Energy efficient computing in datacenters

Prof. Tajana Šimunić Rosing
Dept. of Computer Science

System Energy Efficiency Lab
seelab.ucsd.edu
Energy efficiency at global scale

NSF GreenLight & IRNC: TransLight/StarLight

Wind from NREL, 2MW Solar @UCSD

2.8MW Fuel Cell Power Plant, UCSD
Key issues for distributed renewable-powered datacenters

- Green energy availability varies dramatically
  - Instantaneous use leads to significant energy efficiency losses -> prediction

- Datacenters energy needs are higher than green energy availability
  - Brown energy needs to be present to both supplement green and as “insurance” to meet performance constraints
  - Improvements in computation & networking infrastructure energy efficiency are necessary (power, thermal and cooling management)

- Datacenter computing requires consistent performance
  - Infrastructure that monitors and manages computation in datacenters has to be aware of performance metrics – guaranteeing QoS is key!
    - Service response times are around 100ms, Max 10% batch job throughput hit

- Fast network connections are critical
  - Telecom infrastructure up to 40% of the overall IT energy cost [SMART 2020]
  - Energy costs are skyrocketing; e.g. Telecom Italia is the 2nd largest electricity consumer in Italy [INTELEC 2007]
Datacenter energy efficiency

\[
\text{Efficiency} = \frac{\text{Computation}}{\text{Total Energy}} = \left( \frac{1}{\text{PUE}} \right) \times \left( \frac{1}{\text{SPUE}} \right) \times \left( \frac{\text{Computation}}{\text{Total Energy to Electronic Components}} \right)
\]

**Computer Air Handling Unit (CRAU)**
- Up to 31 Ton Nominal Capacity Per Unit
- Air Handling Can Be Exploded Off
- Configuration Can Be Customized

**Individual Compute Cabinets**
- Typically Configured (32" W x 32" D x 48" H)
- Standard 1U to 4U With Power Supply

**Emergency Diesel Generators**
- Total Generator Capacity = Total Electrical Load x (1.0-1.2)
- Multiple Generators Can Be Electrically Combined
- Fuel Oil Storage Tanks
- Can Be Located Underground Or Grade

**UPS System**
- Uninterruptible Power Supply Modules
- Can Be Extended To Meet Site Requirements

**Electrical Primary Switchgear**
- Includes Power Transformers
- Can Be Extended To Meet Site Requirements

**Heat Rejection Devices**
- Chiller, Air-Cooled Chiller, etc.
- Used To Pump Condensed Chilled Water Between Datacenter and CRAU Units

**Pump Room**
- Used To Pump Condensed Chilled Water Between Datacenter and CRAU Units

**Graph**
- Distribution of CPU Utilization
- Power as a Percentage of Peak

**Bar Chart**
- Power Consumption: CPU, DRAM, Disk, Other

One server
- DRAM: 16GB, 100ns, 20GB/s
- Disk: 2TB, 10ms, 200MB/s

Local rack (80 servers)
- DRAM: 1TB, 300us, 100MB/s
- Disk: 160TB, 11ms, 100MB/s

Cluster (30 racks)
- DRAM: 30TB, 500us, 10MB/s
- Disk: 4.80PB, 12ms, 10MB/s

Barroso & Hölsze, 2009
Energy efficiency of the infrastructure

Dynamic thermal management (DTM)
- Workload scheduling:
  - Machine learning for dynamic adaptation
- Proactive thermal management
  - Reduces thermal hot spots by average 80% with no performance overhead
- Cooling aware management
  - Savings of 70% in cooling subsystem

Dynamic power management (DPM)
- HW level: adaptive power gating gives 40% energy savings with no perf. impact
- SW level: 92% reduction in performance variability with DVFS
- Optimal DPM for a class of workloads
- Machine learning to adapt
  - Measured energy savings of 70%
- VM management

NSF Project GreenLight
- Green cyber-infrastructure in energy-efficient mobile facilities
- Closed-loop power and thermal management
**HW level: TAP - Token-based Adaptive Power gating**

