

Choosing the Right Grid Architecture



By : Zvika Markfeld, Consultant, Tikal
Knowledge

Classic Clustering

- ▶ Replicate State Between Nodes
 - » Http Session, Entity Beans, Hibernate + distributed 2nd level cache, ...
- ▶ Balance the load
 - » Client-side smart proxies
 - » Server side LB
- ▶ Master Election Mechanisms



Classic Clustering

- ▶ Pros:
 - » Does not require architectural changes to the application programming model
 - » Provides certain amount of scalability, fail-over
- ▶ Cons:
 - » Slow(topology changes, expensive election)
 - » No linear scalability when data is replicated
 - » Partial Replication? Dealing with Object Graphs?



Grid Types

- ▶ Data Grids
 - » Data is divided between nodes
 - » ...and re-assembled on demand

- ▶ Computational Grids
 - » Master/Worker
 - » Split long tasks into shorter sub-tasks
 - » Subtasks executed in parallel
 - » Results are collected, merged and returned
 - » Usually requires expressing the problem in Map/Reduce compliant way



Existing Specs

- ▶ The Jini/JavaSpace–Based Approach
 - » Network–based shared memory
 - » Initiated by Sun in 1998
 - » Not widely adopted, since then forsaken
- ▶ JCache(JSR107)
 - » Started in 2001
 - » Never completed
 - » Regarded as simplistic
 - » Implemented by some



Grid Implementations

- ▶ ***EHCache:***
Data grid, open-source, fast, small, simple, RMI-based
- ▶ ***Oracle Coherence:***
Data grid, fast, commercial, standalone / inside AS, replicated / partitioned, supports queries & indices
- ▶ ***JBossCache:***
Data grid, open-source, transactional, instrumented bytecode, fine-grained, JGroups-based communication

Grid Implementations

- ▶ ***Gigaspace:***
Commercial/open-source, supports JavaSpaces, map, queries, jdbc, jms, ...
- ▶ ***Open Terracotta:***
Data grid, Strong JVM integration, pure virtualization, transparent "wired" memory, GC, synchronization
- ▶ ***GridGain:***
Map/Reduce implementation, open-source, linear, extensive SPI support

Key Considerations

- ▶ Cache, Data Grid, Compute Grid OR Clustered VM?
- ▶ Open source OR Commercial?
- ▶ API driven OR transparent?
- ▶ Container OR standalone jar?





Q & A



Thank You

zvika@tikalk.com