Scale Out and Conquer: Architectural Decisions Behind Distributed In-Memory Systems

Akmal Chaudhri
Technology Evangelist
GridGain Systems

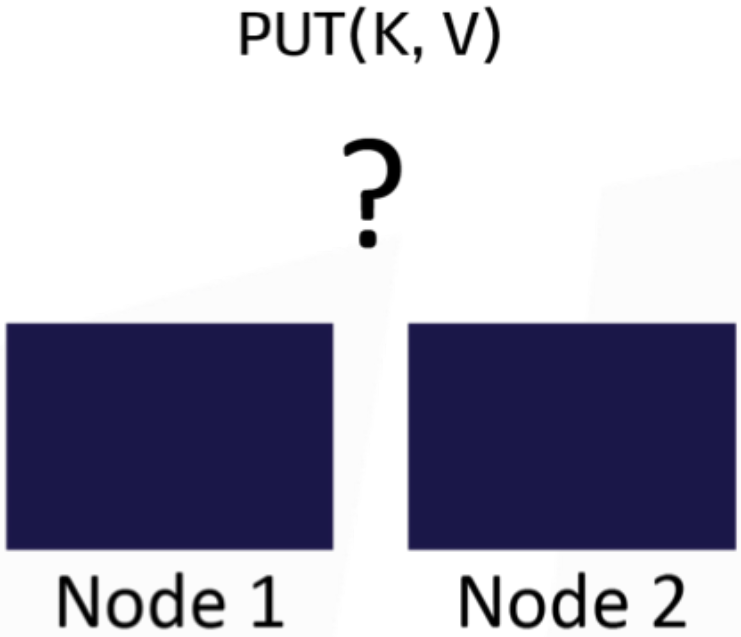
GridGain

Agenda

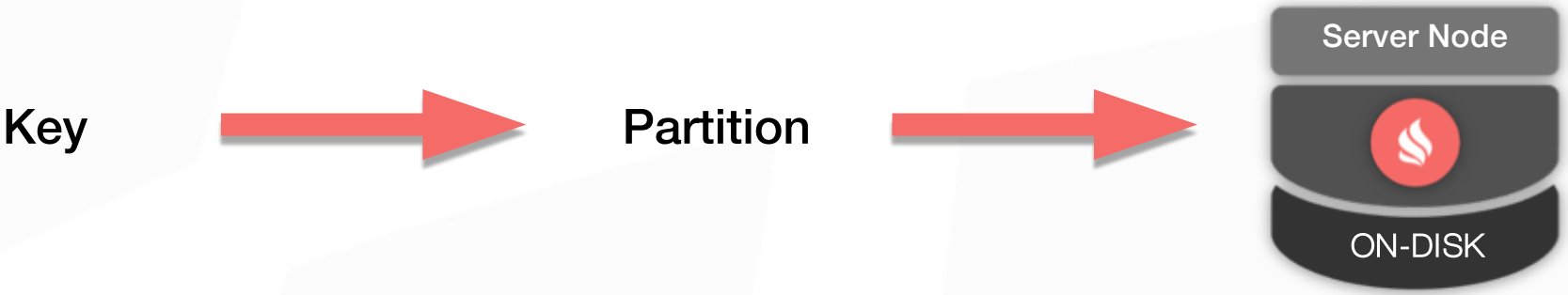
- Partitioning – Pitfalls of Even Distribution
- Affinity Collocation – Laid Out in Numbers
- Multi-threading of Distributed Systems – Things to Know

Partitioning – Pitfalls of Even Distribution

Where request goes?



