HEROS: Energy-Efficient Load Balancing for Heterogeneous Data Centers

Mateusz Guzek, Dzmitry Kliazovich and Pascal Bouvry

SnT & CSC, University of Luxembourg, Luxembourg
Summary

1. Context & Motivations
2. Proposed solution – HEROS
3. Experimental Evaluation
4. Conclusions
Energy Consumed by Data Centers

The average data center power consumption was 2.6 MW in 2013, up from 2.2 MW in 2012.

Source: “N. America Campos Survey Results”, Digital Reality, 2013
Taxonomy of CC Resource Management

Cloud Computing Resource Management

Static
Mapping
Where?
Cloud Brokering
VM Placement
Service Placement

Dynamic
Scheduling
Where and when?
Workflow Scheduling

Load Balancing
What to do now?
Server Load Balancing

Capacity Planning
What to do in the future?
Cloud Capacity Planning

Cloud Computing Resource Management

Static

Mapping

Where?

Scheduling

Where and when?

Load Balancing

What to do now?

Capacity Planning

What to do in the future?

Dynamic

Cloud Brokering

VM Placement

Service Placement

Workflow Scheduling

Server Load Balancing

Cloud Capacity Planning

Sample CC Application Stack

Multiple resource types
Virtualization

Cloud
Application
Virtual
Machine
Computing
Node

VM1
C M S N

VM2
C M S N

VM3
C M S N

Task 1
Task 2
Task 3
Task 4
Task 5
Task 6

CMMSN

CMMSN

CMMSN

CMMSN

CMMSN

CMMSN
Sample CC Application Stack

Multiple resource types
Virtualization

Computing Node

Virtual Machine

Cloud Application

VM1

VM2

VM3

Task 1

Task 2

Task 3

Task 4

Task 5

Task 6

Cloud Application

HEROS: Energy-Efficient Load Balancing for Heterogeneous Data Centers
Summary

1 Context & Motivations

2 Proposed solution – HEROS

3 Experimental Evaluation

4 Conclusions
Dynamic Independent Task Scheduling

- Schedule Software-as-a-Service (SaaS) tasks on a set of servers in a data center
- Tasks have deadlines, input and output communications
- Dynamic: tasks release times are not known in advance
- $M/D/n$ queueing model

Objectives

Minimise:

- mean response time
- energy consumption
- no failed tasks
**HEROS principles**

**HEROS** – Heterogenous Energy-efficient Resource allocation

Optimizing Scheduler

1. Immediately assign incoming tasks to servers

   → Response time reduction.

2. The selection of server is based on its score.
   - Complexity: $O(n)$ or $O(\log n)$

3. Score can be calculated locally on each server.
   - Complexity: $O(1)$, Message number: $O(m)$
HEROS principles

**HEROS** – Heterogenous Energy-efficient Resource allocation

Optimizing Scheduler

1. Immediately assign incoming tasks to servers
   → Response time reduction.

2. The selection of server is based on its **score**.
   → Complexity: $O(n)$ or $O(\log n)$
HEROS principles

HEROS – Heterogenous Energy-efficient Resource allocation Optimizing Scheduler

1. Immediately assign incoming tasks to servers
   - Response time reduction.

2. The selection of server is based on its score.
   - Complexity: $O(n)$ or $O(\log n)$

3. Score can be calculated locally on each server.
   - Complexity: $O(1)$, Message number: $O(m)$
Servers Characteristics

Servers are heterogenous and vary in consumed power:

![Graph showing the relationship between MIPS and Watt for Normal, HPC, and Micro servers.](image)

CloudCom13

HEROS: Energy-Efficient Load Balancing for Heterogeneous Data Centers
Performance per Watt (PpW) Metric

PpW is based on the application performance $Perf_s(l)$ function:

$$PpW_s(l) = \frac{Perf_s(l)}{P_s(l)}$$  \hspace{1cm} (1)
Server Selection

$PpW_s(l)$ is modified by sigmoid to avoid servers overloading:

$$H_s(l) = PpW_s(l) \cdot (1 - \gamma \cdot \frac{1}{1 + e^{-\alpha \max_l (l - \beta \cdot \max_l)}})$$  \hspace{1cm} (2)
Communication potential

Function based on the load of the top-of-the-rack switch:

\[ Q(u) = e^{-\left(\frac{2u}{U_{\text{max}}}\right)^2} \]
Finally, the server selection is multiplied by communication potential:

\[ F_s(l, u) = H_s(l) \cdot e^{-\left(\frac{2u}{U_{max}}\right)^2} \]
At startup all are scores equal to 0:
**Implementation details**

