The Impact of Dynamic Power Management in Computational Clusters with Multi-Core Processors

X T Tran\textsuperscript{1}, Tien Van Do\textsuperscript{2,3,*} and R Chakka\textsuperscript{4}

\textsuperscript{1}Department of Networked Systems and Services, Budapest University of Technology and Economics, H-1117, Magyar tudósok körútja 2, Budapest, Hungary

\textsuperscript{2}Division of Knowledge and System Engineering for ICT (KSE-ICT), Faculty of Information Technology, Ton Duc Thang University, HCM City, Vietnam

\textsuperscript{3}Analysis, Design and Development of ICT systems (AddICT) Laboratory, Budapest University of Technology and Economics, H-1117, Magyar tudósok körútja 2, Budapest, Hungary

\textsuperscript{4}Rajeev Gandhi Memorial College of Engineering and Technology (RGM CET), Nandyal-518501, India

Received 3 April 2015; revised 27 December 2015; accepted 24 March 2016

In this paper, we study a question related to the execution of jobs in computing clusters built from servers with multi-core processors. Scenarios where a single job is executed or multiple jobs are simultaneously processed by multi-core processors are investigated. Workloads based on captured traces are used in our study. Numerical results demonstrate that the computing resources are efficiently utilized in a multiple job executing scenario and the setup time has a slight impact on the average response time of a computing cluster. Furthermore, a scenario where multiple jobs are simultaneously executed by cores and Dynamic Power Management (DPM) is applied for each processor core yields the most efficient energy consumption. As a consequence, schedulers should take account the feature of multicore processors to save the energy consumption of computing clusters.

Keywords: multi-core processor, heterogeneous cluster, HP policy, EE policy, single-job scenario, full-active, dynamic-active, Dynamic Power Management

Introduction

Multi-core processors have been applied in commercial electronic systems for a long time\textsuperscript{1,2,3}. It is shown that low-frequency multi-core processors enhance the overall performance capacity of systems with low energy consumptions. The availability of power management techniques allows the energy efficiency in complex computational systems\textsuperscript{4,5,6,7,8}. An industrial report\textsuperscript{5} presented a vSMP (variable Symmetric Multi Processing) multi-core architecture that enables and disables cores according to workloads to save power consumptions. Grochowski et al.\textsuperscript{3} discussed an asymmetric architecture with dynamic voltage and frequency scaling (DVFS) to balance between latency and performance. Some processor cores can be combined as a higher performance core to satisfy the different demands\textsuperscript{9,10}. Zikos and Karatza\textsuperscript{11} compared three policies applicable for cluster-level scheduling: SQEE (Shortest Queue based policy with Energy Efficiency priority), SQHP (Shortest Queue based policy with High Performance priority), and PBP-SQ (Performance-Based Probabilistic - Shortest Queue). Their simulation results indicated that SQEE is the best from the aspect of energy consumption, SQHP outperforms the other two schemes at the price of higher energy consumption, and PBP-SQ proved to be the worst among the three schemes. In addition, Terzopoulos and Karatza\textsuperscript{12}, and Gkoutioudi and Karatza\textsuperscript{13} investigated the scheduling in real-time grid systems. Do et al.\textsuperscript{14} also studied the impact of buffering schemes on performance of computational clusters. However, the scheduling issue of jobs and the energy consumption problem in computational clusters with multi-core processors have not been investigated. In this paper, we study scenarios related to the execution of jobs in heterogeneous computational clusters with multicore processors. We show that the computing resources are efficiently utilized in a multiple job executing scenario. Numerical results reveal that a scenario where
multiple jobs are simultaneously executed by cores and Dynamic Power Management (DPM) is applied for each processor core yields the most efficient energy consumption. As a consequence, schedulers should take account the feature of multicore processors to save the energy consumption of computing clusters. Following sections include a system description, a scheduling algorithm, planned scenarios, numerical results and conclusions.

A System Description
We consider a computational cluster of heterogeneous commercial servers and includes one common queue\textsuperscript{21}. We assume that the cluster is composed of

- \( K \) classes of identical servers, therein class \( i (i = 1, ..., K) \) consists of \( M_i \) servers,
- a common queue to store waiting jobs when none server is available,
- a local scheduler that is aware of servers’ performance and power and responsible for job assignment.

