



Apache Ignite 2.8: Improved Production Maintenance With Machine Learning

Denis Magda

Head of DevRel at GridGain

Apache Ignite PMC member



Topics

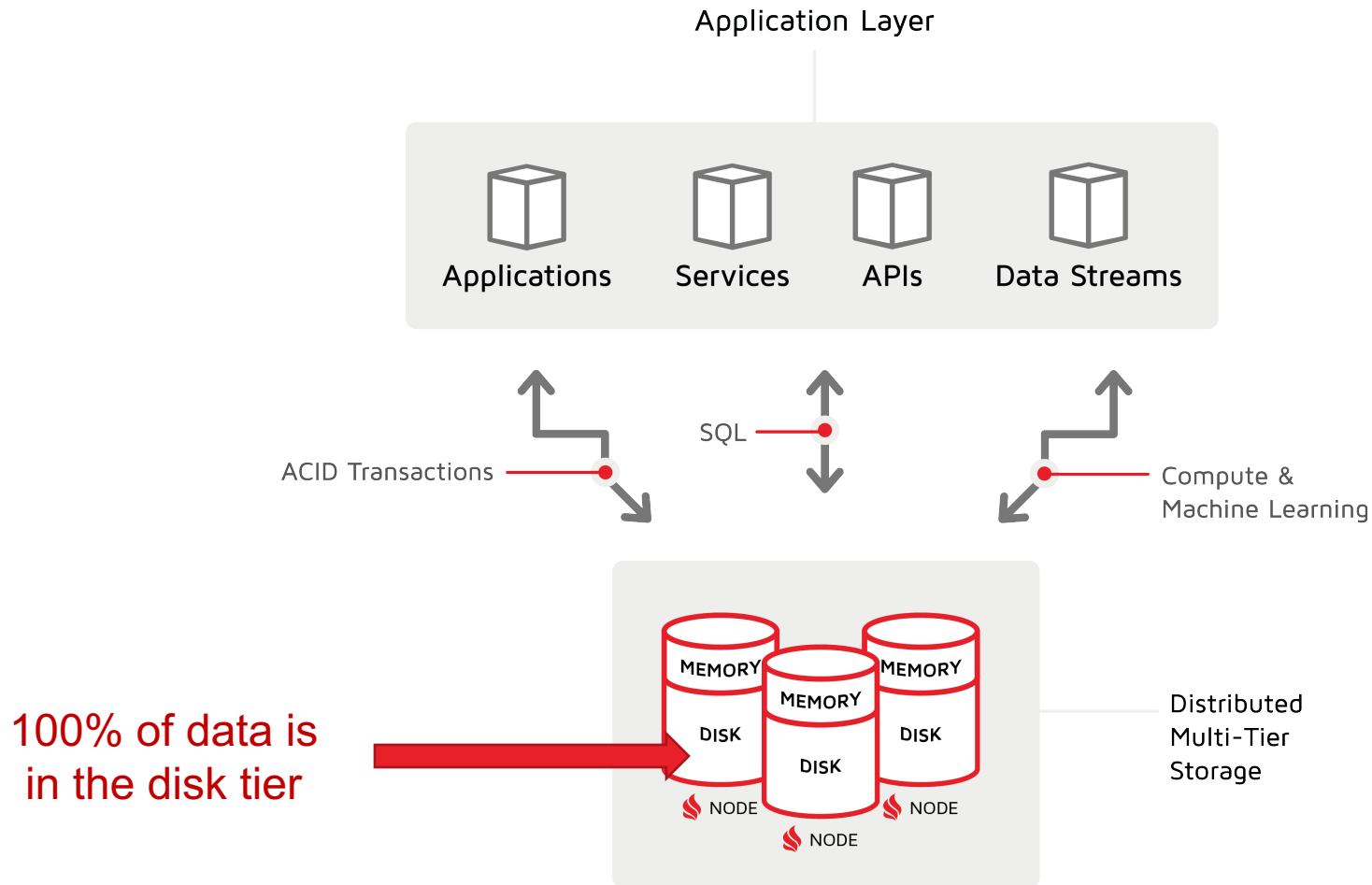


- Sustainable productions under high load
- Next-gen system for monitoring and tracing/profiling
- Partition-awareness in thin clients
- Machine Learning major upgrade

