Open. Together.
Power Capping with Cloud Super Apps Performance SLA

Speakers:
Justin Song @ Alibaba
Edmund Song, Nishi Ahuja @Intel

Contributors:
Hao Zhu, Guan Wang, Xiaolin Meng @ Alibaba
Feng Jiang, Mohan J. Kumar, Xiaoguo Liang, John Leung @ Intel
Agenda

- Alibaba Power Capping with Performance SLA
- Fine Granularity Power Management Knobs
- Redfish Adoption and Practices
- Call to Action
Alibaba Power Capping with Performance SLA
Alibaba Power Management Architecture

- **Apps driven**
- **Run-time management**
- **Availability first, efficiency second**

### Case Studies

- **Power Management**
- **Apps**
- **Infrastructure**

### Diagram Overview

- **Data & AI**
  - Power Thermal Aware Scheduling
  - Cooling Optimization
  - Power Delivery Optimization
  - Data collection

- **P/T Resource Management**
  - Health Management
  - Notification & Alarming
  - Anomaly Recovery & Isolation
  - Quota & Budgeting
  - Anomaly Detection
  - Power Capping

- **Physical Layer**
  - Power Agent

- **Operational Zones**
  - ECS, Storage, DB, Fuxi, Hybrid, sigma, OpenAPI, Operation

- **Public/Private/Hybrid Cloud & In-House Apps**
Working with Apps

Case Studies

Open. Together.
Power Capping Triggering

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>85%, 3min Rebalance</td>
</tr>
<tr>
<td>B</td>
<td>100%, 3min Stop adding instances</td>
</tr>
<tr>
<td>C</td>
<td>110%, 3min Super apps flow control</td>
</tr>
<tr>
<td>D</td>
<td>120%, 5min Power capping with performance SLA</td>
</tr>
</tbody>
</table>
## Performance SLA

<table>
<thead>
<tr>
<th>Type</th>
<th>Target</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td></td>
<td>Align w/ apps</td>
</tr>
<tr>
<td>Service delay</td>
<td>1s</td>
<td>Local</td>
</tr>
<tr>
<td></td>
<td>30s</td>
<td>Global</td>
</tr>
<tr>
<td>Models coverage</td>
<td>Based on spec &amp; test results</td>
<td></td>
</tr>
<tr>
<td>Racks coverage</td>
<td>Based on spec &amp; test results</td>
<td></td>
</tr>
<tr>
<td>Power watermarks</td>
<td>Defined by apps &amp; platform</td>
<td></td>
</tr>
<tr>
<td>Capping accuracy</td>
<td>5%</td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td>Defined by apps</td>
<td>Low priority nodes first capped</td>
</tr>
<tr>
<td>Fmin</td>
<td>Defined by apps, Lifted by AI</td>
<td>Anytime higher than Fmin</td>
</tr>
<tr>
<td>Granularity</td>
<td>By core (CPU), rank (mem), link (IO) and device (storage)</td>
<td></td>
</tr>
<tr>
<td>Capping - DVFS</td>
<td>Minimal performance impact</td>
<td>Defined by apps</td>
</tr>
<tr>
<td>Capping - CCx</td>
<td>Minimal latency impact</td>
<td>Defined by apps</td>
</tr>
<tr>
<td>In-Band</td>
<td>supported</td>
<td></td>
</tr>
<tr>
<td>Out-of-Band</td>
<td>Partially supported</td>
<td></td>
</tr>
<tr>
<td>Thermal watermarks</td>
<td>Defined by apps and platform</td>
<td></td>
</tr>
<tr>
<td>Failover</td>
<td>Unconditional capping, autonomous capping, or SS</td>
<td></td>
</tr>
</tbody>
</table>
Results — Examples

High & Low Priority Instances Co-exist on Same CPU

Scenario: high priority instances performance guaranteed with low priority instances capped

Power capping with Fmin=2.2GHz, Cap=250W

Scenario: capping with no Fmin violation

Case Studies
Fine Granularity Power Management Knobs
Cloud Power-Performance Requirement

- Rack density
- Utilization of provisioned resource

Capex Optimization

- Power-Performance proportionality (e.g. SLA matched energy efficiency)
- Performance-per-watt efficiency

Opex Optimization

- Convergence between infrastructure plane and resource plane
- Hardware intelligences into resource scheduling and orchestration

SLA and Reliability
Fine Grain Platform Power-Performance Knobs

Platform
Processor
Core

Power
Thermal

Frequency
Package Px/Cx states

Power
Online/Offline
Core Px/Cx states

IDC
Rack
Platform

Efficiency
Performance
Reliability

DIMM
Storage
Network

CPU

Thermal
Power
Intel Practices in Cloud Power-Performance Optimization

Intel Cloud Power and Performance Solution

Workload
- Group Power Capping
- Dynamic Core Mgmt.
- Peak Shaving via Smart BBU (Turbo Rack)
- HW Telemetry Awareness
- Workload PnP Analytic and Simulation

Cloud OS
- K8S
- Open Stack
- OpenDCM
- Proprietary Cloud OS

Platform
- RAPL/DVFS
- Px/Cx
- Core Freq.
- Core Off/On
- Un-core Freq.
- RAPL
- DIMM
- NIC, Storage

Open RMC, DC Power
Smart BBU, DC Cooling
Rack & DC Facility
Redfish Adoption and Practices
API Requirement in Cloud Power-Performance Optimization

- Interoperability - Infrastructure plane vs. Resource plane, server vs. facility (rack, IDC) etc.
- Consistent API model for In band interface and Out of band interface
- Support runtime configuration and cloud scale deployment

Starting with extensible and scalable Redfish model
Redfish Practices

- Unified power control API model to support hierarchical power capping (platform, rack, cluster)
- Consistent model to support RAPL and Smart BBS based peak shaving.
- New power line to trigger power limiting if DynamicPeakPowerCapacity is not null
- Orchestration need insight of power redundancy of managed system for intelligent scheduling policy
- Describe complex power redundancy model which is across platform, chassis and rack

Redfish based API reduce deployment complexity of data center power management
Call to Action
Call to Action

• Performance SLA driven power optimization is critical for TCO optimization and PUE efficiency. Need platform and solution co-innovation to support dynamic, flexible and workload aware optimization policies.

• Open and standard API able to reduce deployment cost in large scale cloud environment. Need collaboration to define common telemetries and control interface for OCP platform, e.g. via baseline OCP HW Mgmt. profile.

• Cloud developers and users need to understand and define their performance requirements for their cloud apps

• Get involved: https://www.opencompute.org/projects/hardware-management
Open. Together.

OCP Global Summit | March 14–15, 2019