

Painless Cache Allocation in Cloud

Lin Yang, lin.a.yang@intel.com

Sunku Ranganath, sunku.ranganath@intel.com

Malini Bhandaru, malini.k.bhandaru@intel.com

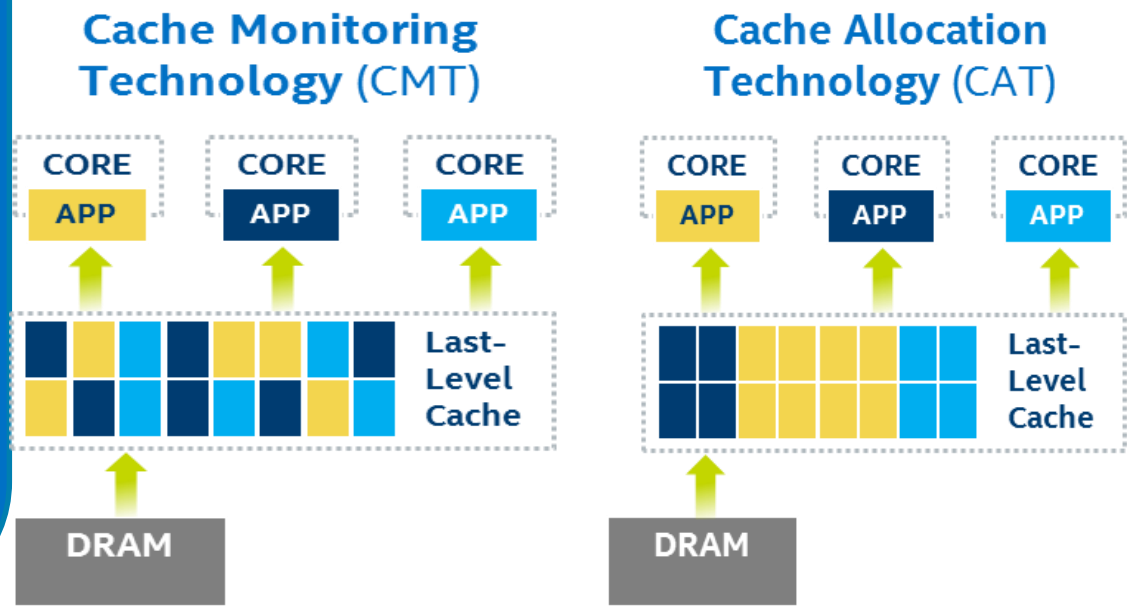
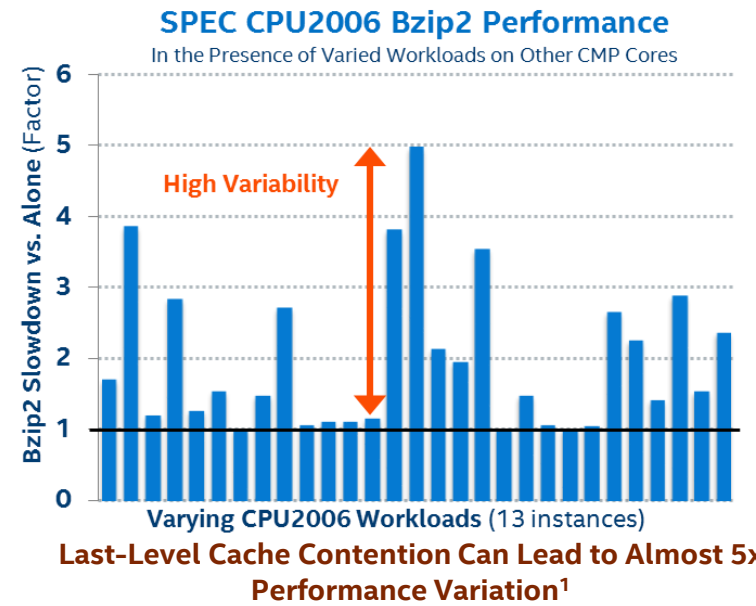
Last Level Cache Matters

- Cache resource contention introduces jitter and increased latency

NFV & RT workloads are time sensitive

Intel Resource Director Technology (RDT)

- Hardware based CPU cache control
- Allocation per process (VM, application, etc.)
- Shared and isolated usage models
- Monitoring and LLC partitioning mechanisms provide isolation and prioritization of VMs or apps
- Software Support: Resctrl fs, PQoS toolset, Resource Management Daemon