Sustainable Productions Under High Load



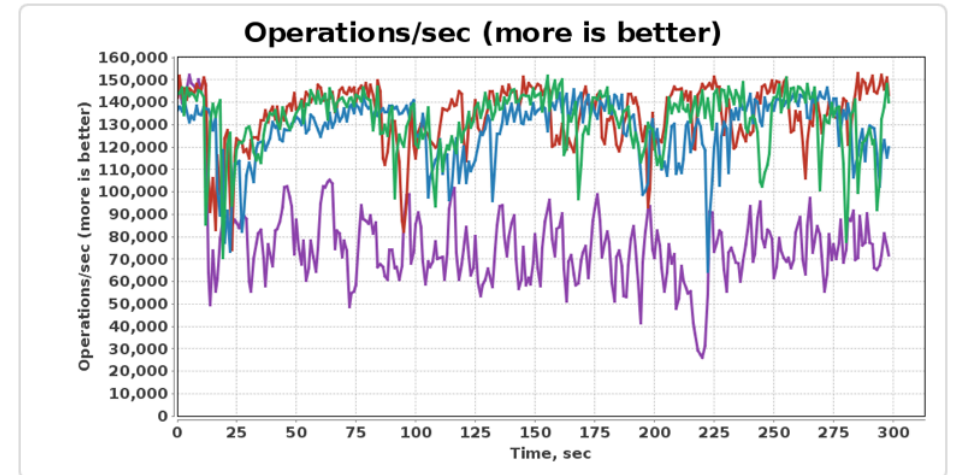
Ignite as a Database With Multi-Tier Storage



Data Pages and WAL Compaction



- On-disk data compression
 - Memory pages are uncompressed
- 2x-4x disk space usage reduction
 - LZ4, Snappy, ZSTD
- Less I/O => increased throughput and better/comparable latencies
 - Note: Use-case specific



