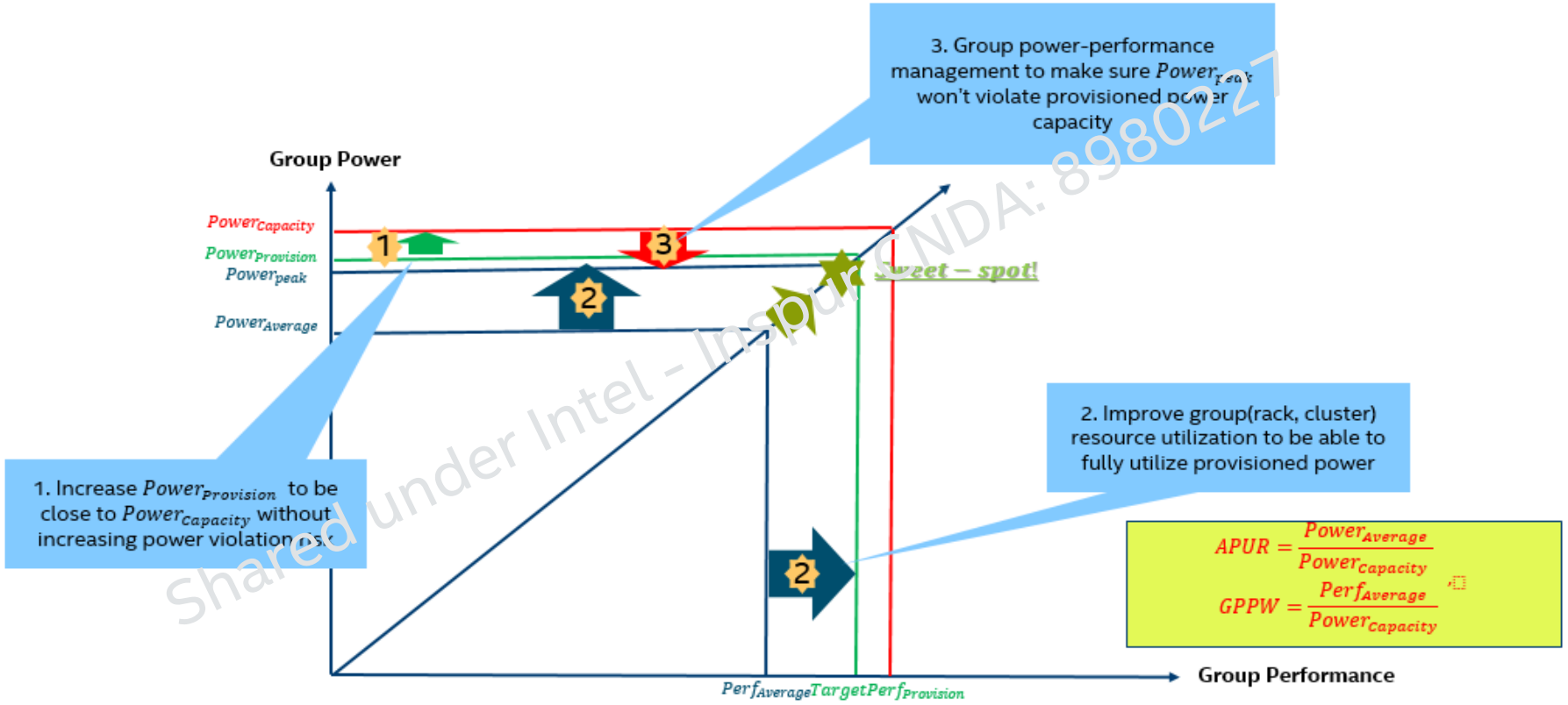
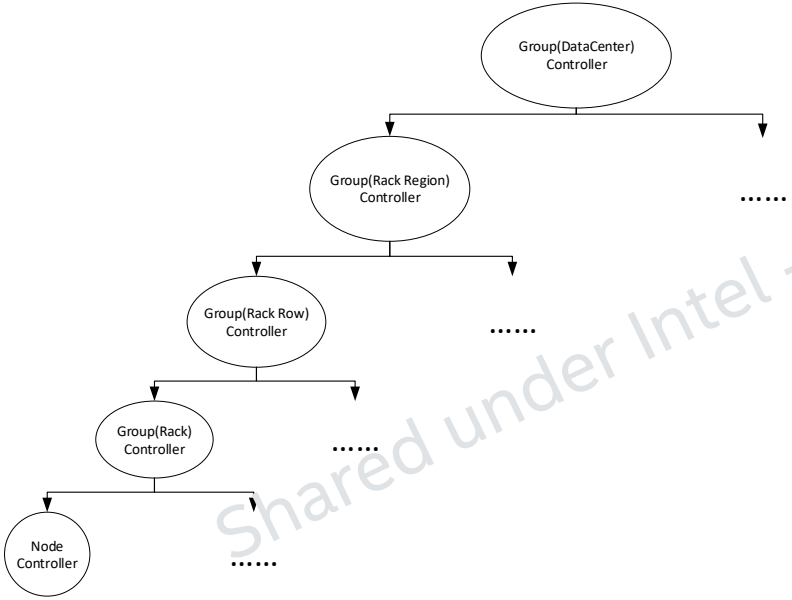
INTEL GROUP POWER CAPPING