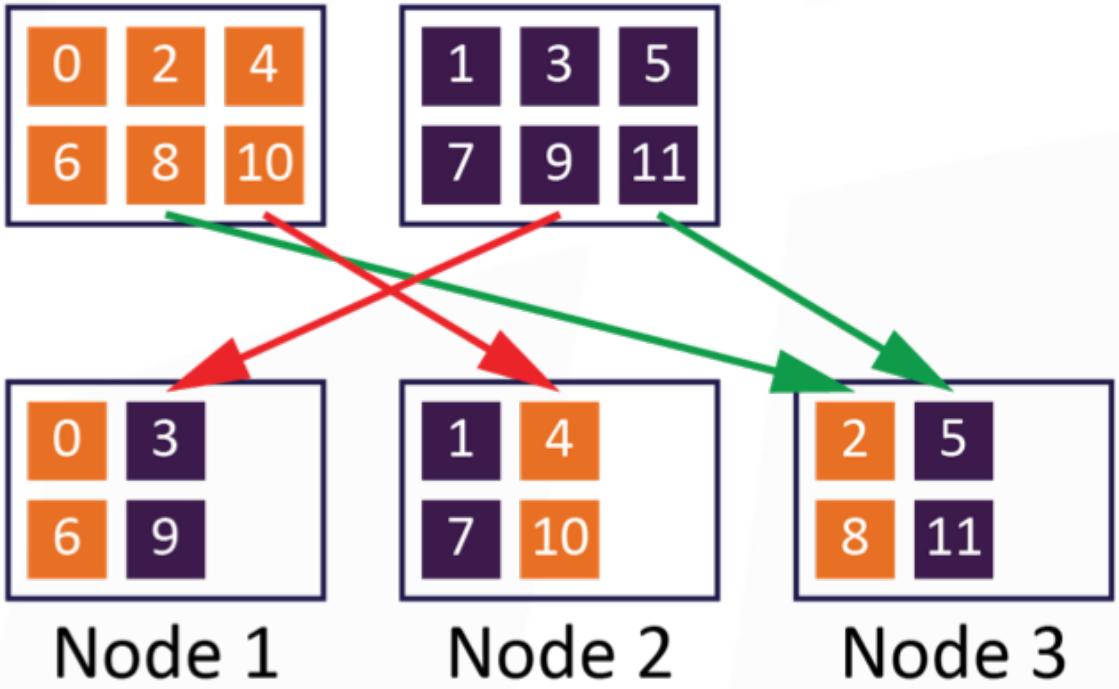
Affinity Function



Naïve Function: What's Wrong?

Node 1 Node 2

Naïve Affinity: Redundant Partitions Re-shuffling!

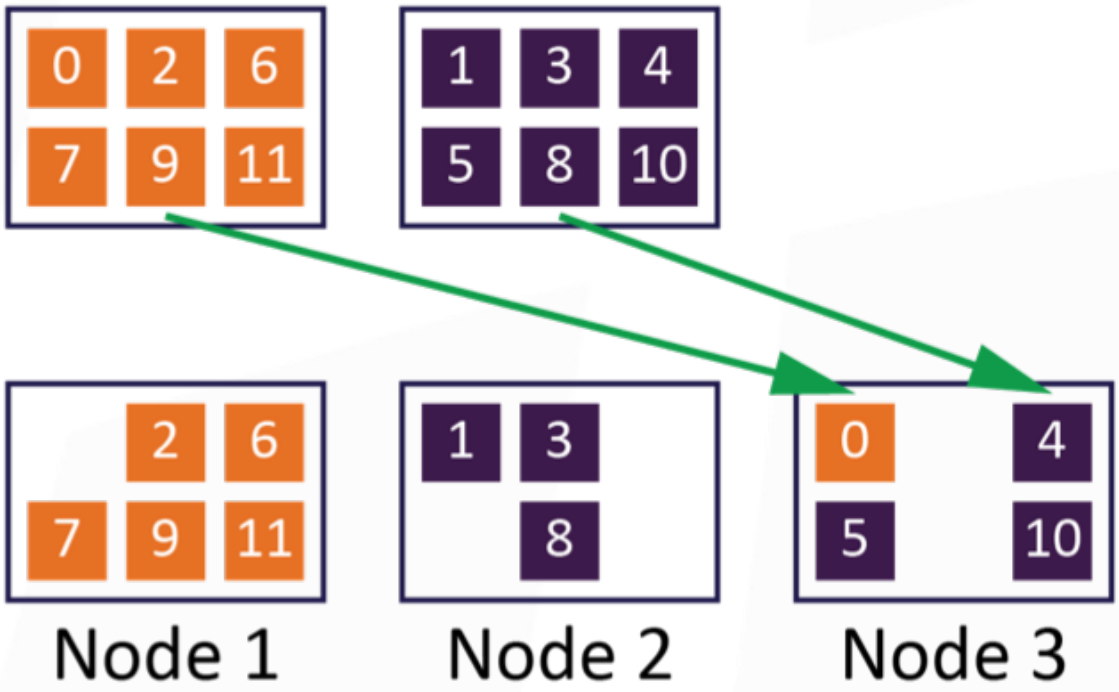


Productized Affinity Functions

- Consistent hashing [1]
- Rendezvous hashing (HRW) [2]