**Goal:** Power gate cores on L2 and L3 cache misses with no performance hit by effectively utilizing a token based mechanism

- **PPGS:** Programmable Power Gating Switch
  - Variable wakeup latency and peak current
  - Leakage reduction does not depend on wakeup time
- **Token Controller** manages PPGS wakeup modes
  - Adapts to system utilization
    - More idle cores relax constraints
  - Manages peak current and voltage swing constraints

Jointly with: Andrew Kahng, UCSD & GSRC
TAP: Tokens In Flight

Jointly with: Andrew Kahng, UCSD & GSRC
We run SPEC benchmarks within M5 with McPat for 32nm technology node per ITRIS 2010 data with TAP implemented on the following cores:

- One In-Order (IO) core: Atom+@2GHz
- Three Out Of Order (O3) cores:
  - ARM9@2GHz with 2 issue
  - EV6@3GHz with 6 issue
  - EV8@3GHz with 8 issue

Results:

- IO vs. O3 cores spend up to 89% vs. 72% of time stalling for memory
- IO vs. O3 can power gate 74% of run time for lbm vs. 60% for mcf
- IO vs. O3 can save up to 42% energy for lbm vs. 27% for mcf

Jointly with: Andrew Kahng, UCSD & GSRC
OS level: Meeting performance targets with minimal cost

OS/HW interface

*Ondemand* (Linux)

OSPM

Platform

P-State requests

Utilization

Our approach

Application

Performance target set

OSPM

Platform

Scalability factor ($S_F$), Instruction execution rate

P-State requests

- Guarantees meeting performance target
- Runs at the lowest frequency while meeting required performance targets

- Does not consider performance constraints
  - Actual performance can vary
- Does not consider characteristics of application & cannot minimize frequency
• **Goal:** Design a controller to meet the performance target by regulating the slack.

• Target performance is a ratio of max performance; $\beta = 0.9$, provided by the application:

$$IPS_{ref} = \beta \times IPS_{max}$$

**Target frequency**

$$f_{ref} = \frac{S_F \times f_{max}}{1/\beta - (1 - S_F)}$$

• $S_F = \frac{T_{scalable}}{T_{active}}$
Performance variations with Linux DVFS policy

**OS Policy:** SpecWeb with 2000 connections

High performance variations

**Our Policy:** SpecWeb with 2000 connections

Performance variations are minimized
Meeting performance targets

- Implementation in Linux kernel 2.6.33.2 using SPEC benchmarks
- Experiments are conducted on a Westmere processor with two 6 core CPU sockets capable of 6 operating frequencies: from 1.6, to 2.267 GHz

**FFPA policy:**

\[
 f = \text{ceil} \left( \beta \times f_{\text{max}} \right)
\]

- Our technique is able to meet the performance target for different workload combinations running on multiple sockets
- The reduction in average standard deviation from target performance over other policies reaches 92% with average power reduction of 17%
OS level DPM:
DPM: Workloads - Idle State

Storage Trace

Pareto Distribution:

\[ E_{user} = 1 - a \cdot t^{-b} \]

Network Traces
WWW – left; Messaging – right
DPM: TISMDP model

Assumptions:
- General distribution governs the first request arrival
- Exponential distribution represents arrivals after the first arrival
- User, device and queue are stationary

Obtain globally optimal policy using linear programming

Measurements within 11% of ideal oracle policy
- Factor of 2.4 lower than always-on
- Factor of 1.7 lower than default time-out
Online Learning for Power Management

- Measured large energy savings – up to 70% per device
- Experts control sleep states and voltage/freq. settings

Selected expert manages power for the operative period

Selects the best performing expert for managing power

Converges to the best performing expert at $O\left(\sqrt{\ln N / T}\right)$

Evaluates performance of all the experts

Controller

Device

EXP 1, EXP 2, EXP 3, ..., EXP n

EXP y: Dormant Experts
EXP y: Operational Expert
Adaptive Cooling & Thermal Management