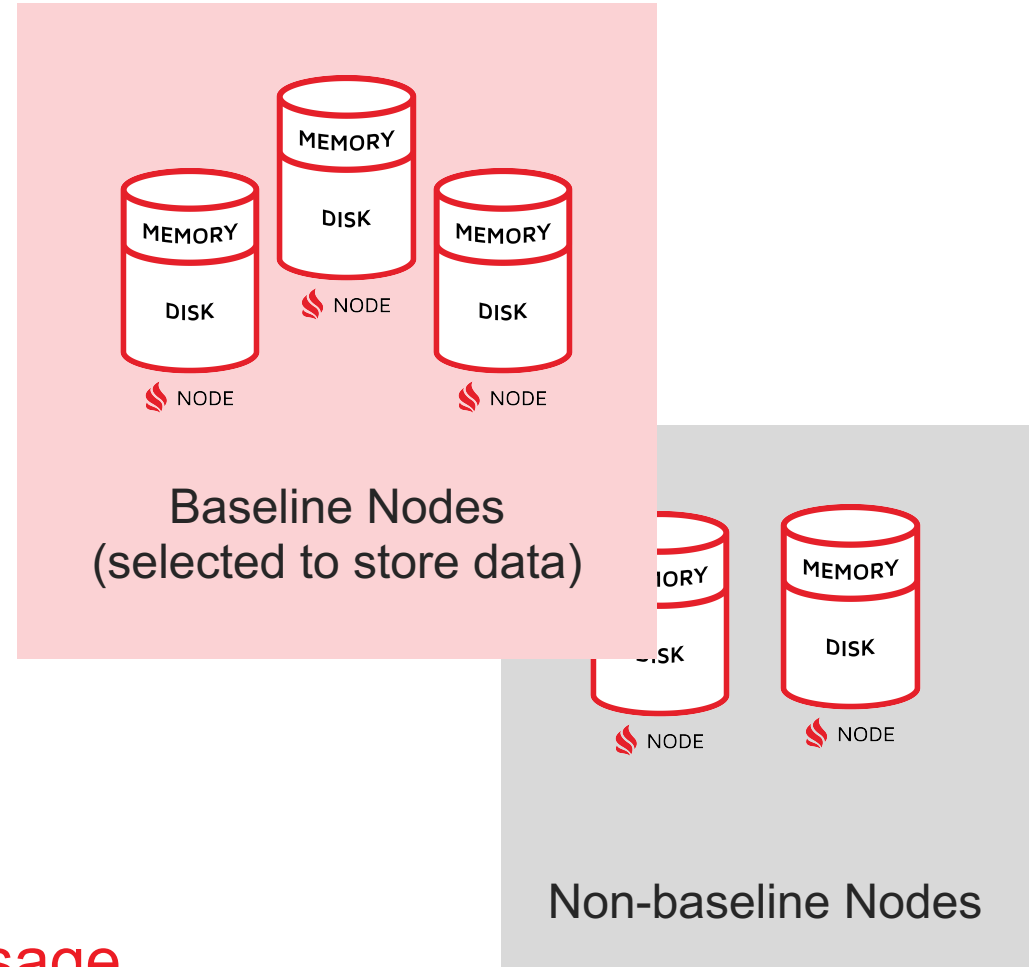
| | Avg | Min | Max | SD |
|---|------------|-----------|------------|-----------|
| ■ | 132,092.94 | 70,507.00 | 151,576.00 | 13,287.90 |
| ■ | 127,903.10 | 64,578.00 | 146,381.00 | 13,671.20 |
| ■ | 134,352.18 | 74,033.00 | 152,806.00 | 13,465.08 |
| ■ | 77,094.36 | 26,161.00 | 152,156.00 | 20,129.02 |

[Disk Compression Documentation](#)