1. At startup all are scores equal to 0:
   - Use maximal server PpW as its score.
Implementation details

1. At startup all are scores equal to 0:
   - Use maximal server PpW as its score.

2. Ties (e.g. idle servers of the same type):
Implementation details

1. At startup all are scores equal to 0:
   - Use maximal server PpW as its score.

2. Ties (e.g. idle servers of the same type):
   - Random choice (balances load among racks)
Summary

1. Context & Motivations

2. Proposed solution – HEROS

3. Experimental Evaluation

4. Conclusions
Benchmark description

- Three-tier data center network topology
- Tasks are generated using exponential distribution.
- 50 independent runs
- Homogeneous and heterogeneous scenarios
  - up to 1536 computing nodes
  - up to 348497 tasks (5808 tasks/s)
# Experimental scenarios

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Small</th>
<th>Full-scale</th>
<th>Small Hetero.</th>
<th>Full-scale Hetero.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Switches</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Aggregation Switches</td>
<td>2</td>
<td>16</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Access Switches</td>
<td>3</td>
<td>64</td>
<td>3</td>
<td>64</td>
</tr>
<tr>
<td>Servers in a Rack</td>
<td>48</td>
<td>3</td>
<td>48</td>
<td>3</td>
</tr>
<tr>
<td>Total Servers</td>
<td>144</td>
<td>1536</td>
<td>144</td>
<td>1536</td>
</tr>
<tr>
<td>Commodity Servers</td>
<td>144</td>
<td>1536</td>
<td>48</td>
<td>512</td>
</tr>
<tr>
<td>HPC Servers</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>128</td>
</tr>
<tr>
<td>Micro Servers</td>
<td>0</td>
<td>0</td>
<td>84</td>
<td>896</td>
</tr>
<tr>
<td>Avg. Submitted Tasks</td>
<td>32760</td>
<td>348497</td>
<td>21976</td>
<td>233783</td>
</tr>
<tr>
<td>Simulation Time</td>
<td></td>
<td></td>
<td>60 s</td>
<td></td>
</tr>
<tr>
<td>Target System Load</td>
<td></td>
<td></td>
<td>30 %</td>
<td></td>
</tr>
</tbody>
</table>
Results – Small Homogenous Data Center

32760 tasks, 546 tasks/s
Results – Full-Scale Homogenous Data Center

348497 tasks, 5808 tasks/s
### Results – Small Heterogenous Data Center

<table>
<thead>
<tr>
<th></th>
<th>HEROS</th>
<th>DENS</th>
<th>Green</th>
<th>RoundRobin</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Energy</td>
<td>0.00</td>
<td>0.25</td>
<td>0.50</td>
<td>0.75</td>
<td>1.00</td>
</tr>
<tr>
<td>Servers Energy</td>
<td>21976</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean Response Time</td>
<td></td>
<td>21 / 26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response Time sd</td>
<td></td>
<td></td>
<td>367 tasks/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unfinished Tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

21976 tasks, 367 tasks/s
Results – Full-Scale Heterogenous Data Center

<table>
<thead>
<tr>
<th>Total Energy</th>
<th>Servers Energy</th>
<th>Mean Response Time</th>
<th>Response Time sd</th>
<th>Unfinished Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEROS</td>
<td>DENS</td>
<td>Green</td>
<td>RoundRobin</td>
<td>Random</td>
</tr>
<tr>
<td>0.00</td>
<td>0.25</td>
<td>0.50</td>
<td>0.75</td>
<td>1.00</td>
</tr>
</tbody>
</table>

233783 tasks, 3896 tasks/s
Summary

1 Context & Motivations

2 Proposed solution – HEROS

3 Experimental Evaluation

4 Conclusions
Contributions

1. Scalable decision making:
   - Distributed, low complexity \((O(1))\) selection function calculation
   - Low complexity server selection \((O(n))\)

2. State-of-the-art behaviour in homogenous data centers

3. Significant energy-savings in heterogenous data centers

4. Modular design
   - possibility to include other metrics/objectives
Future Work

- Implementation in real systems
- Cooperation between schedulers of different applications
- Including other resource types in decision making
  - Storage
  - Memory
  - Specialized hardware (GPUs, FPGAs)
Thank you for your attention!

HEROS code is freely available:

https://greencloud.gforge.uni.lu

GreenCloud 2