Group power performance



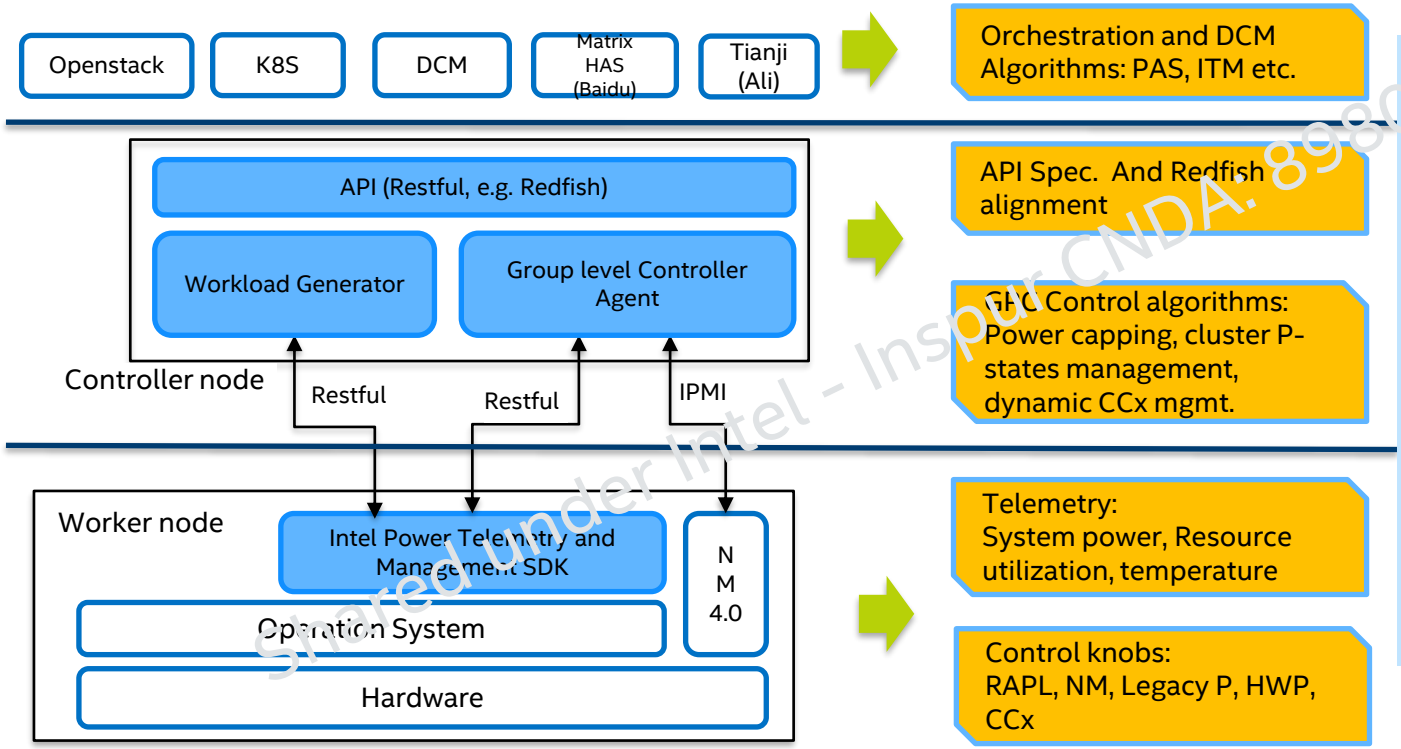
Goals: increase average power utilization ratio(APUR) xx%,
with implicated improvement (~xx%) of group level performance per provisioned watt(GPPW)

