Power-aware Provisioning of Cloud Resources for Real-time Services

Kyong Hoon Kim
Dept. of Informatics
Gyeongsang National University900 Gajwadong, Jinju, Korea

Anton Beloglazov, Rajkumar Buyya
CLOUDS Lab
Dept. Of Computer Science and Software Eng.
The University of Melbourne, Australia
Outline

• Introduction
• Framework
• Power-aware RT Cloud Service
• Simulation Results
• Conclusion
Introduction

• Cloud service datacenters consume 10 to 100 times more energy per square foot than typical office buildings. They can even consume as much electricity as a city!

• The main contribution of this paper is to provide real-time Cloud service framework for requesting a virtual platform, and to investigate various power-aware VM provisioning schemes based on DVFS (Dynamic Voltage Frequency Scaling) schemes.
Framework (1/5)

• Real-time Service Model

  – A real-time service is defined by:
    \[ \{ T_i(r_i, c_i, d_i, p_i, f_i) \mid i = 1, \ldots, n \} \] (n: Subtask number)
    
    • \( r_i \): release time
    • \( c_i \): worst-case execution time
    • \( d_i \): relative deadline
    • \( p_i \): period
    • \( f_i \): finish time

(Non-periodic application)
\[ T = r_i + d_i \]

(Periodic application)
\[ T = r_i + kp_i + d_i \]
**Framework (2/5)**

- **Real-time Virtual Machine Model**
  - In this paper, we define RT-VM as the requirement of a VM for providing a real-time service.
  
  \[
  \text{RT-VM } V_i = (u_i, m_i, d_i)
  \]
  
  - \(u_i\): utilization of real-time applications
  - \(m_i\): MIPS rate of the based VM
  - \(d_i\): lifetime or deadline
  
  - Thus, we assume that a RT-VM \(V_i\) is defined from multiple real-time applications, \(\{T_k | k = 1, \ldots, n\}\) set.
Framework (3/5)

• Real-time Cloud Service Framework

1. Requesting a virtual platform
2. Generating the RT-VM from real-time applications
3. Requesting a real-time virtual machine
4. Mapping the physical processors
5. Executing the real-time applications

Figure 1: Framework
Framework (4/5)

• Energy Model
  – The main power consumption in CMOS circuits is composed of *dynamic* and *static* power. We only consider the *dynamic* power because it is more dominating factor.
  – The dynamic energy consumption by an application is proportional to $V_{dd}^2$ (Supply voltage) and $f$ (Frequency)
Since the frequency is usually in proportion to supply voltage, $P = C \cdot f^3$

Consider an application of $t$ execution time at the frequency $f_{max}$ of the processor that runs at $f$ frequency level:

If $S = 1/2$, $t = 2x$ (The lower the freq., the longer the time)

\[ E = \int_0^{t/f_{max}} t \cdot f_{max} \]

\[ P = C \cdot t \cdot f_{max} \cdot f^2 = \alpha \cdot t \cdot S^2 \]

- $\alpha$: Coefficient
- $t$: Execution time
- $S$: Associated processor speed related to the frequency $f$
  ($S = f/f_{max}$)
Power-aware RT Cloud Service (1/9)

• Problem Description
  
  – A physical machine with one PE of 2400 MIPS
  – 3 RT-VMs to run

  • $V_1 \{0.2, 1000, 10\}$ – need 1000MIPS 20% for 10secs (2000)
  • $V_2 \{0.8, 500, 15\}$ – need 500MIPS 80% for 15secs (6000)
  • $V_3 \{0.5, 1200, 20\}$ – need 1200MIPS 50% for 20secs (12000)
Power-aware RT Cloud Service (2/9)

– Maximum Speed

• The proportional share of $V_i$ is defined by: \[
\frac{m_i \times u_i}{\sum (m_j \times u_j)}
\]

  - $V_1 = 0.2 \times \frac{1000}{1200} = \frac{1}{6}$ \quad 2400 \times \frac{1}{6} = 400
  - $V_2 = 0.8 \times \frac{500}{1200} = \frac{1}{3}$ \quad 2400 \times \frac{1}{3} = 800
  - $V_3 = 0.5 \times \frac{1200}{1200} = \frac{1}{2}$ \quad 2400 \times \frac{1}{2} = 1200

• Total Energy = 1 * 8.34 * 1^2 = 8.34 (Assume $\alpha = 1$)