1: Source: UC Berkeley (UCB) Tests, 2016, see backup for details

Demo in Intel Booth



Last Level Cache as a Resource – NFV use case

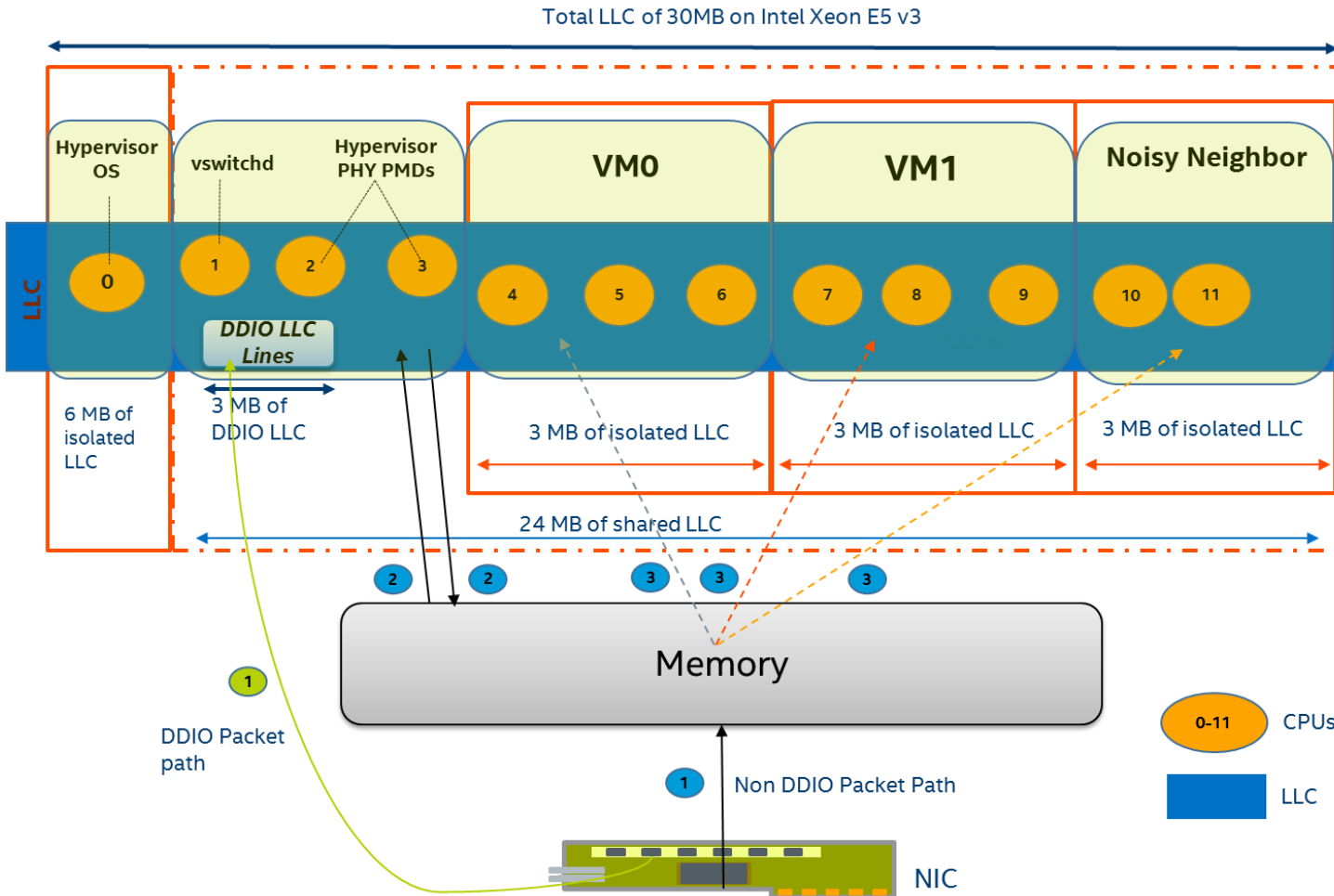


Figure: Traffic flow from NIC to VMs

Throughput Improvement of up to 46% in presence of noisy neighbor¹

Scheduling Considerations:

- Node capacity of cache
- Workload need for DDIO
- Workload sensitivity and mix
- Isolated vs. Shared allocations

Node level cache manager:

- Crucial to have local agent on the host
- Dynamic cache control
 - Millisecond timescales for time sensitive workloads

Allocation with Resource Management Daemon (RMD)