Dynamic Power Management (DPM) is applied in the cluster. DPM switches an idle server off and turns on a server to the active state when it is needed. The power consumption of a server in sleep state and the setup power consumption when \( k \) cores execute jobs \( (k = 1, ..., K) \) can be formulated as follows:

\[
P_{\text{sleep}} = 0 \text{W} \quad \text{and} \quad T_{\text{setup}} = 20\text{s.}
\]

Let \( S \) denote set of server types, then \( |S| = K \). The considered parameters of a server of type \( s (s \in S) \) are:

- \( N_s \) (the number of homogeneous cores),
- \( C_s \) (the ssj_ops value, defined by the number of operations per second based on the SPECpower ssj2008 benchmark of the Standard Performance Evaluation Corporation—SPEC),
- \( P_{\text{ac},s} \) (the average active power),
- and \( P_{\text{id},s} \) (power consumption when a server is idle).

We assume that jobs

- have service demand unknown to the local scheduler,
- can be executed on any server,
- are attended to scheduler by the First Come First Served (FCFS) service policy, and
- are uninterrupted during being executed (non-preemptible property).

We consider some performance measures\textsuperscript{14} that are listed as the following:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K )</td>
<td>Number of server classes</td>
</tr>
<tr>
<td>( M_i )</td>
<td>Number of servers in class ( i )</td>
</tr>
<tr>
<td>( \mu_i )</td>
<td>Service rate of each server in class ( i )</td>
</tr>
<tr>
<td>( \mu_{\text{system}} )</td>
<td>Service rate of entire system</td>
</tr>
<tr>
<td>( U )</td>
<td>Average system load,</td>
</tr>
<tr>
<td>( WT )</td>
<td>Expected waiting time per job</td>
</tr>
<tr>
<td>( ST )</td>
<td>Expected service time per job</td>
</tr>
<tr>
<td>( RT )</td>
<td>Expected response time per job</td>
</tr>
<tr>
<td>( AIE )</td>
<td>Average idle energy consumption per job</td>
</tr>
<tr>
<td>( AOE )</td>
<td>Average operation energy consumption per job</td>
</tr>
</tbody>
</table>

In this work, we investigate three scenarios to distribute jobs into computing servers.

- Scenario 1 (Single-job)- We assume that one server of type \( s (s \in S) \) handles only a single job at the full workload capacity \( C_s \) and consumes the maximum active power \( P_{\text{ac},s} \). DPM is deployed at the server level and performs switching off the server after it waits \( t_{\text{wait},s} \) seconds on the idle state.
- Scenario 2 (Full-active) - Cores are computing entities with the same processing capacity in servers. That means, a server of type \( s \) can handle \( N_s \) jobs in parallel, and each core has performance capacity \( C_s / N_s \). The server consumes the maximum active power \( P_{\text{ac},s} \) when it processes jobs. DPM is applied at the server level.
- Scenario 3 (Dynamic-active) - Each core of server of type \( s \) handles one job with the performance capacity of \( C_s / N_s \). The performance capacity and the power consumption of servers are controlled at the core level. DPM switches off unused cores and turns on cores when they are needed. Let \( C^k_s \) and \( P^k_{\text{ac},s} \) be the performance capacity and power consumption when \( k \) cores execute jobs concurrently in a server of type \( s \). Following the linear model of energy consumption\textsuperscript{15}, the dynamic active power \( P^{k}_{\text{ac},s} \) and performance \( C^k_s \) for a server of type \( s \) can be formulated as follows:

\[
P^{k}_{\text{ac},s} = P_{\text{id},s} + k/N_s (P_{\text{ac},s} - P_{\text{id},s}), \quad s=1...K, \quad ... (2)
\]

\[
C^k_s = k C_s / N_s, \quad s=1...K. \quad ... (3)
\]

In multi-job scenarios (scenario 2 and 3), the symmetric multiprocessoring (SMP) technology is supposed to be deployed. I.e., when a new job comes, job will be served immediately if there exists unused
core in the system. It is worth emphasizing that one server is able to handle jobs (a server is free) if it contains unused cores. We assume that a setup time to bring a server from the sleep to the active state is significant, but time to switch on a core from the sleep state is negligible in scenario 3. A job scheduling procedure is illustrated in Algorithm 1.

