

Weilin Peng  
Sept 7<sup>th</sup>, 2009

# Server Power Management

## OUTLINE

- Introduction
- Dynamic Voltage and Frequency Scaling
- Request Batching
- Heterogeneity Aware Provisioning
- Measurement/ Characterizing Power Usage
- Future Outstanding Research Issues
- Example on DVFS
- Reading Material

## OUTLINE

- Introduction
- Dynamic Voltage and Frequency Scaling
- Request Batching
- Heterogeneity Aware Provisioning
- Measurement/ Characterizing Power Usage
- Future Outstanding Research Issues
- Example on DVFS
- Reading Material

## INTRODUCTION

- What is Data Center
  - Concentrated clusters of compute and data storage resources that are connected via high-speed networks and routers.

```

    graph TD
        HVAC1[HVAC] --- Server1[Server]
        Server1 --- Storage1[Storage]
        Storage1 --- Network((Network))
        Network --- Storage2[Storage]
        Storage2 --- Server2[Server]
        Server2 --- HVAC2[HVAC]
    
```

## ENERGY COST

- A 30,000 square feet data center
  - IT power consumption: 10MW=\$2-\$5Million
  - HVAC power consumption: \$4-\$8Million
- All Data Centers in US
  - \$5Billion in 2007
  - Raise to \$7.5Billion by 2010

## ENERGY COST cont.

**Data centric Data Center**

- HVAC
- Server
- Storage
- Network

**High Performance Computing Data Center**

- HVAC
- Server
- Storage
- Network

## DESIGN OBJECTIVE

- $P_{\text{cpu}} \propto V^2 F$ 
  - Static or Idle component
  - Dynamic component
- Minimizing dynamic power usage
- Minimizing static power usage
- Mapping workloads to most power efficient platforms
- Combination of the above three
- Measurement of power usage

## OUTLINE

- Introduction
- **Dynamic Voltage and Frequency Scaling**
- Request Batching
- Heterogeneity Aware Provisioning
- Measurement/ Characterizing Power Usage
- Future Outstanding Research Issues
- Example on DVFS
- Reading Material

## DVFS

- Change frequency and voltage of processors dynamically during runtime without significant delays.
- Processors are not the bottleneck
- Using DVFS do not violate the system performance requirement
- Need to predict the workload

## WORKLOAD PREDICT

- PAST algorithm
  - Introduced as early as 1994 by M. Weiser et al.
  - Predict the coming interval will be equally busy to the last completed interval
    - If predict "busy", increase the frequency
    - If predict "Idle", decrease the frequency
- The author identify the range of time interval from 0.01 seconds to 0.05 seconds

## DVFS IN Intel & AMD

- SpeedStep in Inter processors
  - V1.1 used in Pentium III processors with two modes.
  - From V3.1 (EITS), it has multiple levels from 40% to 100%
- Cool'n'Quiet and PowerNow! in AMD processors
  - Former used in desktop and latter in laptop
  - They also adjust speed of fans

## OUTLINE

- Introduction
- Dynamic Voltage and Frequency Scaling
- **Request Batching**
- Heterogeneity Aware Provisioning
- Measurement/ Characterizing Power Usage
- Future Outstanding Research Issues
- Example on DVFS
- Reading Material

## REQUEST BATCHING

- Throttling workload such that processors have longer idle periods
- A tradeoff between performance and responsiveness
- Example: Web Server, batches requests in memory and combines with DVFS

## OUTLINE

- Introduction
- Dynamic Voltage and Frequency Scaling
- Request Batching
- **Heterogeneity Aware Provisioning**
- Measurement/ Characterizing Power Usage
- Future Outstanding Research Issues
- Example on DVFS
- Reading Material

## HETEROGENEITY

- Heterogeneity
  - Large system are made up of collections of hundreds of smaller components
  - Nodes can be different in various dimensions
- Workload Provisioning (Heath et al.[05])
  - Minimize the ratio of overall power consumption and throughput
  - Nodes could forward requests to others based on cluster wide optimality

## WORKLOAD MAPPING

- Mapping workload to different platform
- How to quantify workload?
  - $CPI = CPI_{Core} + CPI_{Mem}$
  - $CPI(f_1) = CPI_{Core}(f_1) + MPI * M_L * BF(f_1)$
  - $CPI(f_2) = CPI_{Core}(f_1) + MPI * M_L * (f_2/f_1) * BF(f_1)$
- Nathuji et al. [07]

