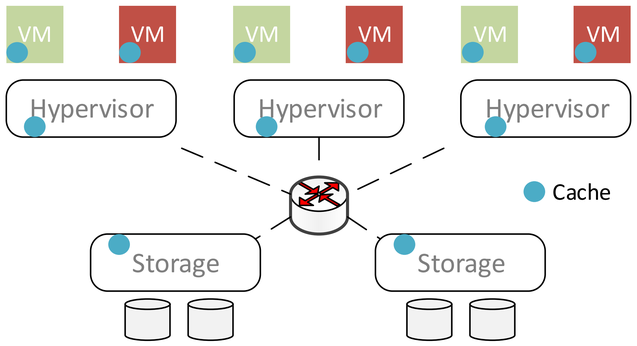
Software-Defined Caching: Managing Caches in Multi-Tenant Data Centers

Ioan Stefanovici, Eno Thereska, Greg O'Shea, Bianca Schroeder, Hitesh Ballani, Thomas Karagiannis, Antony Rowstron, Tom Talpey

Motivation

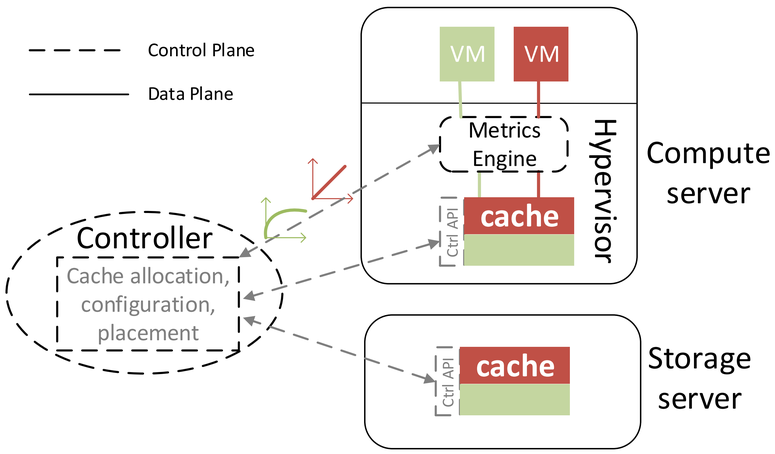


Storage caches today are not designed for multi-tenancy, and cannot be configured to match tenant or provider objectives.

- Lack of performance isolation
- Lack of customization
- Lack of coordination
- Lack of adaptability

Solution

Moirai: a tenant- and workload-aware system to control the distributed caching infrastructure.



Achieves a variety of tenant and provider objectives.

Programmable Caches

createCache (<size *s*, eviction policy *e*, write policy *w*>)
returns a reference to the newly created cache *c*
createRule (IO Header *<src, op, file, range>*, Cache *c*)
creates cache filter rule: <src, op, file, range> → c
configureCache (<size *s*, eviction policy *e*, write policy *w*>, Cache *c*)
getCacheStats (Cache *c*)
returns cache statistics

Metrics Engine

Workload characteristics:

- Hit rate curves, throughput, R/W ratio, etc.

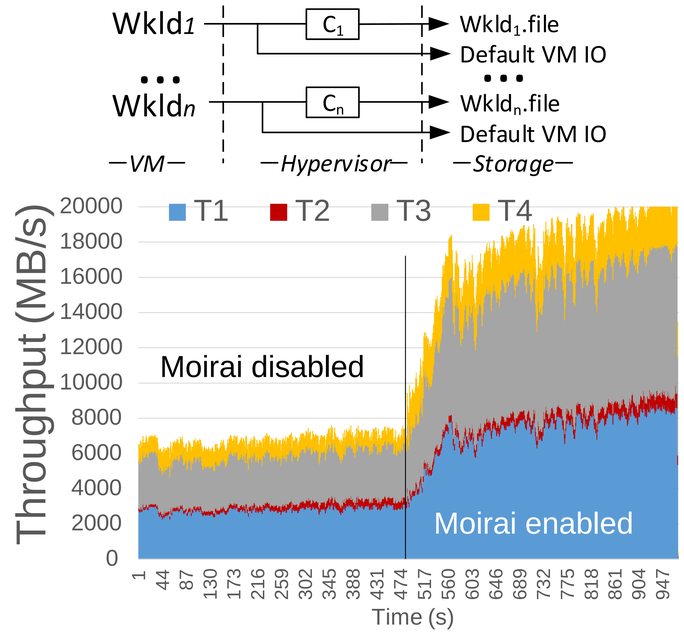
Controller

Enforces higher-level provider objectives by programming the data plane according to workload characteristics:

- Cache location, size, eviction/write policies
- Explicitly control which data is stored in each cache

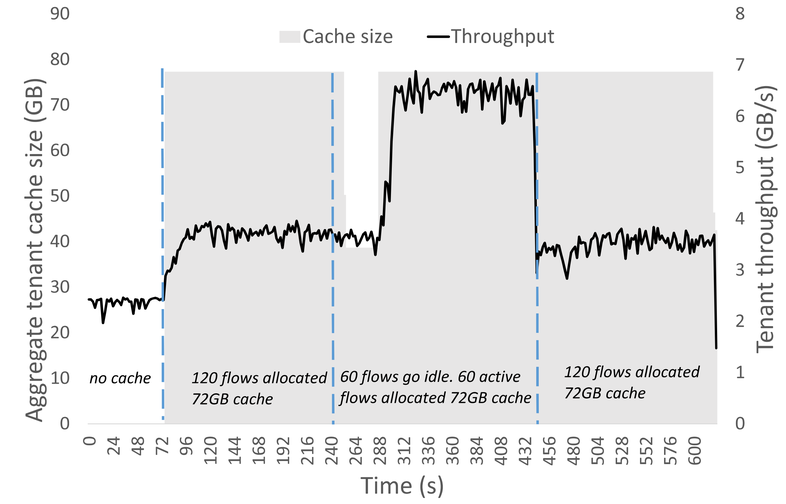
Cache Transformations on the Data Plane

Example: Per-Workload Specialized Caches



Cache Transformations on the Data Plane

Example: Adapting to Dynamic Workloads



Example: Cache Scale-out