\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
$t$ & $t = 0$ & $t = 5$ & $t = 7.09$ & $t = 8.34$ \\
\hline
\hline
$w_i$ & $ST_i$ & $w_i$ & $ST_i$ & $w_i$ & $ST_i$ \\
\hline
$V_1$ & 2000 & [0, 5] & 0 & - & - \\
$V_2$ & 6000 & [5, 7.09] & 2000 & 0 & 2990 \\
$V_3$ & 12000 & [7.09, 8.34] & 2990 & 0 & - \\
\hline
\end{tabular}
\caption{Remaining service times of Figure 2(a)}
\end{table}

\[(ST_i; t_1, t_2) : \text{The service time of } V_i \text{ from } t_1 \text{ to } t_2\]
Power-aware RT Cloud Service (3/9)

- DVS (Dynamic Voltage Scaling)
  - The processor dynamically adjust its speed to
    \[ \sum (m_j \times u_j)/2400 = S \]
    - \( V_1 = 0.2 \times 1000 = 200 \)
    - \( V_2 = 0.8 \times 500 = 400 \)
    - \( V_3 = 0.5 \times 1200 = 600 \)
  - Total Energy = \( 1 \times 10 \times (1200/2400)^2 + 1 \times 5 \times (1000/2400)^2 + 1 \times 5 \times (600/2400)^2 \)
    \[ = 2.5 + 0.882 + 0.3125 = 3.6945 \approx 3.69 \]
Power-aware RT Cloud Service (4/9)

– Acceptance Problem (Tradeoffs)
  • Operations in higher speed processor can accept more RT-VMs with more energy consumption.
  • On the contrary, scaling down to lower processor speed consumes less energy with lower acceptance.
  • If we have a new RT-VM $V_4$ (0.8, 2000, 10) that is required at time 10:
    – Maximum Speed scheme can accept it since the processor is idle.
    – DVS scheme cannot provision it due to lack of processor capacity.
Power-aware RT Cloud Service (5/9)

• Profit
  – Datacenters can increase their profit by:
    1. Provisioning more virtual machines to users
    2. Reducing energy consumption also increase profit by reducing the cost
  – Thus, this paper provides several schemes on power-aware provisioning of real-time VMs for the purpose of maximizing profits.
    1. Lowest-DVS
    2. $\delta$-Advanced-DVS
    3. Adaptive-DVS
  – Also, the provisioning policy in this paper is to select the processing element with the minimum price for the sake of users. (Next Slide)
Power-aware RT Cloud Service (6/9)

• DVS-enabled RT-VM Provisioning

  – Min-Price RT-VM Provisioning

  – For a given new RT-VM $V_i(u_i, m_i, d_i)$:

  • Check the schedulability of $V_i$ on the processing element $PE_k$ of $Q_k$ MIPS rate.

  \[
  u_i \times m_i + \sum_{j=1}^{n_k} \frac{w_j}{d_j - t} \leq Q_k \\
  w_j = u_j \times m_j \times (d_j - t)
  \]

  • Find the minimum-price processor.
  For the same price, less energy is preferable because it produces higher profits.

  • Create a VM on the selected processor for the user to execute services.

  – The resource provider provision the VM using DVS schemes to reduce the power consumption. The following subsections describe them.
Power-aware RT Cloud Service (6/9)

- DVS-enabled RT-VM Provisioning

```
Algorithm Min-Price RT-VM Provisioning (V_i)
1: VM ← null;
2: alloc ← -1;
3: e_min ← MAX_VALUE;
4: price_min ← MAX_VALUE;
5: for k from 1 to N do
6:   if ( u_i × m_i + ∑_{j=1}^{k} w_j × d_j ≤ Q_k ) then
7:     e_k ← energy_estimate (PE_k, V_i);
8:     price_k ← price for the RT-VM V_i in PE_k;
9:     if price_k < price_min or
10:        (price_k = price_min and e_k < e_min) then
11:        price_min ← price_k;
12:        e_min ← e_k;
13:        alloc ← k;
14:   endif
15: endfor
16: if alloc ≠ -1 then
17:   VM ← create_VM (PE_alloc, V_i);
18: endif
19: return VM;
```

Figure 3: Min-Price RT-VM Provisioning

\[ u_i, d_i \]:

The processing element \( PE_k \) of \( Q_k \) MIPS

\[ v_j = u_j × m_j × (d_j - t) \]

referable because it produces higher

for the user to execute services.

/\M using DVS schemes to reduce the power

/\ons describe them.
1. Lowest-DVS for VM Provisioning

– Adjusts the processor speed to the lowest level at which RT-VMs meet their deadlines.