## OUTLINE

- Introduction
- Dynamic Voltage and Frequency Scaling
- Request Batching
- Heterogeneity Aware Provisioning
- **Measurement/ Characterizing Power Usage**
- Future Outstanding Research Issues
- Example on DVFS
- Reading Material

## POWER USAGE MEASUREMENT

- By collecting data in Google data center,
  - A gap between nameplate power and peak power consumption
  - Actual peak power consumption occurs only for short duration
- By profiling a subset of standard programs,
  - Strong correlation between IPC and OS routine power consumption

## OUTLINE

- Introduction
- Dynamic Voltage and Frequency Scaling
- Request Batching
- Heterogeneity Aware Provisioning
- Measurement/ Characterizing Power Usage
- **Future Outstanding Research Issues**
- Example on DVFS
- Reading Material

## CURRENT & FUTURE

- No good solutions so far, since
  - Lots of parameters in data center are missing
  - Deep understanding of tradeoffs of factors
  - Experiment environment
- Future work may focus on,
  - Multi-core processor ---performance & power
  - Graphics Processing Units (GPU)
  - Flash based SSD as primary memory

## OUTLINE

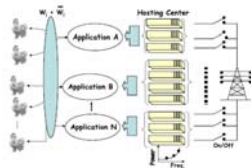
- Introduction
- Dynamic Voltage and Frequency Scaling
- Request Batching
- Heterogeneity Aware Provisioning
- Measurement/ Characterizing Power Usage
- **Future Outstanding Research Issues**
- **Example on DVFS**
- Reading Material

## EXAMPLE ON DVFS

- Paper Name: Managing Server Energy and Operational Costs in Hosting Centers
- Author: Yiyu Chen et al.
- Abstract of Abstract
  - Service-level agreement (SLA) for applications
  - Energy consumption is critical
  - Switching on/off server or modulating speed may break SLAs and cause some more cost
  - Three techniques solve the above problem

## SYSTEM MODEL

- SLA = Average Respond Time
- Two methods
  - Temporarily switch off servers
  - DVFS
- For DVFS
  - Coordinated strategy where a controller is assigned for each application (Centralize Control)



## SOLUTION STEPS

1. Perform server provisioning to decide how many servers to allocate to each application.
2. Decide the operating frequency for the server allocated to each application.

### ENERGY COST

- $P = P_{fix} + P_f * f^3$ 
  - P: Dynamic power consumption
  - $P_{fix}$ : Other components in server except CPUs
  - f: Frequency
  - P is proportional to  $V^2 * f$
  - Frequency and voltage:  $V \propto f$
- Energy of M servers over t time
 
$$E = M * \int_t P * dt = M * (P_{fixed} + P_f * f^3) * t$$

### ENERGY COST cont.

- Electricity Cost of Operation:
 
$$\sum_{z=1}^Z \sum_{i=1}^N K_{\$} * m_i(z) * (P_{fixed} + P_f * f_i(z)^3) * t$$
- Cost of Server Turn-ons:
 
$$\sum_{z=1}^Z E_o * \left[ \sum_{i=1}^N m_i(z) - \sum_{i=1}^N m_i(z-1) \right]^+$$
- Total Cost:
 
$$\sum_{z=1}^Z \left( \sum_{i=1}^N K_{\$} * m_i(z) * (P_{fixed} + P_f * f_i(z)^3) * t + E_o * \left[ \sum_{i=1}^N m_i(z) - \sum_{i=1}^N m_i(z-1) \right]^+ \right)$$

### METHODOLOGY

- Three mechanisms
  - Queuing theory based approach: pro-active
  - Control theoretic approach: reactive
  - Hybrid Approach: Combine with above two

### METHODOLOGY cont.

- All  $m_i$  server of application i run at same frequency  $f_i$
- Server allocation/provisioning performed in large time granularities
- Turning servers to different frequencies performed in small time granularities

### QUEUING THEORY BASED APPROACH

- Considering N parallel queues denoting N applications
- Client requests for application arrive into corresponding queue
- Steps:
  - Predict request arrival pattern and service time requirements
  - Determine mean response time for each apps
  - optimization

### PREDICTION

- Predicting four parameters:
  - Mean arrival rate of requests ( $\lambda$ )
  - Squared coefficient of variation of request inter-arrival time ( $C_a^2$ )
  - Mean file size in bytes ( $\phi$ )
  - Squared coefficient of variation of file size ( $C_s^2$ )
- Method
  - First two, using S-ARMA
  - Next two, using Winter's smoothing method