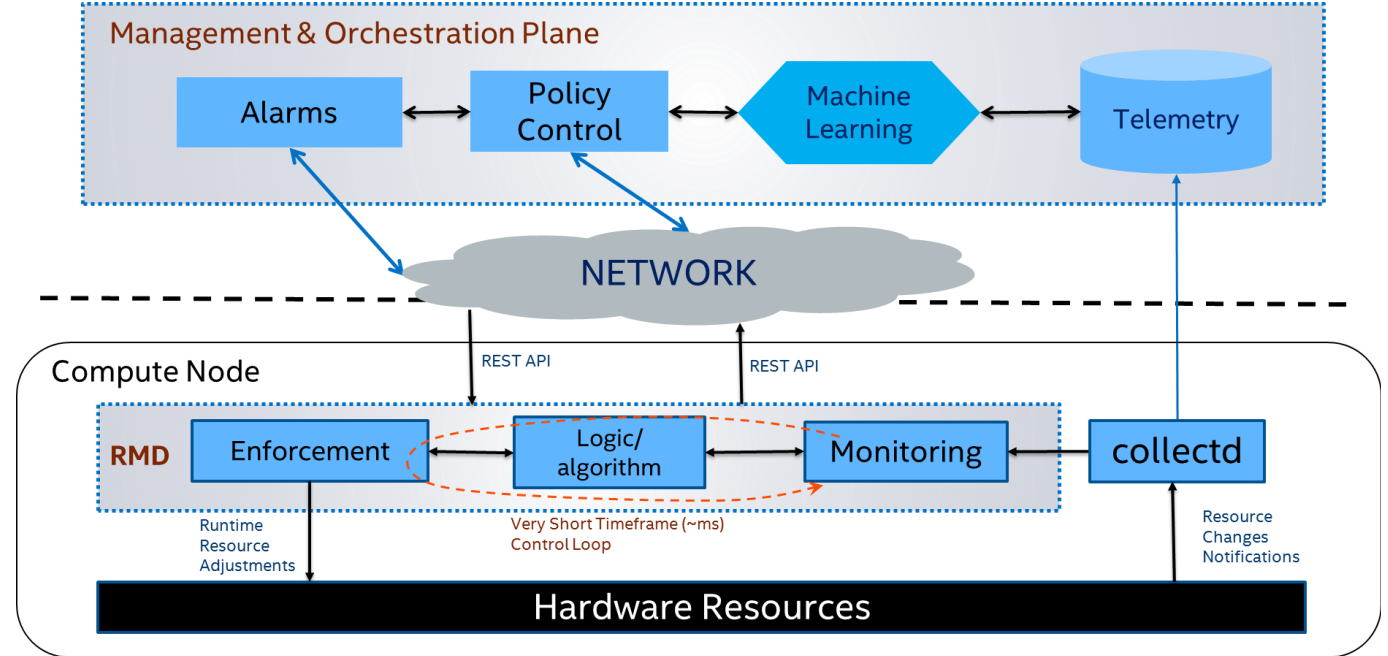
RMD - A Linux daemon that:

- Runs on individual hosts
- REST API, accessible to orchestrator
- Accepts & enforces policy

Open Source: <https://github.com/intel/rmd>

Why use RMD:

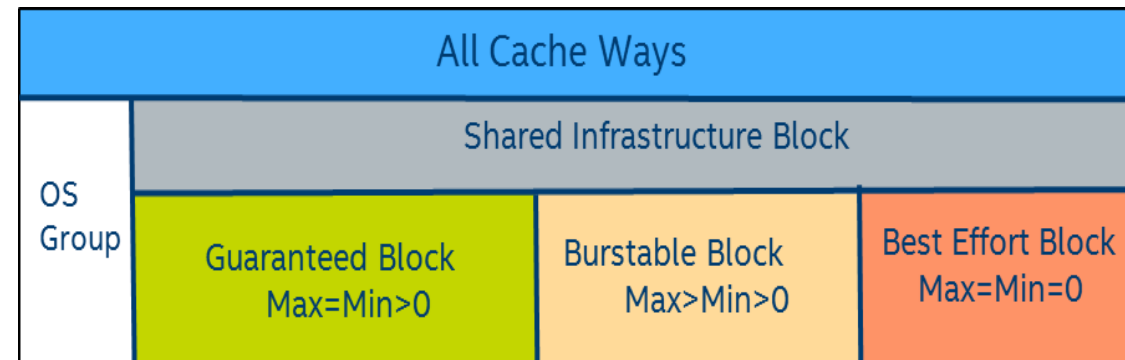
- Ability to use LLC as a resource
- Satisfies multiple usecases with varying resource policies



