times are also collected. Resource management of idle times is now done in a dynamic way, using statistical information not only on the foreground and background job demands, but also on the idle intervals of the system. This statistical information is collected online while the system is in operation, and is incorporated on-the-fly into scheduling policies. Detailed analysis of various systems with different statistical characteristics of foreground/background jobs and idle times shows that the effectiveness of idle wait critically depends on the variability in the empirical distribution of idle times. In systems with low variability of idle times, limiting the idle wait to zero is beneficial for system performance, the opposite holds for idle times of high variability. We use the cumulative data histogram of idle times to dynamically determine the length of idle wait. Additionally, we show that estimating the number of background jobs to be served in any given idle interval is an effective way to meet foreground performance targets.

2. SCHEDULING BACKGROUND JOBS

To evaluate the effectivess of idel waiting, the following scheduling policies are considered.

Mean-based: This serves as a base-line comparison [1]. When an idle interval occurs, no background job is scheduled during a delay period which is statically defined as the mean service time of background jobs. After the delay period elapses, the system starts serving background jobs until a foreground job arrives.

CDF-based: Similar to mean-based, this policy starts serving background jobs after an idle wait until a foreground job arrives. Different from the mean-based policy, the CDF-based policy continuously monitors the empirical cumulative histogram of idle intervals and the mean of background service times to dynamically calculate the idle wait. If idle times have low C.V., then the system does not idle wait, i.e., it schedules background jobs immediately. If idle times have high C.V., then the idle wait is calculated based on the empirical CDF.

CDF/w-estimates: This policy estimates the idle wait the same way as the CDF-based policy but is more conservative by limiting the number of background jobs to be served in an idle interval according to the following equation:

$$T \cdot \frac{90^{th} \text{ percentile of idle intervals - idle wait}}{\text{Average background service time}}.$$

T is a parameter that weighs the estimated number of background jobs assuming that the interval is large (i.e., equal to the 90^{th} percentile). This parameter controls the performance degradation of foreground jobs,

---

[*] This work was partially supported by the National Science Foundation under grant ITR-0428330 and by Seagate Research.

SIGMETRICS'07, June 12–16, 2007, San Diego, California, USA.
ACM 978-1-59593-639-4/07/0006.

