iPOEM: A GPS Tool for Integrated Management in Virtualized Data Centers

Hui Zhang¹, Kenji Yoshihira¹, Ya-Yunn Su², Guofei Jiang¹, Ming Chen³, Xiaorui Wang³

1. NEC Laboratories America
2. National Taiwan University
3. University of Tennessee
Virtualized data centers: server consolidation and green IT

- **Server consolidation** - virtualization facilitates consolidation of several physical servers onto a single high end system
  - Reduces management costs/overheads
  - Increases overall utilization

- **Green IT** - computing more, consume less
  - Improving infrastructure efficiency
  - Increasing IT productivity

Today

\[
DCiE = \frac{IT \text{ load power}}{\text{Total data center input power}}
\]

Future

\[
DCpW = \frac{\text{Data center useful work}}{\text{Total facility power}}
\]

DCiE: Data center infrastructure efficiency

DCpW: Data center performance per Watt
Virtualized data center management

- Server utilization based performance and power management mechanisms
  - VMware DPM, NEC SSC, IBM Tivoli…
iPOEM: a middleware for integrated power and performance management

- Features declarative management methodology
  1. accepts higher-level management objectives
     - target system status set on individual management components
  2. generates low-level management configurations.
     - Configuration settings of individual management components

- A GPS tool is a good metaphor.

  driver • How can I go to NYC?

  car • How can I get the system to 20% less power cost?

  operator

  system

  1. Map
  2. Direction

  GPS device

  1. System status
  2. Management decisions

  iPOEM
Research Goal

Management challenges

Data center administrator

Human-friendly management interfaces

This paper

System Complexity

Workload Dynamics

System Dynamics

Performance management

Power management

Application management

Thermal management

Virtualized data center

iPOEM

failure

migration

remove/add
iPOEM APIs

**API 1**: `get_position()`

**Input**: Management Configuration
- (Time start, Time end)
- Workload (reshaping-scheme)
- VM-server map, resource inventory

**Output**: System Status

**API 2**: `put_position()`

**Input**: Target Performance & Power
- (Time start, Time end)
- Workload (reshaping-scheme)
- VM-server map, resource inventory

**Output**: Management Configuration

System status is described in 3 metrics

- **Performance cost**: server overloading time in percentage.
- **Power cost**: KWatts, total power consumed.
- **Operation cost**: VM migrations.
iPOEM architecture
iPOEM management configuration engine

**API 1**: `get_position()`
System status as a function of management configurations

Fig. 7. The system status as a function of $(CPU_{low}, CPU_{high})$. 
Formal description of system status functions

Assume a homogeneous system, and the workload remains the same for different configuration settings.

**Theorem 1.** *Performance-cost(CPU\textsubscript{high}) is a non-decreasing function of CPU\textsubscript{high}.*

**Theorem 2.** *Power-cost(CPU\textsubscript{low}) is a non-increasing function of CPU\textsubscript{low}.*
iPOEM configuration searching algorithm

- A $O(\log R)$ searching algorithm
  - where $R = CPU_{max} - CPU_{min}$, the allowable load range

API: put_position().

Input: $t_{start}$, $t_{end}$, workload re-shaping scheme, Performance target, Power target, $e$ & $\alpha$ (error tolerance).

Output: management configurations ($CPU_{low}$, $CPU_{high}$), operation cost & actions.

Algorithm:

1. Assign $CPU_{low} = CPU_{max}$, $CPU_{high} = CPU_{max}$.
2. $(Performance_{cost}, Power_{cost}) = get\_position(CPU_{low}, CPU_{high}, t_{start}, t_{end}, \text{workload re-shaping scheme})$.
3. If $Power_{cost} > Power_{target}$, then a subset of servers has been turned off forcedly to meet $Power_{target}$:
   - (a) (VM-server map, resource inventory) = forced\_down($Power_{target}$, $t_{start}$).
   - (b) $Power_{cost} = Power_{target}$, $CPU_{low} = CPU_{max}$.
4. Else, //start binary searching for $CPU_{low}$
   - (a) $CPU_{left} = CPU_{min}$, $CPU_{temp} = CPU_{low}$, $CPU_{right} = CPU_{low}$.
   - (b) while($CPU_{left} < CPU_{right}$)
     - (c) $CPU_{temp} = \frac{CPU_{left} + CPU_{right}}{2}$.
     - (d) $(Performance_{cost}, Power_{cost}) = get\_position(CPU_{temp}, CPU_{high}, t_{start}, t_{end}, \text{workload re-shaping scheme})$.
     - (e) If ($Power_{cost} > Power_{target}$)
       - (f) $CPU_{left} = CPU_{temp}$, $CPU_{left} + 1$.
       - (g) Else if ($Power_{cost} < Power_{target}$ - $\frac{e}{2}$)
         - (h) $CPU_{right} = CPU_{temp} - 1$.
     - (i) Else, //find the configuration for $CPU_{low}$
       - (j) $CPU_{low} = CPU_{temp}$, break;
5. $CPU_{left} = CPU_{low}$, $CPU_{temp} = CPU_{high}$, $CPU_{right} = CPU_{high}$;
   - (a) while($CPU_{left} < CPU_{right}$)
   - (b) repeat binary searching for $CPU_{high}$
6. return $(CPU_{low}, CPU_{high})$, and the corresponding operation cost & actions.
iPOEM prototype implementation
iPOEM System positioning services