Power management hierarchy



We design a power management hierarchy based on the power delivery hierarchy in a typical modern data center. In this management hierarchy, there are 2 types of agent, node and group. A node
represents a minimal unit (mostly a physical machine) that can be controlled and monitored by its
parent group, and a group can control its children nodes and its sub-groups using our power management algorithm.
Figure 1 shows a power management hierarchy tree of a typical data center.

Intel Group Power Control (GPC) Reference Solution



- Alpha quality reference module to support quick prototype at CSP environment
- Reference design guide, algorithms, BKM for CSP customization
- Provide workload generator to simulate power runtime pattern according to historical data
- Rick PnP telemetry and control capability support on IA platform

Intel BKM, Algorithms, Design Guide, Spec.

Intel Reference Codes

POWER MANAGEMENT ALGORITHM

Terminology

Actual Power: Actual power consumption of group/node. For a physical machine node, this value can be obtained by NM/RAPL/ACPI/Power Meter. For a group, this value can be obtained by count of children's Actual Power or Power Meter.

Power Capacity: The power budget of group/node. For head group, this value is set by the admin, for others, this value is allocated by its parent. For any controller, insurance of the Actual Power under the Power Capacity takes the 1st priority in our algorithm.

Power Provision: The power that a group/node can actual use. For a node, it's the capping value, for a parent group, it's the power that it can allocate to its children.

Default Power: A value that represents the power level of baseline performance, a child's Default Power is calculated is calculated by its parents through its priority, the sum of all children's default power equals to parent's Power Provision. The application from any child to parent to get its default power takes the 2nd priority in our algorithm.

Power Pool: A pool that contains the unused power budget.

Terminology

Time Interval: The interval of power control time, and the time interval is different between different control-level.

Power Action: When there is a need for node or group to adjust its power budget, it will apply for a power action from its parent. If a power action is approved, for node, it will change its power capping value; for group, it will reallocated power to its children.

Power Action Margin: a value that, when the Actual Power reach the line (Power Provision - Power Action Margin), it triggers a power action.

Power consumption prediction & power pre-allocation

By now there is no method to mapping complex workloads and power consumption
computing cluster, the types of workloads are relatively stable, which means the
prediction and power pre-allocation based on the time-series data of Actual Pow
We import 2 time-window to do prediction: long-window and short-window.

Under two situations a child will try to apply for more power from its parent:

- Avg(short-window) > Avg(long window)+△ (2-1)
- Actual Power > Power Provision - Power Action Margin (2-2)

Where △ is a set value to block noise, and there is a cooling-off period (same to long time window time period in our example) for this in order to avoid re-trigger in a short period.

Power Pool

Any group in our power management system has a power pool to collect power unused from its children, when there is a child apply for a Power Action to get more power budget, its parent will allocate power to it from the Power Pool rapidly. When the Power Pool have no enough power, the parent will make a decision to judge whether make a power repossession from other children according to the Default Power (which is calculated by priority).
To decide whether a child has power unused, we use Max(long window), for a child when:

- Max(long window) < Power Provision - Power Action Margin +△ (2-3)

Power from child to parent and the new Power Provision of the node:

- Power to parent = Power Provision - Power Action Margin -Max(long window) (2-4)
- New Power Provision = Max(long window) + Power Action Margin (2-5)

Where △ is a set value to block noise

Priority and Default Power allocation

In our power management, when all children try to get power from parents and there is no enough power to allocate, which is the worst case. Parent will make all children running under its default power, Default power of child k is:
Default power 的公式:

Child Default Powerk = Baselinek +

[Parent Power Provision – Sum(Baseline)]×priorityk/Sum(priority) (2-6)