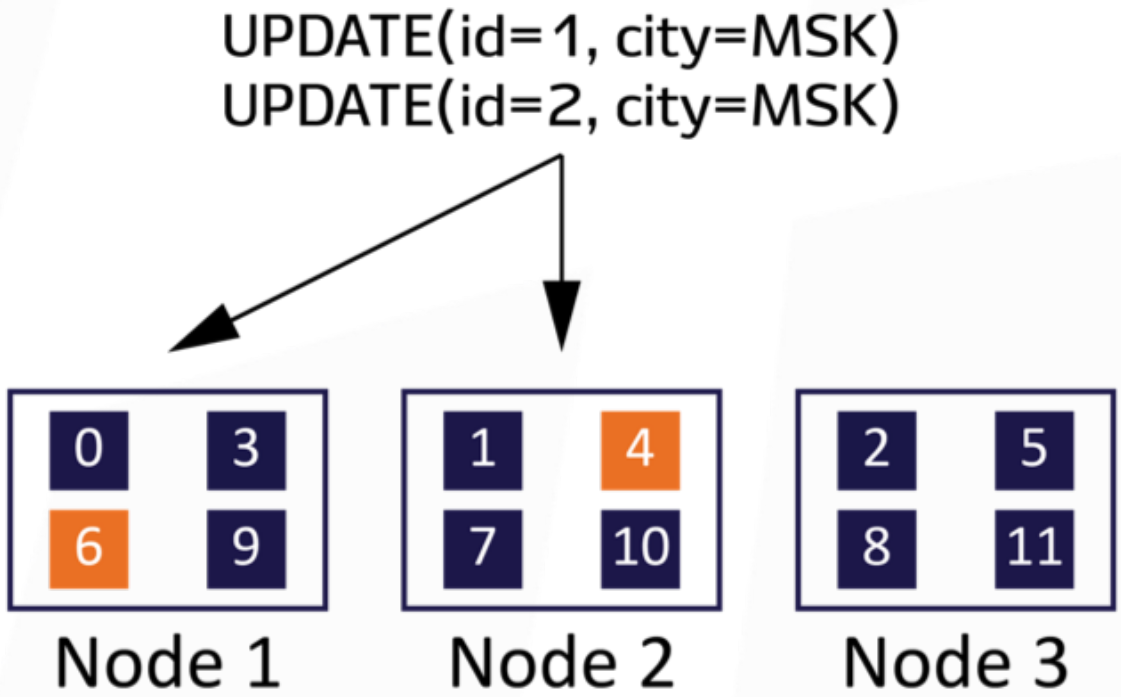
[1] https://en.wikipedia.org/wiki/Consistent_hashing
[2] https://en.wikipedia.org/wiki/Rendezvous_hashing