Baseline Topology Auto-Adjustment

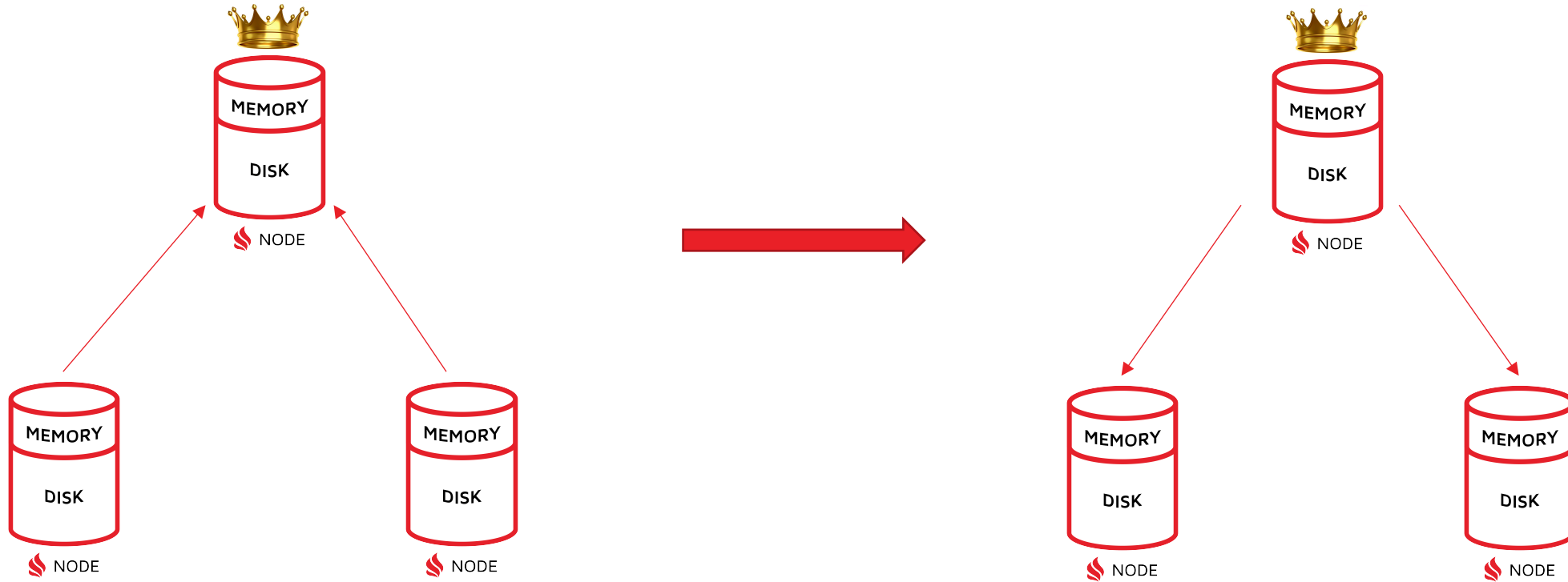


- Primarily used for deployments with Ignite persistence
- Auto-adjust changes and triggers rebalancing
 - Configurable timeout
- Automates cluster maintenance routines



Auto-Adjustment Usage

Partition Map Exchange (PME) Process



Each node shares the current state of its partitions with the coordinator (oldest node)

Coordinator merges the shared states and responds with consistent/reconciled partitions map

PME Triggers and Changes in Ignite 2.8



- PME triggers:
 - Topology change events (addition of new nodes, exit of nodes)
 - New caches creation, baseline topology changes
 - More in [the PME Under the Hood Guide](#)
- PME has its "stop-the-world" phases to preserve cluster-wide consistency:
 - Unnoticed for a majority of use cases
 - Spikes are detected in ultra low-latency scenarios
- "Stop-the-world" pauses are eliminated in **Ignite 2.8** when:
 - A baseline topology node leaves the cluster
 - Thick client connect to the cluster

Automatic Handling of Failures and Inconsistencies



- Read-repair – inconsistencies reconciliation between primary and backup copies (<https://apacheignite.readme.io/docs/read-repair>)
- Improved cluster recovery from write-ahead logs
 - [IGNITE-7196](#) and [IGNITE-9420](#)

Next-gen system for monitoring and tracing/profiling



Rationale for the New Monitoring Foundation



- *Fragmented and non-pluggable APIs* exposed via a *fixed* set of protocols
- *What consumes cluster resources?*
 - No way to monitor the state of internal structures and processes holistically (compute tasks, transactions, scan queries, etc.)
- *What's exactly running slow?*
 - No foundation for cluster and applications profiling

Ignite 2.8: New Monitoring & Tracing System



ZABBIX



Ignite 2.8

Metrics Exporters and System Views
(JMX, OpenCensus, SQL, Logs, etc.)

Metrics Registries

Tracing Exporters
(OpenCensus, OpenTracing)

Tracing Subsystem

Coming...

Cache

Transactions

SQL

...