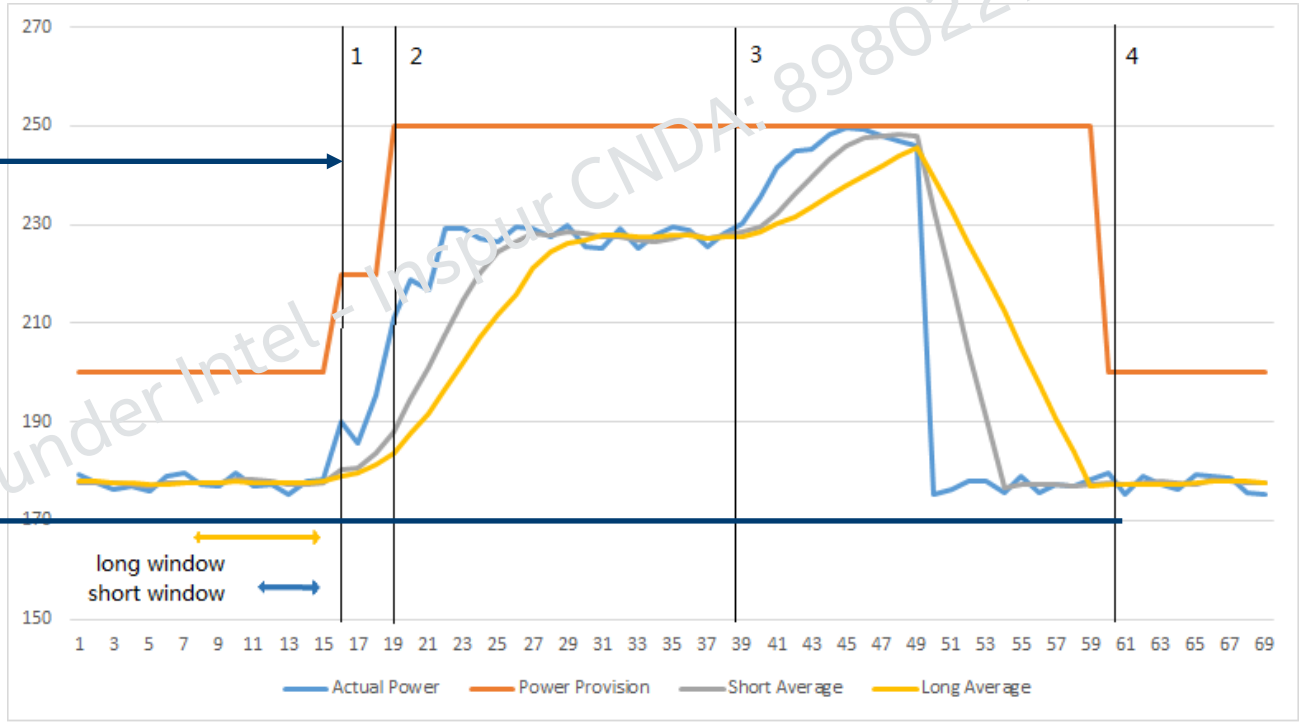
Where BaseLine and Priority shall be decided by workload type, server type and admin.

Parent Power Provision = Sum(Child Default Power) (2-7)

So when a child apply for a Power Action to get the budget of its Default Power, it can always be satisfied, if the Power Pool of its parent has no power budget, the parent controller will repossess power from other child who's Power Provision > Default Power and then sanctify the Power Action