– Each RT-VMs executes its service at the required MIPS rate.

• Consumes the lowest energy
• Lowest service acceptance rate
Power-aware RT Cloud Service (8/9)

2. $\delta$(Delta)-Advanced-DVS for VM Provisioning

– To overcome the low service acceptance rate of Lowest-DVS scheme.

– Over-scales more up to $\delta\%$ of the required MIPS rate for current RT-VMs. Thus, it operates the processor speed $\delta\%$ faster in order to increase the possibility of accepting coming RT-VM requests. (The value of $\delta\%$ is predefined in the systems according to the system load.)

– The processor scale $s$ is adjusted as the following equation at time $t$ for a given RT-VM set $T_k$:

$$s = \min \left\{ 1, \left( 1 + \frac{\delta}{100} \right) \times \frac{1}{Q_k} \sum_{V_i \in T_k} \frac{w_i}{d_i - t} \right\} \frac{f}{f_{\text{max}}} = s$$

All MIPS on this PE
3. Adaptive-DVS for VM Provisioning

- When the RT-VM arrival rate and its service time are known in advance, we can analyze an optimal scale.

- Using M/M/1 queuing model with arrival rate $\lambda$, service rate $\mu$ and processor speed scale $s$ to count average response time (RT).

\[
RT = \frac{1}{s\mu - \lambda} \leq d \text{ (Deadline)}
\]

\[
s^* = \frac{1}{\mu} \left( \lambda + \frac{1}{d} \right)
\]

- With the average arrival rate $\hat{\lambda}$, the average service rate $\hat{\mu}$ and the average deadline $\hat{d}$, we can count the scale $s$ at time $t$ for a given RT-VM set $T_k$.

\[
s = \max \left\{ \min \left\{ 1, \frac{1}{\hat{\mu}} \left( \hat{\lambda} + \frac{1}{\hat{d}} \right) \right\}, \frac{1}{Q_k} \sum_{i \in T_k} \frac{w_i}{d_i - t} \right\} \frac{f}{f_{\max}} = S
\]

\[
S \leq s^* \leq 1
\]
Simulation Results (1/5)

- **Environment**
  - Software: CloudSim
  - Hardware: 4 machines
    - Each machine has 4 DVS-enabled processors (Process Element)

- Generate 500 RT-VMs. The total service amount \( w_i \) of each RT-VM is randomly selected.

---

**Table 2: Characteristics of datacenter**

<table>
<thead>
<tr>
<th># of PEs</th>
<th>MIPS of PE</th>
<th>DVS level</th>
<th>( \alpha ) ((10^{-3}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine 0</td>
<td>4</td>
<td>1,800</td>
<td>[0, 1.0]</td>
</tr>
<tr>
<td>Machine 1</td>
<td>4</td>
<td>2,400</td>
<td>[0, 1.0]</td>
</tr>
<tr>
<td>Machine 2</td>
<td>4</td>
<td>3,000</td>
<td>[0, 1.0]</td>
</tr>
<tr>
<td>Machine 3</td>
<td>4</td>
<td>3,400</td>
<td>[0, 1.0]</td>
</tr>
</tbody>
</table>
Simulation Results (2/5)

• (a) Acceptance rate

The acceptance rate: Lowest-DVS < δ-Advanced-DVS < Adaptive-DVS ≈ Static. The lower the arrival rate, the higher the acceptance rate. On lower arrival rate there’s no difference (=100%).
Simulation Results (3/5)

• (b) Normalized power consumption

The power consumption is proportional to acceptance rate. The acceptance rate of Adaptive-DVS is close to Static but reduces much energy in case of low arrival rate.
Simulation Results (4/5)

• (c) Total profit

The total profit is proportional to acceptance rate. Static produces more profits since it accepts more RT-VMs, while other DVS schemes show more profits in lower arrival rates due to lower energy consumption.
Simulation Results (5/5)

- Extra: Impact of $\delta$ in $\delta$-Advanced-DVS

Higher $\delta$ shows better performance in higher arrival rate since it may accept more VMs. On the contrary, lower $\delta$ produces more profit in case of lower arrival rate.

Though $\delta$ is adjusted according to the system load, in the simulation the system utilization is generally high regardless of arrival rate. So $\delta$ has little impact on the profit.
Conclusion

• Simulation results show that datacenters can reduce power consumption and increase their profit using DVS schemes.

• Future work includes more analysis and improvement of the proposed adaptive schemes. (Ex: compare with other approaches such as bin packing or linear programming, and analyze the impact in the cooling systems.)