Discovery

Ignite APIs

CPU

Network

RAM & Disk

OS & JVM

Resources

How to Start With the New System Today

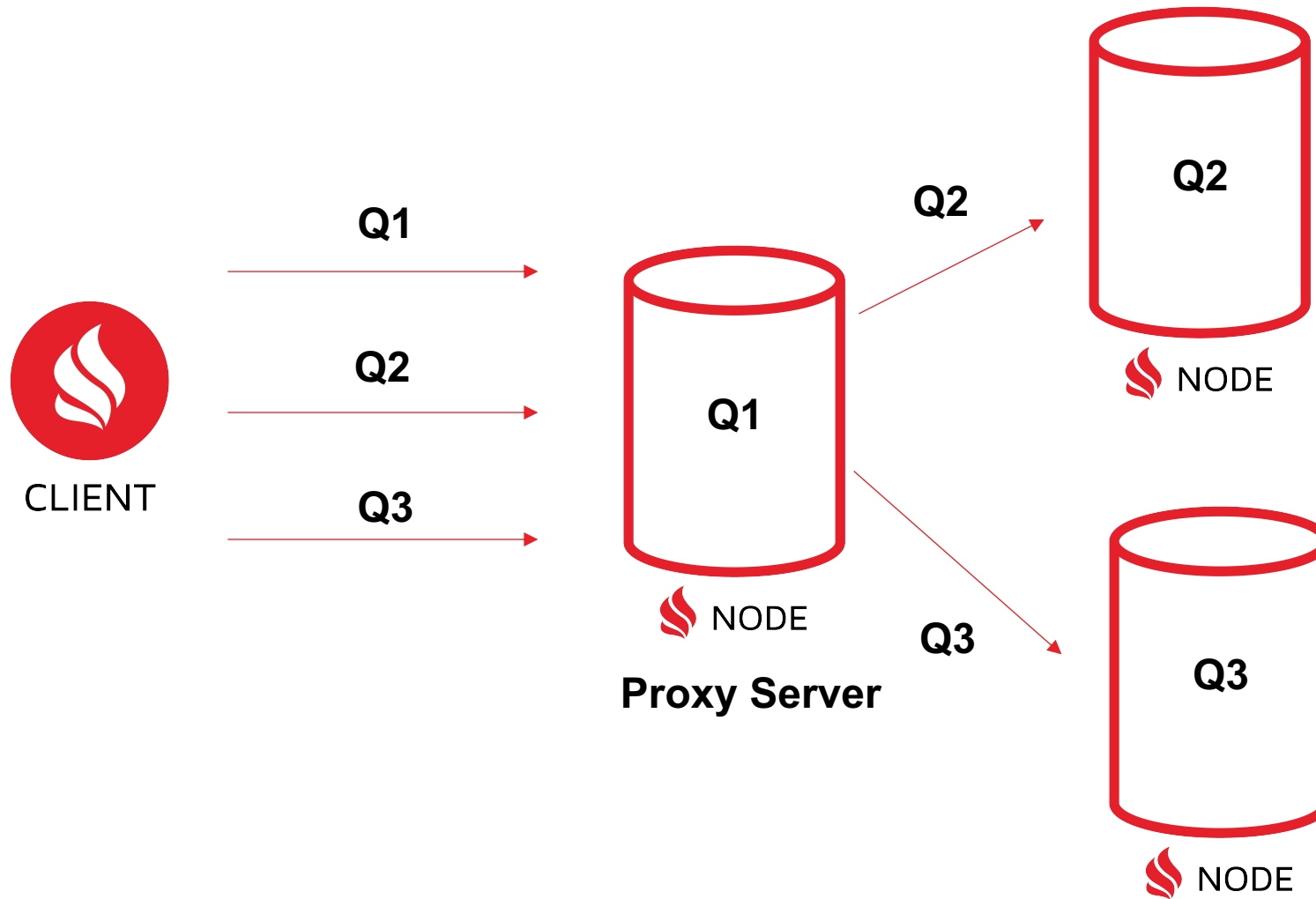


- Metrics Exporters and System Views usage:
 - <https://apacheignite.readme.io/docs/new-metrics>
 - <https://apacheignite.readme.io/docs/system-views>
- JMX example:
 - <https://github.com/gridgain/demos/tree/master/ignite-metrics-demo>
- OpenCensus example:
 - <https://github.com/apache/ignite/tree/master/examples/src/main/java/org/apache/ignite/examples/opencensus>
- IEP-35: Monitoring and Profiling
 - <https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=112820392>

