Supplementary Material for Improving Multicore Server Performance and Reducing Energy Consumption by Workload Dependent Dynamic Power Management

Keqin Li, *Fellow, IEEE*

TABLE 1
Summary of Notations

Notation	Definition
S	a multicore server processor
m	number of cores
λ	task arrival rate
r	task execution requirement
s	core execution speed
x	task execution time
μ	the average service rate
ρ	core utilization
p_k	the probability that there are k tasks in S
k	number of tasks waiting or being processed
P_q	the probability of queueing
$\bar{N}$	the average number of tasks in S
T	the average task response time
P	dynamic power consumption
w	an activity factor
C	the loading capacitance
V	the supply voltage
f	the clock frequency
α	an exponent such that $P \propto s^{\alpha}$
P^*	static power dissipation
μ_k	workload dependent task service rate
s_k	workload dependent core execution speed
ρ_k	workload dependent core utilization
(c, d)	an arithmetic-speed scheme $s_k = c + kd$
(q, a)	a geometric-speed scheme $s_k = qa^k$
(b, s_1, s_2)	a two-speed scheme
$\tilde{P}$	power constraint
$\tilde{T}$	performance constraint

1 Notations

For reader's convenience, we provide Table 1, which gives a summary of notations and their definitions in the order introduced in the paper.

2 Proofs of Theorems and Corollaries

Theorem 1: A speed scheme $(s_1, s_2, s_3, ...)$ is valid if and only if the summation

$$\sum_{k=m}^{\infty} \rho_1\rho_2\cdots\rho_k$$

converges.

Proof. It is clear that

$$p_k = \begin{cases} p_0 \dfrac{m^k}{k!} \cdot \rho_1\rho_2\cdots\rho_k, & 1 \le k \le m-1; \\ p_0 \dfrac{m^m}{m!} \cdot \rho_1\rho_2\cdots\rho_k, & k \ge m; \end{cases}$$

where

$$p_0 = \left(1 + \sum_{k=1}^{m-1} \frac{m^k}{k!} \cdot \rho_1\rho_2\cdots\rho_k + \frac{m^m}{m!} \sum_{k=m}^{\infty} \rho_1\rho_2\cdots\rho_k \right)^{-1}.$$

Therefore, for a fixed m, $p_0 > 0$ if and only if

$$\sum_{k=m}^{\infty} \rho_1\rho_2\cdots\rho_k$$

converges. ∎

Corollary 1: If there is some $n \ge 1$, such that $\rho_k < 1$ for all $k \ge n$, then a speed scheme $(s_1, s_2, s_3, ...)$ is valid.

Proof. Without loss of generality, we assume that $n \ge m$. It is clear that

$$\sum_{k=m}^{\infty} \rho_1\rho_2\cdots\rho_k$$

converges if and only if

$$\sum_{k=n}^{\infty} \rho_1\rho_2\cdots\rho_k$$

- *K. Li is with the Department of Computer Science, State University of New York, New Paltz, New York 12561, USA. E-mail: lik@newpaltz.edu* converges. Let $\rho = \max\{\rho_n, \rho_{n+1}, ...\}$. Then,

$$\sum_{k=n}^{\infty} \rho_1\rho_2\cdots\rho_k < \sum_{k=n}^{\infty} \rho^k = \frac{\rho^n}{1-\rho},$$

and

$$\sum_{k=n}^{\infty} \rho_1\rho_2\cdots\rho_k$$

converges. ▮

Corollary 2: Any arithmetic-speed scheme with $c > \lambda\bar{r}/m$ or $d > 0$ is valid.

Proof. It is clear that

$$\rho_k = \frac{\lambda\bar{r}}{m(c+kd)},$$

and $\rho_k < 1$ for all $k \geq n$, where

$$n = \left\lceil \frac{1}{d}\left(\frac{\lambda\bar{r}}{m} - c\right)\right\rceil.$$

The result follows Corollary 1. ▮

Corollary 3: Any geometric-speed scheme with $q > \lambda\bar{r}/m$ or $a > 1$ is valid.

Proof. It is clear that

$$\rho_k = \frac{\lambda\bar{r}}{mqa^k},$$

and $\rho_k < 1$ for all $k \geq n$, where

$$n = \left\lceil \log_a \frac{\lambda\bar{r}}{mq}\right\rceil.$$

The result follows Corollary 1. ▮

Corollary 4: Any two-speed scheme with $\rho_2 < 1$ is valid.

Proof. The result follows Corollary 1 immediately by taking $n = b + 1$. ▮

3 A Numerical Method and Examples of Power Consumption Reduction

In Table 2, we show numerical data for an optimal two-speed scheme, with the same parameter setting as Table 9 of the main paper. We observe that there is an optimal choice of (s_1, s_2), such that a two-speed scheme with a given b is optimized, in the sense that the power consumption P is minimized, while the average task response time is no more than (actually fixed at) T.