Borrow & Lend power budget to power pool

Power Pool



Sample interval：1s

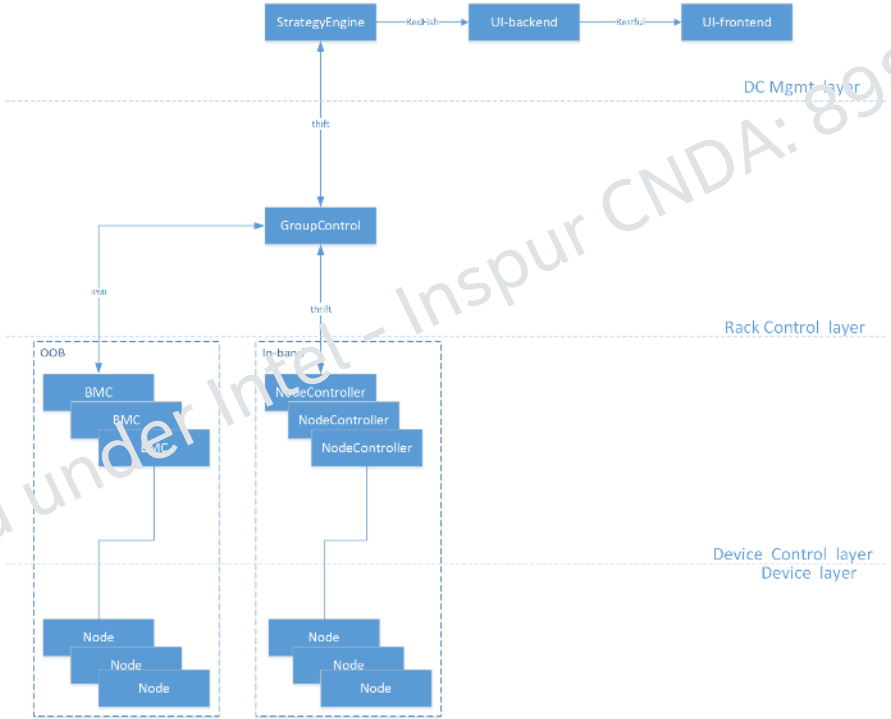
Group Power Management Strategies/Policy

Strategies	Description	Comments
Static Power Capping	Rack controller capping nodes through node manager. But static power allocating will result in that some nodes reach power capping line while there is still power margin within the rack which results in worse group performance power per watt (GPPW).	Pros: No power violation Cons: High performance impact. Low APUR & GPPW.
Conservative Dynamic Power Capping	Rack controller monitors nodes' power consumption and reallocates power as workloads change over time. Rack controller capping nodes through node manager in real time. In configuration list, cap_always = yes	Pros: Minimum power violation. Cons: Medium performance impact.
Performance- Aggressive Dynamic Power Capping	Rack controller monitors nodes' power consumption and reallocates power as workloads change over time Rack controller capping nodes through node manager only when rack overall power consumption exceeds its capacity to get the max performance, but it may cause peak transient rack power. In configuration list, cap_always = no	Pros: Low performance impact, especially fit to the scenario where workload in different servers has different peak traffic time. Cons: Potential power budget violation with limited over-shooting