## QUEUING ANALYSIS

- With predictions of four parameters
- Predicted average response time  $W_i$

$$W_i = \frac{\phi(i)}{\beta * f_i} + \frac{\alpha m_i \phi(i)}{\beta * f_i} \left( \frac{1}{1 - \rho_i} \right) \left( \frac{C_a^2(i) + C_s^2(i)}{2m_i} \right)$$

- Where

$$\rho_i = \frac{\lambda(i) \phi(i)}{m_i * \beta * f_i} \quad \alpha m_i = \rho_i^{\frac{m_i + 1}{2}}$$

- $\beta * f_i$  is the bandwidth of the server in bytes served per second for application

## OPTIMIZATION

- Objective:

$$\min_{f_i(u,s), m_i(u)} \sum_{u=1}^U \left( \sum_{s=1}^S \sum_{i=1}^M K_i m_i(u) (P_{fixed} + P_f * f_i(u,s)^3) t + B_0 \left[ \sum_{i=1}^M m_i(u) - \sum_{i=1}^M m_i(u-1) \right]^+ \right)$$

subject to

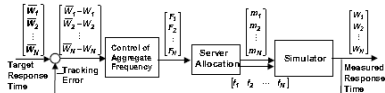
$$W_i \leq \bar{W}_i \text{ for all } i$$

$$\sum_{i=1}^M m_i(u) \leq M \text{ for all } u, \text{ and } m_i(u) \text{ is a non-negative integer}$$

$$f_i \in \mathcal{F}$$

- Finite pair of  $f$  and  $m$  so Brute force

## CONTROL THEORETIC APPROACH



- Two sub-problem
  - Computing aggregate frequency for each application with a single server
  - Allocating servers to meet aggregate frequency

## AGGREGATE FREQUENCY

- Re-write Electricity Cost of Operation to:

$$\sum_{i=1}^N \sum_{k=1}^{U+S} (R_F * F_i^3(k) + R_W * (W_i(k) - \bar{W}_i)^2)$$

- $R_F$  and  $R_W$  are parameters of tradeoffs between target response time and energy cost
- Constrain of aggregate frequency  $F_i$  is

$$\sum_{i=1}^N F_i(k) \leq M * f_{max}$$

## SERVER ALLOCATION

- Given the aggregate frequency  $F_i(u,s)$  for each application  $i$

$$m_i(u) = \lceil m(u) * \frac{\max_s F_i(u,s)}{F(u)} \rceil$$

- $m(u)$  denote total # of servers turn on
- $F(u)$  denote total capacity that  $m(u)$  provide
- Cost function 
$$\sum_{u=1}^U (K_i * m(u) * T * (P_{fixed} + P_f * (F(u)/m(u))^3)) + \sum_{u=1}^U B_0 * [m(u) - m(u-1)]^+$$

## SERVER ALLOCATION cont.

- With feedback, only  $F(u-1)$  is available
- An online greedy algorithm to get  $m(u)$ 
  - Ignore the server turn-on cost
  - Take first derivation of last equation

$$m^*(u) = \lfloor F(u) * \left( \frac{2P_f}{P_{fixed}} \right)^{1/3} \rfloor$$

- $m^*(u)$  denote the # of server should be turn on

- Finally, 
$$f_i(u,s) = \mathcal{D} \left( \frac{F_i(u,s)}{m_i(u)} \right) \quad \mathcal{F} = (f_1, f_2, \dots, f_\ell)$$

## HYBRID APPROACH

- Predict information of queuing approach to determine server allocation at large granularity
- Use feedback based control theoretic approach for frequency setting in small granularity

## OUTLINE

- Introduction
- Dynamic Voltage and Frequency Scaling
- Request Batching
- Heterogeneity Aware Provisioning
- Measurement/ Characterizing Power Usage
- Future Outstanding Research Issues
- Example on DVFS
- **Reading Material**

## READING MATERIAL

- N. Mandagere, D. Du, "*Data Center Power Management*"
- Y. Chen, A. Das, et al., "Managing server energy and operational costs in hosting centers"
- X. Fan, W.D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer"
- R. Nathuji, C. Isci, and E. Gorbatoov, "Exploiting platform heterogeneity for power efficient data centers"

## THANKYOU!

QUESTION?

COMMENT?