TABLE 2
Numerical Data for Optimal Two-Speed Schemes

	Idle-Speed Model		Constant-Speed Model	
s_1	s_2	P	s_2	P
$T = 0.80$				
1.20	4.7123790	90.5475024	4.7123790	91.4555628
1.25	3.5422746	63.1557243	3.5422746	64.2889058
1.30	2.9545801	52.6442432	2.9545801	54.0465778
1.35	2.6021857	47.5959053	2.6021857	49.3174316
1.40	2.3680850	44.9321569	2.3680850	47.0293046
1.45	2.2018070	43.5171008	2.2018070	46.0530438
1.50	2.0780066	42.8445489	2.0780066	45.8895223
1.55	1.9825587	42.6583950	1.9825587	46.2899715
1.60	1.9069675	42.8166746	1.9069675	47.1199978
1.65	1.8458156	43.2350867	1.8458156	48.3030594
$T = 1.00$				
1.15	2.0056416	41.8051046	2.0056416	42.2728288
1.20	1.9509196	40.8553629	1.9509196	41.4690733
1.25	1.9022092	40.1691893	1.9022092	40.9637048
1.30	1.8587553	39.7134783	1.8587553	40.7293344
1.35	1.8199114	39.4620199	1.8199114	40.7459277
1.40	1.7851217	39.3939612	1.7851217	40.9992359
1.45	1.7539062	39.4926306	1.7539062	41.4795854
1.50	1.7258491	39.7446313	1.7258491	42.1809280
1.55	1.7005890	40.1391357	1.7005890	43.1000900
1.60	1.6778107	40.6673325	1.6778107	44.2361683
$T = 1.20$				
1.10	1.7580956	38.7306356	1.7580956	38.9891602
1.15	1.7394136	38.3625219	1.7394136	38.7143489
1.20	1.7213085	38.0911069	1.7213085	38.5623417
1.25	1.7038463	37.9193108	1.7038463	38.5412141
1.30	1.6870797	37.8494471	1.6870797	38.6589762
1.35	1.6710477	37.8832801	1.6710477	38.9236095
1.40	1.6557767	38.0220867	1.6557767	39.3430968
1.45	1.6412811	38.2667167	1.6412811	39.9254430
1.50	1.6275646	38.6176489	1.6275646	40.6786837
1.55	1.6146217	39.0750405	1.6146217	41.6108811

TABLE 3
Numerical Data for Two-Speed Scheme Optimization

	Idle-Speed Model			Constant-Speed Model		
$\tilde{T}$	s_1	s_2	P	s_1	s_2	P
0.80	1.5494532	1.9834831	42.6583747	1.4872276	2.1064988	45.8703958
1.00	1.3946833	1.7886433	39.3930141	1.3214688	1.8415463	40.7066639
1.20	1.3087933	1.6842061	37.8478401	1.2329176	1.7097364	38.5332143

To minimize $P(s_1, s_2)$ subject to the constraint $T(s_1, s_2) = \tilde{T}$, we need to find s_1, s_2, and ϕ such that

$$\frac{\partial P}{\partial s_i} = \phi\frac{\partial T}{\partial s_i},$$

for $i = 1, 2$, that is,

$$\phi = \frac{s_1}{M_1} = \frac{s_2}{M_2},$$

where M_1 and M_2 are given in Section 5.1 of the main paper. We use a numerical method similar to that in Section 5.3 of the main paper, with the following differences. First, we can find s_2 such that $T(s_1, s_2) = \tilde{T}$ by using the classic bisection method and the fact that $T(s_1, s_2)$ is a decreasing function of s_2. Second, we notice that $\Delta(s_1) = s_1/M_1 - s_2/M_2$ is a decreasing function of s_1 along the path $T(s_1, s_2) = \tilde{T}$.

In Table 3, we display our optimal two-speed schemes with $m = 7$, $P^* = 2$, $\lambda = 10$, $\bar{r} = 1$, and $b = 10$, for both idle-speed and constant-speed models, where $\tilde{T} = 0.8, 1.0, 1.2$.

4 FURTHER RESEARCH DIRECTIONS

We would like to point out that our optimal two-speed scheme problem can be extended to include b as a variable. However, since b is a discrete variable, there hardly exists an analytical solution.

Furthermore, two-speed schemes can be generalized to d-speed schemes.

Definition. In a *d-speed scheme* with parameters $(b_1, b_2, ..., b_{d-1}, s_1, s_2, ..., s_d)$, where $b_1 < b_2 < \cdots < b_{d-1}$, the speed of the m cores is s_1 when there are $k \le b_1$ tasks, and s_2 when there are $b_1 + 1 \le k \le b_2$ tasks, ..., and s_{d-1} when there are $b_{d-2} + 1 \le k \le b_{d-1}$ tasks, and s_d when there are $k \ge b_{d-1} + 1$ tasks.

Our optimization problem for two-speed schemes can be generalized to d-speed schemes, i.e., to minimize T by choosing d, $b_1, b_2, ..., b_{d-1}$, and $s_1, s_2, ..., s_d$, subject to the constraint that the total power consumption does not exceed P. Although closed-form expressions of T and P can be figured out, the numerical procedures will be more involved.

Of course, our ultimate open problem is to find an optimal speed scheme $(s_1, s_2, s_3, ...)$ such that the performance is optimized with a power constraint or the power is minimized with a performance constraint.

Finally, it is of practical importance to incorporate the technique of workload dependent dynamic power management into real servers in cloud computing and data centers to improve server performance and to reduce energy consumption.