- Position reporting
- Destination searching
Evaluation: data center workload traces

- Traces on 2525 servers from 10 IT systems
  - Each is regarded as a VM in the simulations.
- Monitoring data: CPU utilization.
- 1 week length, 15 minute monitoring frequency
  - 672 time points
Evaluation: methodology

- Run the iPOEM prototype as an offline engine.
  - It is driven by the data traces stored in the monitoring database, and emulates the integrated management in a virtualized data center hosting the 2,525 servers as VMs.
- The system and management configuration settings
  - Performance manager and power manager
    - Implementation of the simplified schemes in NEC SigmaSystemCenter middleware
    - The default $<\text{CPU}_{\text{low}}, \text{CPU}_{\text{high}}>$ setting is $<40\%, 80\%>$. 
    - The physical servers are homogeneous with the same CPU specs
      - 3GHZ Quadra-core (the most common CPU model in the traces).
    - Performance cost: number of performance violation in a time epoch
      - A server has a performance violation at a time point when its CPU utilization is larger than a threshold (90% in the paper).
    - Power cost: we assume power consumption per server is either 0 (power-off mode) or 200Watts (power-on mode), simplified on the power model profiled in the local testbed.
    - Operation cost: the number of VM migrations that the performance and power managers need to execute for the server load configuration enforcement.
iPOEM engine performance

iPOEM engine response time to service requests

---

iPOEM engine performance

destination searching
position reporting

iPOEM engine response time to service requests
iPOEM auto-piloting service

- Sensitivity based optimization [Markovic et al. 2004]

\[
\min \text{ Power(configs)} \quad \text{s.t.} \quad \text{Performance(configs)} \leq P_{th}
\]

where \( P_{th} \) is the upper bound of the performance cost.

Service: Auto-piloting.
Input: \((m \times n)\)-grid map, migration cost threshold \( t \)
Output: management configurations \((CPU_{low}, CPU_{high})\).
Algorithm:
1. Prune all grid nodes with migration cost \( > t \) in the map.
2. If no node remains, return the current configuration.
3. Else, for the remaining grid nodes, calculate the cost sensitivity on each \( x \) with the configuration \((CPU_{low}^x, CPU_{high}^x)\) as:
   
   \[
   \text{Sensitivity}(x) = \left| \frac{\Delta \text{Performance}_\text{cost}}{\Delta CPU_{low}^x} - \frac{\Delta \text{Performance}_\text{cost}}{\Delta CPU_{high}^x} \right|
   \]

4. Pick the grid node \( x \) with the minimal \( \text{Sensitivity}(x) \) value, return \((CPU_{low}^x, CPU_{high}^x)\).
# iPOEM auto-piloting evaluation

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Performance cost</th>
<th>Power cost</th>
<th>Migration cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-piloting</td>
<td>1.55% (4.38%)</td>
<td>20.88 (9.9)</td>
<td>37.7 (53)</td>
</tr>
<tr>
<td>Static - (10%, 10%)</td>
<td>0</td>
<td>96 (13.5)</td>
<td>342 (510)</td>
</tr>
<tr>
<td>Static - (10%, 90%)</td>
<td>2.26% (3.66%)</td>
<td>24.68 (5.08)</td>
<td>23.4 (55.4)</td>
</tr>
<tr>
<td>Static - (90%, 90%)</td>
<td>4.46% (3.93%)</td>
<td>16.1 (2)</td>
<td>118.4 (407.2)</td>
</tr>
</tbody>
</table>

**Comparison of Auto-piloting and three static configuration schemes**

![Auto-piloting System Configuration Evolution](image-url)

**Auto-piloting management configuration evolution**
Conclusions & Future Work

• iPOEM, an integrated power and performance management middleware in an virtualized infrastructure.
  – human-friendly interfaces for multi-objective management.

• Future work
  – Meta-management integrating more objectives.
    • explosive growth of the system state space
  – mashup services for customized tenant management
    • new API designs
Thank you.

• Questions?
Appendix