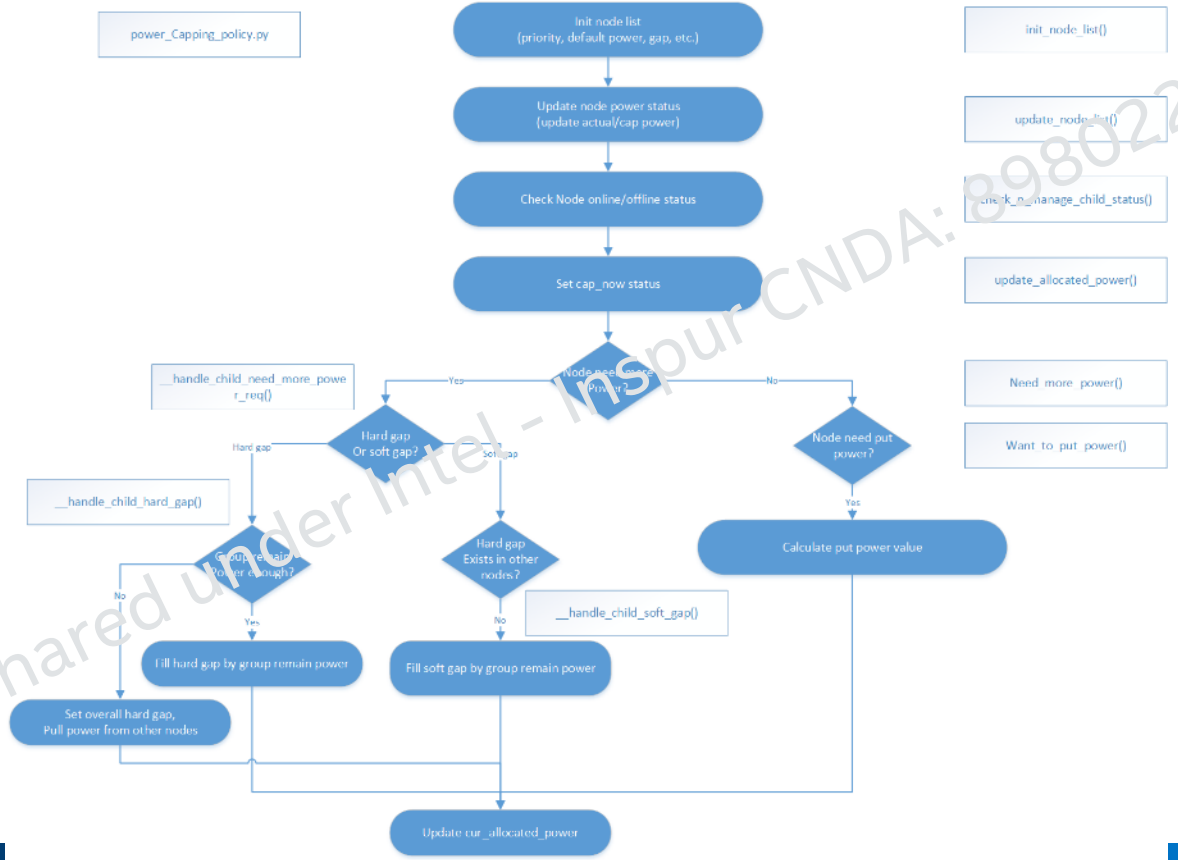
GPC-ARCH/FLOW

Shared under Intel - Inspur CNDA: 8980227

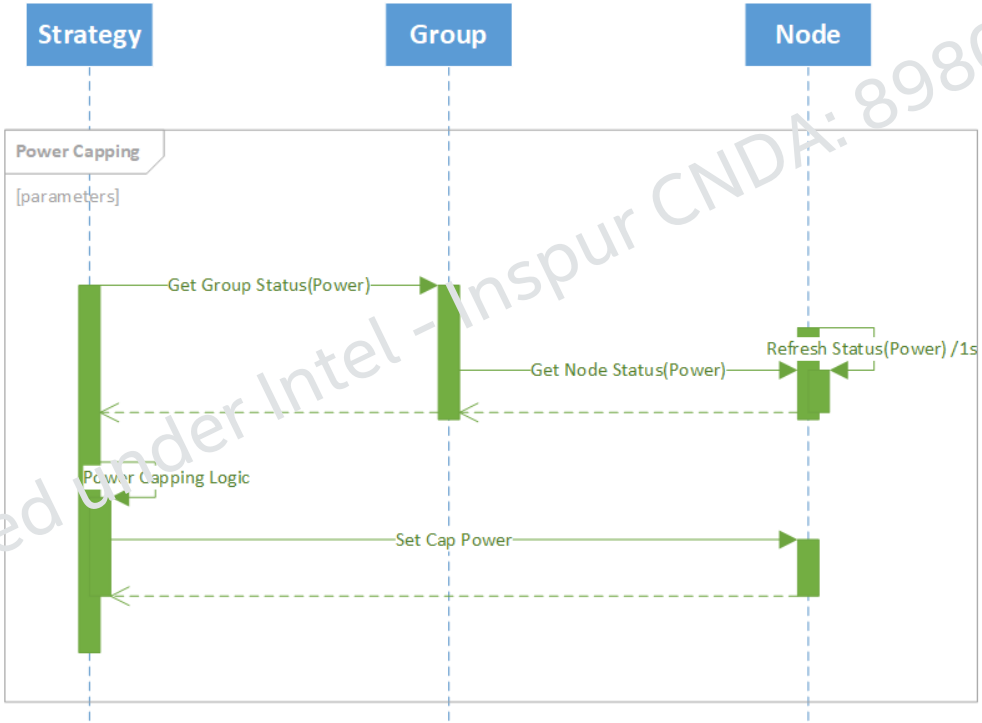
GPC - Arch



GPC-Power Capping Flow



GPC-Power Capping Sequence



POWER MANAGEMENT TEST REPORT

Case Study - Group Power Capping Evaluation

Background – One cloud service provider suffering from rack density and power over budget warning, the group power capping is enabled and we evaluated typical workload[1] for capping and uncapping with different density incremental assumptions.