Rendezvous Affinity: Distribute Evenly at Best

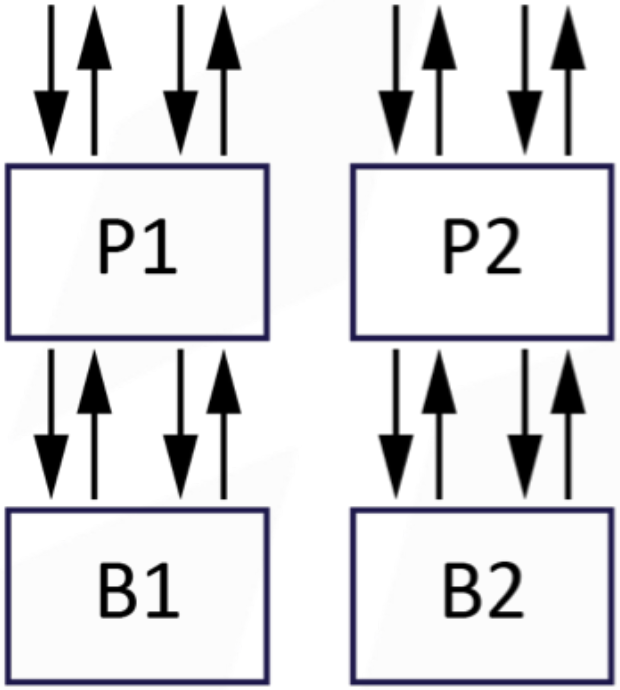


Affinity Collocation – Laid Out in Numbers

Running Transaction: No Collocation

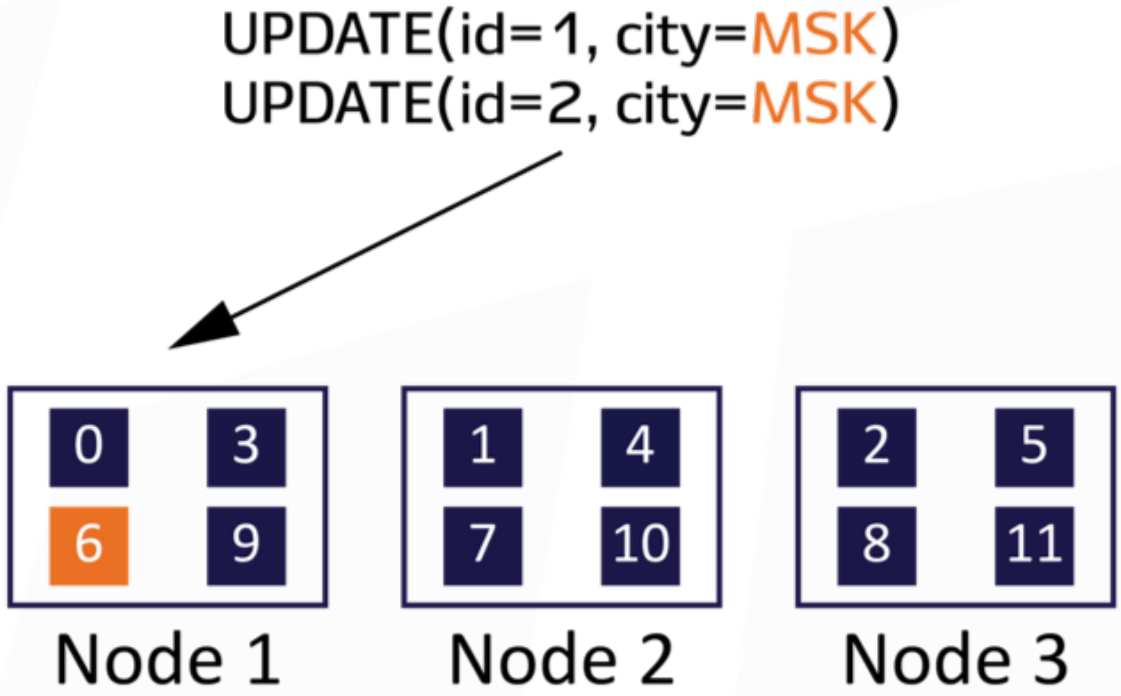


Non Collocated Data: Number of Operations

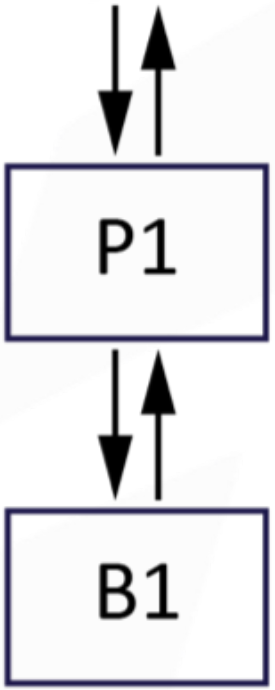



```
 2  (2 nodes)
 2  (primary + backup)
 2  (two-phase commit)
 2  (request-response)
--
16 network operations
```

Running Transaction: Collocated Data!



Collocated Data: Number of Operations




```
 1 (1 node)
 2 (primary + backup)
 1 (one-phase commit)
 2 (request-response)
--
 4 network operations
```

Upshot

Upshot

- Partitioning – it's not about even distribution only
- Affinity Collocation – a golden concept of distributed systems
- Business Data Model should be adopted/reconsidered
- Thread-per-partition – speeds up most but not all usage scenarios

Q/A