Evaluation framework

Inputs:
- Workload – collected at a data center
- Floorplan, temperature (for dynamic policies)

Resource manager
Static: Fixed allocation (ILP)
Dynamic: Dependent on the policy

Power Manager
DPM, DVS

Thermal Simulator
HotSpot [Skadron, ISCA'03]

Transient Temp. Response for Each Unit

Inputs:
- Power trace for each unit
- Floorplan, package and die properties (Niagara-1)
Load balancing vs. optimal policies

- Energy or performance-aware methods are not always sufficient to manage temperature.
Dynamic Policies: Thermal Hot Spots

- Workloads collected at an operational datacenter over a period of a week; concatenated 1hr of each day to show adaptation

Online learning gives 20% hot spot reduction in average in comparison to the best policy.
Reactive vs. Proactive Management

- **Reactive**

![Graph showing Reactive Management](image-url)
Reactive vs. Proactive Management

- **Reactive**
  - e.g., DVFS, fetch-gating, workload migration,
  - ...

- **Proactive**
  - *Forecast*
Reactive vs. Proactive Management

- **Reactive**
  - e.g., DVFS, fetch-gating, workload migration, ...

- **Proactive**
  - Reduce and balance temperature
    - Two techniques: on chip fast technique, and longer time horizon
Proactive vs. Reactive: Hot Spots

- Proactive Balancing (PTB) achieves similar hot spot reduction to P-DVS while improving performance by ~8%
- PTB reduces hot spots 80% over default load balancing (DLB)
Integrating energy, temperature and cooling and management

- Fan control is done jointly with job and memory scheduling
  - state of the art fan control operates independently from workload scheduling
- **Controller** decides the following on each tick:
  - CPU power distribution
  - DIMM power distribution
  - Desired temperature to control fan speed
- **Actuators**: CPU, memory & fan actuators implement controller’s decisions independently
- **Sensor input to controller**: temperature, power, fan speed
Combined Energy Thermal & Cooling, CETC Management Results

- Average cooling savings of 70% relative to state-of-the-art

Intel Xeon Dual Socket Quad core Server; with state-of-the-art PI fan controller

- CETC: Our policy
- DLB: Dynamic load balancing (baseline)
- NFMO: Only page migrations allowed
- NMM: No memory clustering
- NCM: No CPU scheduling optimizations

<table>
<thead>
<tr>
<th>Local ambient Temp.</th>
<th># DIMM</th>
<th>CETC %</th>
<th>NFMO %</th>
<th>NMM %</th>
<th>NCM %</th>
<th>DLB %</th>
</tr>
</thead>
<tbody>
<tr>
<td>45 °C</td>
<td>8</td>
<td>0.175</td>
<td>0.184</td>
<td>0.093</td>
<td>0.102</td>
<td>0</td>
</tr>
<tr>
<td>35 °C</td>
<td>8</td>
<td>0.115</td>
<td>0.120</td>
<td>0.069</td>
<td>0.100</td>
<td>0</td>
</tr>
<tr>
<td>45 °C</td>
<td>16</td>
<td>0.175</td>
<td>0.187</td>
<td>0.109</td>
<td>0.102</td>
<td>0</td>
</tr>
</tbody>
</table>

- CETC performance overhead < 0.2%
- CETC page migration rate < 5 pages/sec -> negligible overhead & high stability
Measuring the datacenter energy costs: UCSD’s NSF GreenLight infrastructure

- Focus on communities with at-scale computing needs:
  - Metagenomics
  - Ocean observing
  - Microscopy
  - Bioinformatics & health
  - Digital Media

- Measure, monitor & publish real-time data
  - Allow researchers anywhere to study computing energy costs
  - Enable scientists to explore tactics for maximizing work/watt

- Develop energy management strategies based on models developed

Jointly with Ingolf Krueger, UCSD
NSF GreenLight: Dashboard & History plots

- Multiple sensor data: temperature, fan speed, liquid flow rate & temp, power
- Use measurements to develop models needed for energy management
**GMVQ VM Power Cost Prediction**

- **Goal:** Estimate how much a VM consumes and predict what the cost would be if it migrates to another machine.
- **Approach:** Gaussian Mixture Vector Quantization (GMVQ) to fit a GMM to the training data.