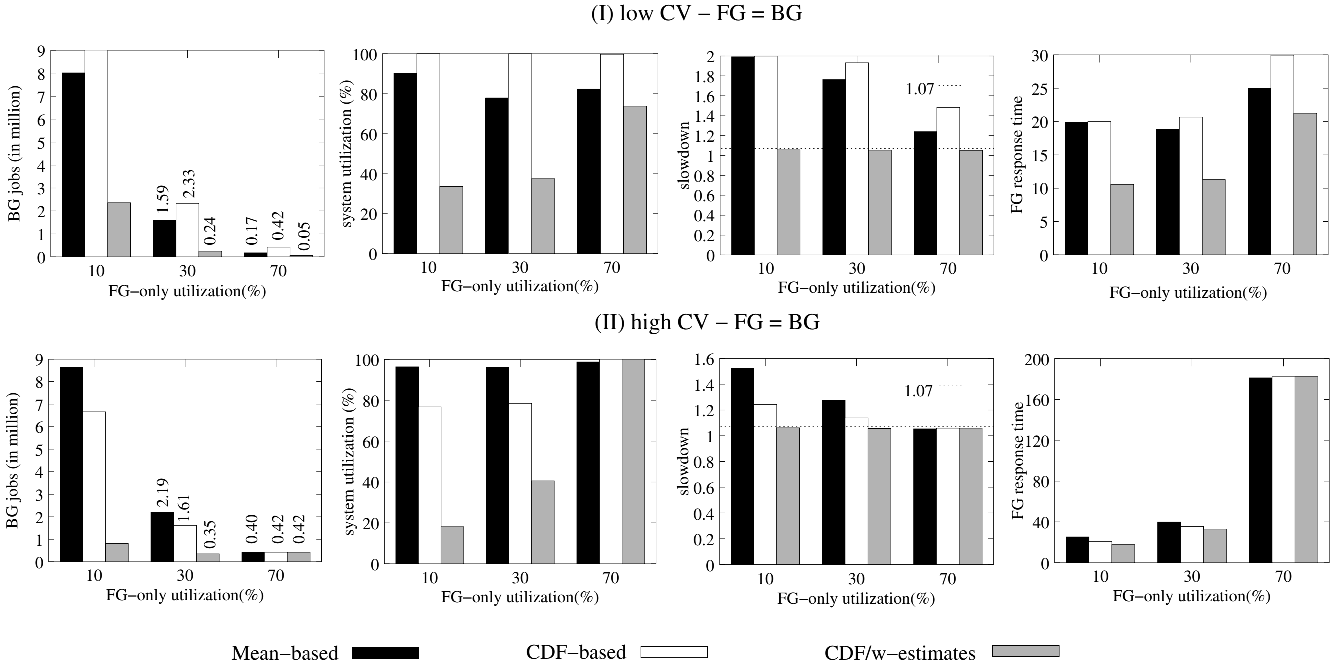

Figure 1: Overall system performance measured by number of completed background jobs in millions, overall system utilization, slowdown of the foreground jobs attributed to background activity (the horizontal line corresponds to 7% slowdown), and the absolute foreground response time. The idle intervals are of low variability (first row) and of high variability (second row). Three foreground system utilizations are evaluated, i.e., 10%, 30%, and 70%. Foreground utilization is controlled by changing the foreground arrival rate and fixing its service time. Results are reported with 98% confidence intervals.

i.e., the impact on foreground performance increases as T increases. The maximum value given to T is 1. This parameter is self-adjusted to reflect variability in the distribution of idle intervals. T is close to 1 under idle intervals of high variability and less than 1 for low variability intervals.

All of the above policies are non-preemptive. The three policies are evaluated via simulation of a single server queue. We assume that there is no limit on the waiting queue capacity and the service process is FCFS. We also assume that there are *always* background jobs waiting for service. The acceptable slowdown of foreground jobs due to background jobs is set to 7%. Service times of background jobs are exponentially distributed and service times of foreground jobs are drawn from a Lognormal distribution. Both foreground and background jobs have the same mean service time.

Figure **??** illustrates the performance of the three policies under idle intervals of low variability (first row of graphs) and of high variability (second row of graphs). The graph shows that the CDF/w-estimates policy consistently meets the performance target of foreground jobs while serving a large number of background jobs. Under low foreground-only utilizations there is more room to exploit idle times and serve large quantities of background jobs with small foreground performance degradation. Similar results are observed for systems where the average background time is larger (as much as seven times) than the average foreground time [2].

3. SUMMARY

We show that monitoring the stochastic characteristics of idle times is as important as monitoring the characteristics of foreground and background jobs. In particular, if idle intervals have low variability, then idle waiting is not effective. However, if idle times are highly variable, then idle waiting remains effective for scheduling background jobs without delaying foreground ones. For idle intervals with high variability, we propose to compute the length of idle wait dynamically using the cumulative histogram of the observed idle times in the system.

Apart from managing effectively idle intervals by distinguishing between low and high variability, we have also identified correlation as a source of additional information to improve idle time utilization. The analysis shows that if correlation exists in the observed idle interval lengths, then it can be used to predict the near future. For more details, we direct the reader to [2], where we also show the effectiveness of the methodology using actual measurements from disk drive traces.

4. REFERENCES

[1] L. Eggert and J. D. Touch. Idletime scheduling with preemption intervals. In *Proceedings of the International Symposium on Operating Systems Principles (SOSP)*, Brighton, UK, October 2005.

[2] Ningfang Mi, Alma Riska, Qi Zhang, Evgenia Smirni, and Erik Riedel. Efficient management of idleness in systems (extended version). Technical Report WM-CS-2007-03, Department of Computer Science, College of William and Mary, March 2007.