[CachePool]
total 30
guaranteed = 12
burstable = 12
besteffort = 3

[OSGroup]
cache = 3

[InfraGroup]
cache = 27



Design/Implementation Options

Features	Via Libvirt	Via RMD
Cache allocation & updates	Static, One Time Configuration	Dynamic, Run time updates based on the policy
Hardware Bitmasks	Operator required to understand bitmasks for complex allocations	Hides bitmasks constraints & calculations
Cache Policies	No concept of Isolated vs. Shared cache policies	Supports Isolated & Shared cache policies
Hyper-Convergence	May not be efficient	Out of the box efficient configuration
Short timeframe resource control	No	Yes
Low level resource control	Just at VM Level	Process level control

RMD is complimentary to Libvirt

OpenStack with RMD

Blueprint: Please provide feedback

<https://review.openstack.org/#/c/568678/>

Scale with RMD

- Enables secure cache allocation policies
- Manage multiple platform resources like memory bandwidth control using Intel RDT

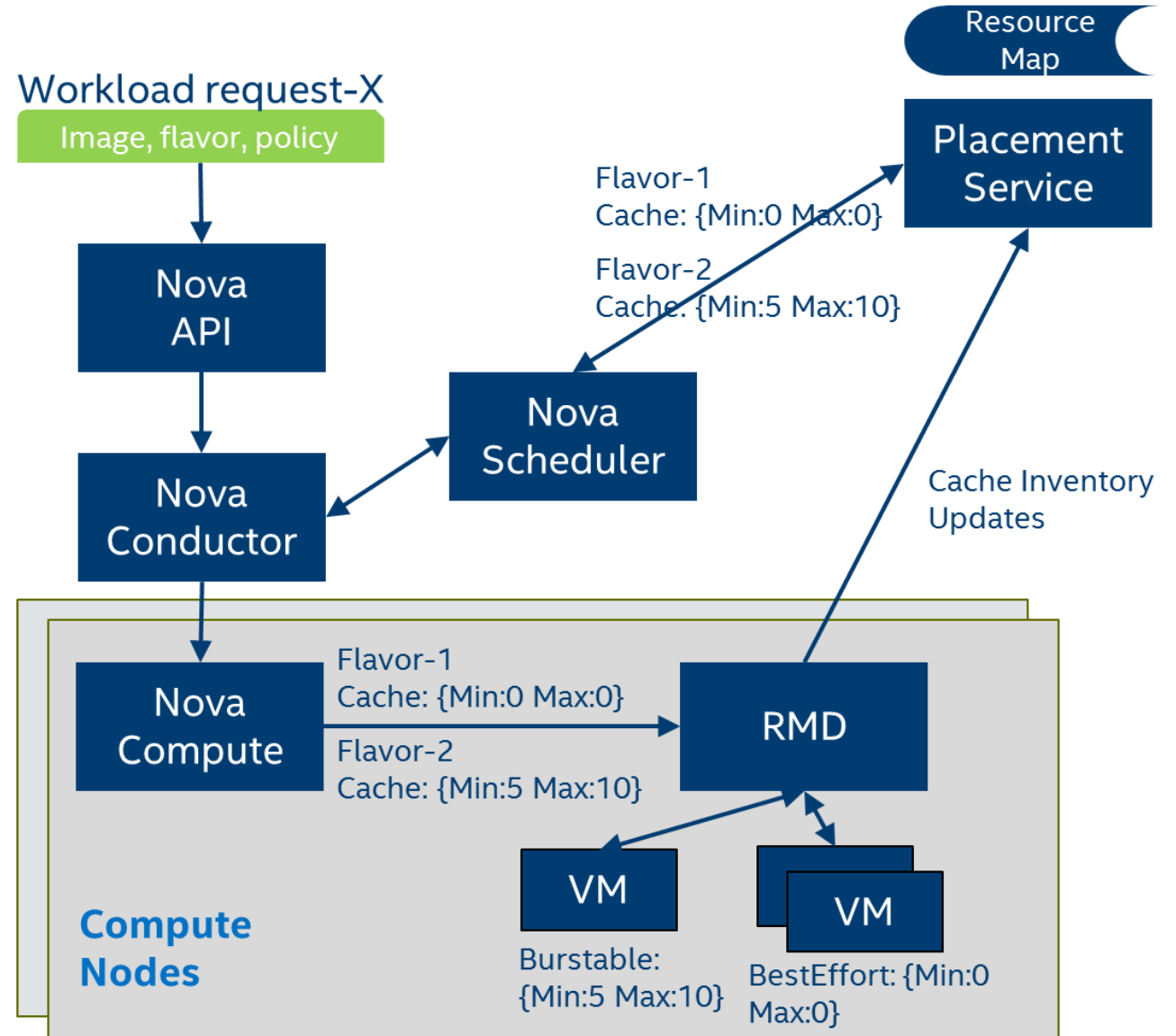


Figure: Flavor Spec with max_cache:min_cache model

Thank you