**NSF FlashGordon:** Design of an energy-efficient supercomputer.
Energy management with virtualization

vGreen

- **Scheduling**
  - Co-locate batch VMs with orthogonal characteristics
- **Management policies**
  - Based on CPU/mem metrics maintained per guest

- Measured energy savings of 35% with speedup of 40%
vGreen++

Batch Jobs:  
• MIPS driven

Services:  
• Latency sensitive

- Maximize $q_{MIPS}/Watt$
  - $q \rightarrow$ QoS ratio < 1
  - MIPS $\rightarrow$ Batch job throughput
  - Watt $\rightarrow$ Power consumption

Service Request

Response

Resource Management Commands

Metric Updates

Resource Management Commands

Node Controller

AppProfiler

sClient

QoS Ratio Feedback

Tajana Simunic Rosing
Managing multi-tier applications

- What happens when we combine:
  - Latency sensitive jobs (e.g. RUBIS)
  - Throughput sensitive (e.g. batch jobs)

- Preliminary results:
  - More than 10x improvement in SLA with background jobs relative to the default scheduler

RUBiS: auction website

vGreen: Rubis & batch job

Graph showing CPU Util(%) and SLA Ratio over time.
Policies compared

- **State of the art:**
  - **Selective Consolidation (tChar):** vGreen
  - **Capping (tCap):** Cap the CPU allotment to bVM to mitigate interference effects (Padala et al, EuroSys’09, Nathuji et al, EuroSys’10)

- **Controller:**
  - Dynamically control the virtual CPU allocation of the service VM to maximize MIPS of batch job while meeting SLA
Our controller is within 7% of baseline; CPU capping gets on average 25% lower throughput
Baseline: running services and batch jobs on separate servers
Energy Efficiency Improvements

- Use a controller to manage virtual CPUs dynamically
  - Maximize CPUs of various batch jobs while meeting Rubis SLAs

- 70% more efficient than running service & batch VMs separately while within 7% of maximum batch job throughput
- 35% more efficient than the ideal version of state of the art
Green Energy Prediction and VM Scheduling

- Predict green energy availability for the next 30min window
  - Schedule additional MapReduce jobs accordingly; they take max 30mins

- Data from solar panels at UCSD
- State of the art: exponential weighted average
  - EWMA: 32.6 % error
  - Extended eEWMA: 23.4% error
- **Our algorithm:**
  - WCMA: less than 9.6% error

- Data gathered from a wind farm in Lake Benton, available by NREL
- State-of-the-art:
  - Integrated predictor: 48.2% error
- **Our algorithm: 21.2% error**
  - Combination of a weighted nearest-neighbor (NN) tables and wind power curve models
Benefits of Green Energy Prediction

Experimental setup:
- Schedule additional MapReduce jobs depending on green energy availability
- Mix services and batch jobs
  - Rubis with 100ms 90th%ile SLA
  - MapReduce with max 10% throughput reduction
- Run on Intel Xeon cluster (250W/server)

- **Prediction has 93% GE Efficiency**
  - ratio of green energy consumed for useful work vs. total green energy available

- **Prediction has 22% faster job completion time vs. default**

- **Prediction has 38% more jobs completed vs. instantaneous GE usage**
  - *GE Job %*: ratio of jobs completed with GE over all completed jobs.
How about networking?