UCs-

S1: The normal condition, no desnity improment as baseline.
S2: The rack runs in higher density with power violation, but no capping.
S3: The rack runs in higher density with power capping.

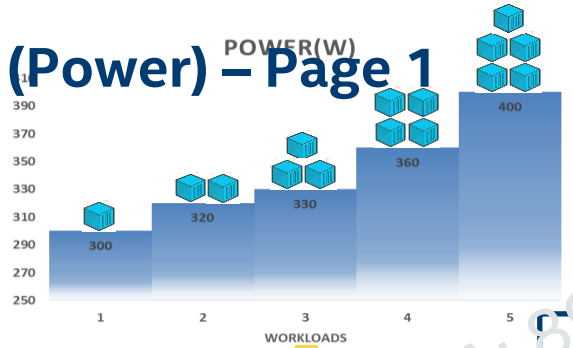
Result: Power capping system can protect the rack from contiuning power violation with limited performance degradation while increasing rack density, e.g. under 18% rack density improment, the performance downgradtion is <=3% and energy capped ratio is 1% .

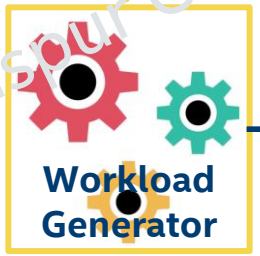
[1] - Random select 24h data and resample to 2h for test. 12 servers are for workload test

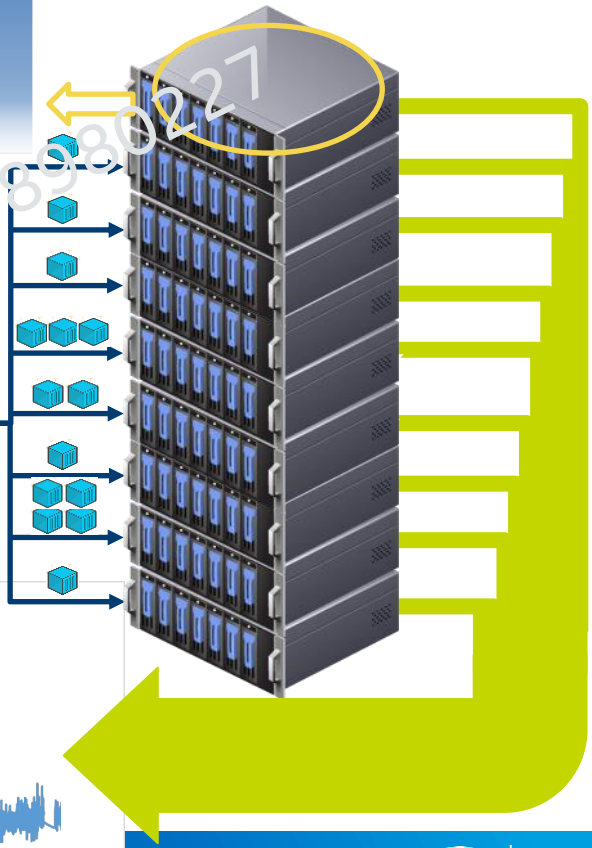
Scenario	Density	Strategy	Power Consumption	Performance	Peak Power	Average Power	Capped energy	Perf. Degradation
S1	1	-	-	-	7898	7056	-	-
S2	1.18	Uncapping	15256740	23524956.1	8676	8339		
S3	1.18	Capping	15056960	22803198.3	8596	8265	1.0473%	3.0681%
S2	1.25	Uncapping	15640260	36616505.8	8505	8148		
S3	1.25	Capping	15379220	34898682.8	8501	8310	1.6690%	4.6914%

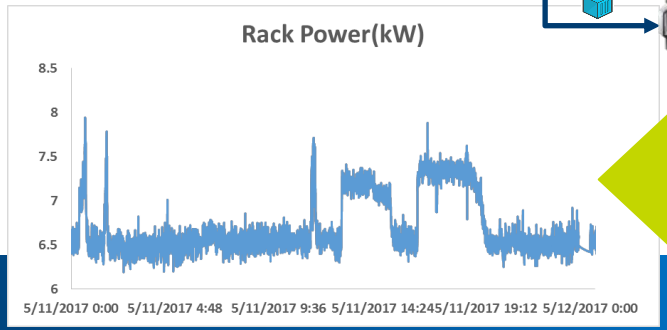
Workload Simulation Tool (Power) – Page 1