Algorithm 1 Schedule

for \( i = 1 \rightarrow K \)
for \( j = 1 \rightarrow M \)
    if server \((i,j)\) is FREE then
        if busy cores of \((i,j)\) are MAXIMUM then
            free_server \(\leftarrow (i,j)\)
        end if
    end if
end for
ALLOCATE:
if found free_server then
    ROUTE job to free_server
else
    ROUTE job to Common Queue
end if

Numerical Results

The considered cluster has three server classes \((K = 3)\). Each class consists of eight identical servers \((M_i = 8, i = 1, \ldots, K)\). Server types and their parameters are presented in Table 1. It is noted that the computing capacity of server of type \(i\), \(C_i\), and its power consumption, \(P_{ac,i}\), are measured at 100% target load. The optimal time for servers of type \(i\) to stay on the idle state before sleeping is calculated by eq. (1) and shown in Table 1 as well. The workloads for experiments are real traces that were collected in a production environment. The quantile-quantile (Q-Q) plots in the online e-companion show that both inter-arrival times and execution times do not follow an exponential distribution. Let \(\mu_1\), \(\mu_2\), and \(\mu_3\) denote the service rates, if jobs are scheduled to a server of type Intel Xeon E3-1240V2, Intel Xeon E5-2640, and Intel Xeon E5-2665, respectively. Assume that the service rate of server Intel Xeon E5-2665 is equivalent to service of input workload (i.e. \(\mu_3 = 1.0\)). We set parameters so that \(\mu_1 = 0.35\mu_3\) and \(\mu_2 = 0.74\mu_3\) if job is executed on servers Intel Xeon E3-1240V2 and Intel Xeon E5-2640, respectively. Note that in SMP scenarios, Full-active and Dynamic-active, without losing generality, every core of machine type \(i\) has the ssj_ops value of \(C_i/N_i\). This means each core handles jobs with rate \(\mu_i/N_i\) (see Table 3). Results are obtained by simulation runs with the confidence level of 95%. The accuracy (i.e. the ratio of the half width of the confidence interval and the mean of collected observations) of the energy metrics is less than 0.01. In this paper, we report results where the job arrivals are from a trace (called INCC) captured in a production environment. More results with other workloads can be found in the online e-companion. We plot the mean response time and the mean waiting time per job in Figures 1 and 2. It is observed that in the single-job scenario jobs wait longest. Figure 2 shows that system performs even worse with HP policy than with EE policy in multi-job scenarios. The reason is that the local scheduler is only aware about the performance at the server level. The scheduler considers servers of type Intel Xeon E5-2665 as the highest performance, but the fastest computing elements are the cores of Intel Xeon E3-1240V2 servers (see Table 3). In addition, it is observed that the mean response time per job is dramatically higher in multi-job scenarios than in a single-job scenario. We depict the energy efficiency in terms of the average operating energy per job (AOE) in Figure 3. The full-active multi-job scenario comes with a higher energy consumption than the single-job scenario.

<table>
<thead>
<tr>
<th>Table 1—Server parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server types</td>
</tr>
<tr>
<td>Fujitsu TX100 S3p (Intel Xeon E3-1240V2)(^{18})</td>
</tr>
<tr>
<td>Acer AR380 F2 (Intel Xeon E5-2640)(^{19})</td>
</tr>
<tr>
<td>Acer AR380 F2 (Intel Xeon E5-2665)(^{20})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2—Statistics of data traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data trace</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>INCC</td>
</tr>
<tr>
<td>NYS</td>
</tr>
<tr>
<td>UPR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3—Capacity of core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server types</td>
</tr>
<tr>
<td>Intel Xeon E3-1240V2</td>
</tr>
<tr>
<td>Intel Xeon E5-2640</td>
</tr>
<tr>
<td>Intel Xeon E5-2665</td>
</tr>
</tbody>
</table>
scenario, especially with HP policy. DPM technique at the core level yields significant energy savings as well. Therefore, the scheduler should take into account the cores of servers to save energy consumptions.

Conclusions

In this paper, we have investigated heterogeneous computational clusters with multi-core processors. The numerical results show that in the multi-processing job scenario, the computing resources are utilized efficiently with the slight increase of consumed energy in the idle state. Our study reveals that Dynamic Power Management (DPM) in the core level of processors can be an important candidate to save the energy consumption of computing clusters. In addition, a scheduler should be aware of the available cores of machines to fully exploit the performance of a computing cluster.

References