- Increase the energy efficiency of backbone network
  - Shutting down the idle network elements [SIGCOMM10]
    - Ensure connectivity is not affected when they are shut down
  - Adjusting the number of active line cards [INFOCOM08]
    - Line cards consume a large portion of the router power
    - Adjust bandwidth accordingly
- Dynamic software solutions
  - Energy aware routing [INFOCOM09]
    - Select the energy efficient path and adjust bandwidth accordingly

Green-energy aware routing

- Shutting down nodes or links affects performance
  - May decrease availability and connection speed
  - Instead, we show the effects of energy proportional hardware on energy efficiency
- Green energy is highly variable -> use prediction
  - Improves the reliability
- Use of dynamic routing policies
  - Brown and/or green energy aware policies
  - Decreases the brown energy use significantly
Load Balancing Across Data Centers

Send available resource profile to controller

{C_1, M_1} → {C_n, M_n}

DC# 1 → DC# 2 → DC# n

Each data center \(i\) sends a set of tuples \{C_i, M_i\}

\(C_i = \#\)CPU cores available in center \(i\)

\(M_i = \)Amount memory available in center \(i\)

Check active MR jobs to determine how many tasks can be sent in real {n_1}

Each active MR job \(j\) sends a number to DC controller

\(n_j = \#\)subtasks that Job \(j\) can transfer

Send data transfer information to the main controller

{C_1, M_1} → {C_n, M_n}

DC# 1 → DC# 2 → DC# n

Each data center \(i\) gets a set of numbers \{N_{ij}\}

where:

\(N_{ij} = \#\)VMs that CAN be sent from data center \(i\) to \(j\)

Each data center \(i\) sends a set of numbers \{N_{ij}\}

Each data center \(i\) sends a set of numbers \{T_{ij}\}

where:

\(T_{ij} = \#\)VMs that WILL be sent from data center \(i\) to \(j\)
Based on typical network characteristics [IEEE10]

- Routers, hubs, storage, computation elements
- Focus is on routers
- Large portion of the power is consumed by routers

Network Power Model

- Optical links have a fixed cost
  - A function of distance between amplifiers

- Router energy cost:
  - Linear model based on bandwidth utilization [NET09]
  - Evaluate energy proportional routers as well

---

**OSCARS vs. Energy Aware Routing**

- On-demand Secure and Advanced Reservation System (OSCARS; by ESnet)
  - Online path computation
  - Constructs virtual circuits with required bandwidth
  - Improves performance substantially
- Works well with systems where the main goal is performance
  - Energy consumption is not a constraint
  - **Solution:** Energy-aware dynamic routing is needed to account for both performance and energy consumption
Routing Policies

- Constructing a path
  - Shortest Path Routing (SPR)
    - Based on Dijkstra's algorithm
  - Green Energy Aware Routing (GEAR)
    - Based on green energy availability on the nodes
    - Available green energy is estimated using prediction
    - Chooses the path with the least brown energy need

SPR selects Path1 in every case as it is shorter than Path2, between N1 and N2
- There is no other criteria (as in OSCARS)

GEAR does not always select the same path between two nodes. It may select Path2, even if it is longer, as its brown energy need may be less
Bandwidth adjustment

• All bandwidth policy (AB)
  – Allocates all the bandwidth available on a path
  – Results in fast transfer times, but decreases availability

• Necessary Bandwidth Policy (NB)
  – Allocates some portion of the bandwidth so that the transfer finishes before a specified deadline (100 sec for our experiments)
  – Results in slower transfer times, compared to AB, but increases the network availability

---

Right after the transfer

Availability decreases during the transfer

Network availability does not generally change, but available bandwidth changes
Experimental setup & validation

- Globally distributed datacenters
  - Modeled after Esnet; 100Gbps per link
  - 5 datacenters, 12 routers
  - 10% BW for background traffic
  - Solar traces from UCSD
  - Wind traces from NREL

- Simulator estimation error <8%

- Green energy supply is ~80% of the energy need per router, 1.6 kW, where available:

<table>
<thead>
<tr>
<th>Location</th>
<th>Type</th>
<th>Location</th>
<th>Type</th>
<th>Location</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chicago</td>
<td>Wind</td>
<td>New York</td>
<td>Wind</td>
<td>San Francisco</td>
<td>Solar &amp; Wind</td>
</tr>
<tr>
<td>Atlanta</td>
<td>Solar</td>
<td>San Diego</td>
<td>Solar</td>
<td>Denver</td>
<td>-</td>
</tr>
<tr>
<td>Kansas</td>
<td>-</td>
<td>El Paso</td>
<td>Solar</td>
<td>Houston</td>
<td>Solar</td>
</tr>
<tr>
<td>Nashville</td>
<td>Wind</td>
<td>Cleveland</td>
<td>Wind</td>
<td>Washington DC</td>
<td>-</td>
</tr>
</tbody>
</table>

- Jobs run in vGreen VMs on Nehalem servers in datacenter container
  - Rubis used for services with 100ms 90th%) response time constraint
  - MapReduce used for batch jobs with 10% max job completion time reduction (max 5 cores on ↑)
  - VM migration enabled
Load Balancing Effects

● Without load balancing:
  ● Average MapReduce job completion time is 22.8 min
● With global datacenter load balancing:

<table>
<thead>
<tr>
<th>Metric</th>
<th>AB</th>
<th>NB</th>
<th>Metric</th>
<th>AB</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ave. MR job completion (min)</td>
<td>17.5</td>
<td>16.8</td>
<td>TotP&lt;sub&gt;ave&lt;/sub&gt;</td>
<td>85%</td>
<td>83%</td>
</tr>
<tr>
<td>Ave. MR task completion (min)</td>
<td>4.22</td>
<td>4.25</td>
<td>BW&lt;sub&gt;ave&lt;/sub&gt;</td>
<td>66</td>
<td>48</td>
</tr>
</tbody>
</table>

- TotP<sub>ave</sub>: Average power consumption per router (ratio to peak power)
- BW<sub>ave</sub>: Average bandwidth per link in Gbps

- 30% better job completion compared to w/o load balancing
- Necessary Bandwidth (NB) leads to 1.5x more bandwidth
  - Can be used for additional transfers
  - Power consumption is similar to AB due to high idle power
  - 34% of the tasks are executed in a remote center, 5% higher than AB
Changing the bandwidth

- Power/performance does not change between 50-100 Gbps
- MapReduce completion time gets closer to no load balancing case with decreasing bandwidth
  - Explains why load balancing is not common today with 10 Gbps links
Advantages of GEAR are more pronounced with increasing router energy proportionality.

Ideally proportional router + green energy + GEAR leaves routers consuming only 3% brown power.
Bandwidth Usage Efficiency

Average bandwidth utilization efficiency per brown energy

- GEAR is better than SPR
  - 2x with step function prop
  - 2.5x with smooth prop
  - 3x with ideal prop

- GEAR + proportionality is better than non-proportional case
  - 8x with step function prop
  - 11x with smooth prop
  - 31x with ideal prop
MapReduce Performance per Watt

- GEAR is better than SPR by ~2x
- GEAR + proportionality: 27x with ideal, 10x with smooth & 7x with step proportionality
Energy Management in Large Scale Computing Systems

- Designed management strategies that exploit workload characteristics and operate across system layers
  - Optimal and adaptive power management policies
  - Efficient thermal and cooling management
  - vGreen virtualized system:
    - Able to meet SLAs while improving energy efficiency >2x

- NSF funding:
  - MRI GreenLight, IRNC ProNet: TransLight/StarLight, FlashGordon, ERC CIAN, Expedition:Variability, SHF: Cooling & Thermal Management

- Leading a large center:
  - Multiscale Systems Center funded by DARPA, DOD, & industry

- Industry funding along with IP transfer:
  - SRC, Microsoft, Google, Oracle, Intel, IBM, TI, Cisco, Qualcomm, etc.
  - Graduated students have jobs in many of these companies

System Energy Efficiency Lab
seelab.ucsd.edu