- Node-level workload-power relationships are known
- With targeted rack power, the workload generator generates different workloads for every node
- All the nodes give out the targeted rack power

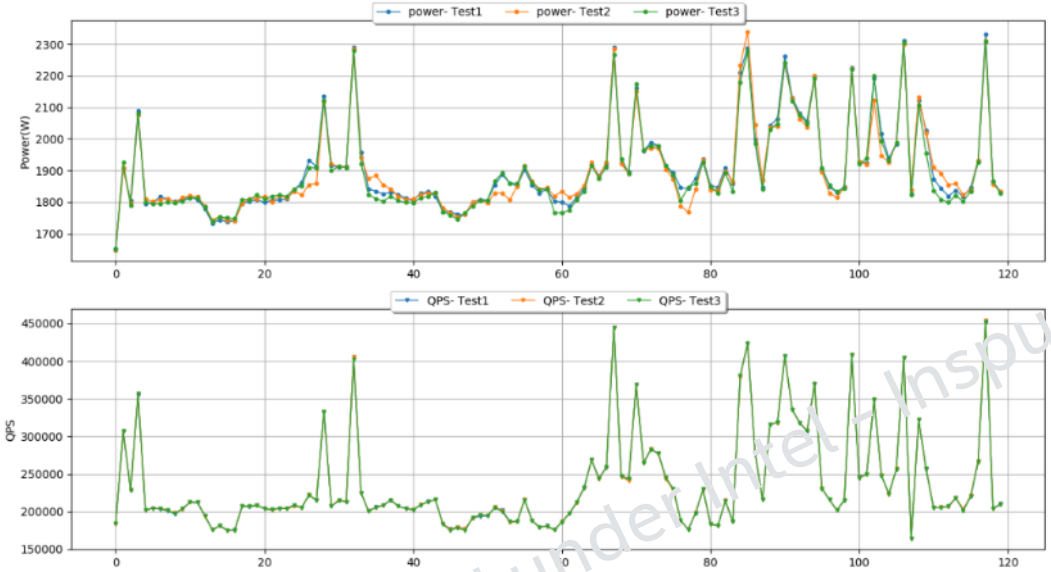








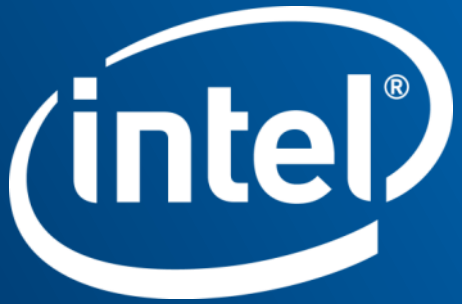
Workload Simulation Tool (Power) Page-2



TestID	Performance	Power(w*s)	Performance variance	Power varian
Test1	28393673	228364	Baseline	Baseline
Test2	28399655	228053	0.34%	1.21%
Test3	28396240	227532	0.30%	0.08%

It is necessary to evaluation the benefit of Power capping and DRPP with a high accuracy tool.
The workload simulation tool has low variances:

- **Performance variance 0.3%**
- **Power variance about 1%**


intel®

POWER MANAGEMENT SPECIAL CASE

For node lose control

1. If the node support power-capping and under our power management system:

   Give the node a power budget the system allocate to it last-time and the node will maintain this budget until its reconnection.

2. If the node not support power-capping and cannot controlled by our power management system:

   Give the node a power budget of its TDP.

Performance Impact:

A small reduction in power achieved through existing power capping methods can cause the application latency to increase uncontrollably and may even reduce throughput to zero. (risk evaluation)

Additionally, even the observed peak is only reached rarely. To avoid provisioning for capacity that will be left unused most of the time, data centers may provision for the 99th percentile of the peak power. Capping would be required for 1% of the time, which may be an acceptable hit on performance in relation to cost savings.