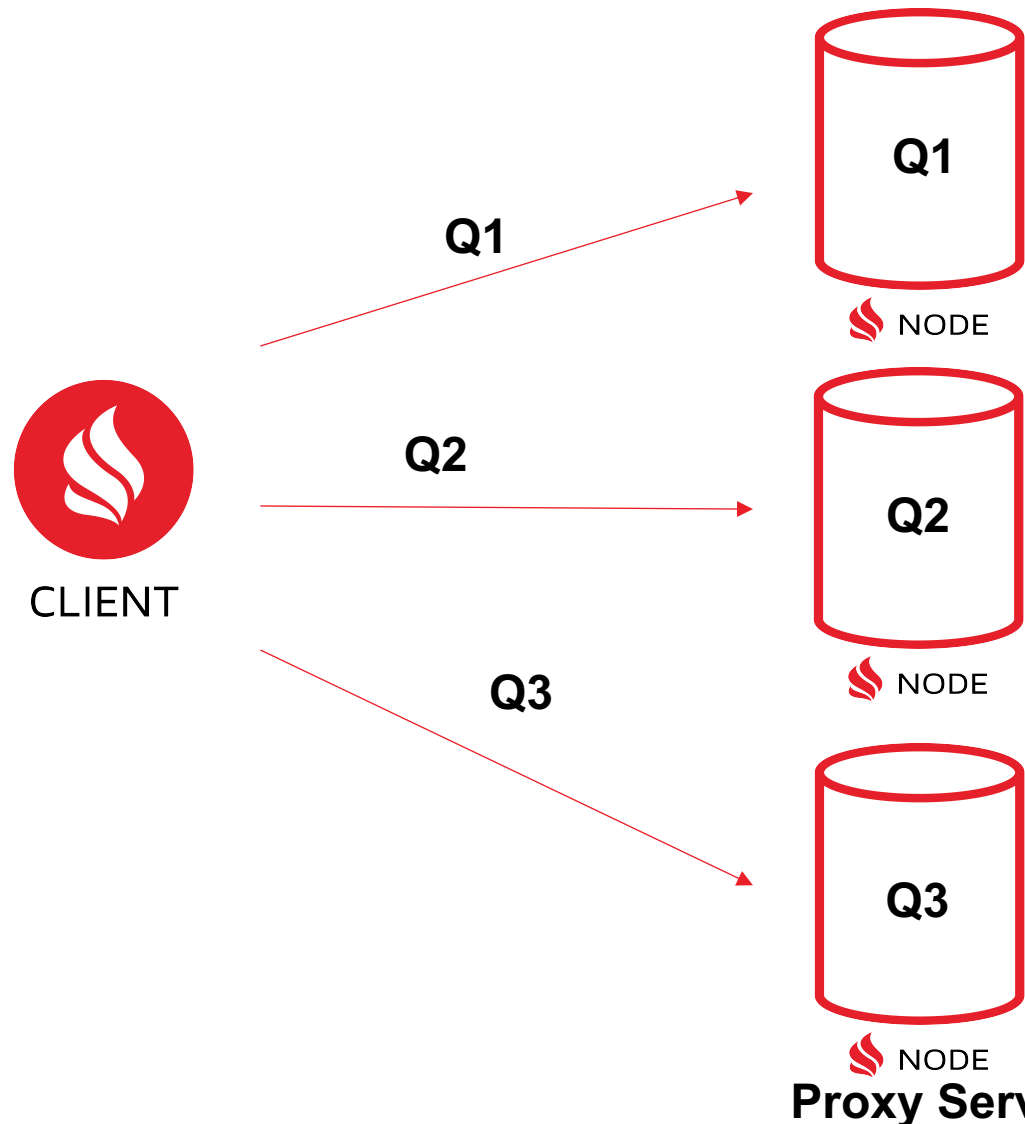
Partition-Awareness in Thin Clients



Thin Clients Interaction With Servers Without Partition-Awareness



Thin Clients Interaction With Servers With Partition-Awareness Enabled



Available for Java, C#, C++ and Python
Starting Ignite 2.8

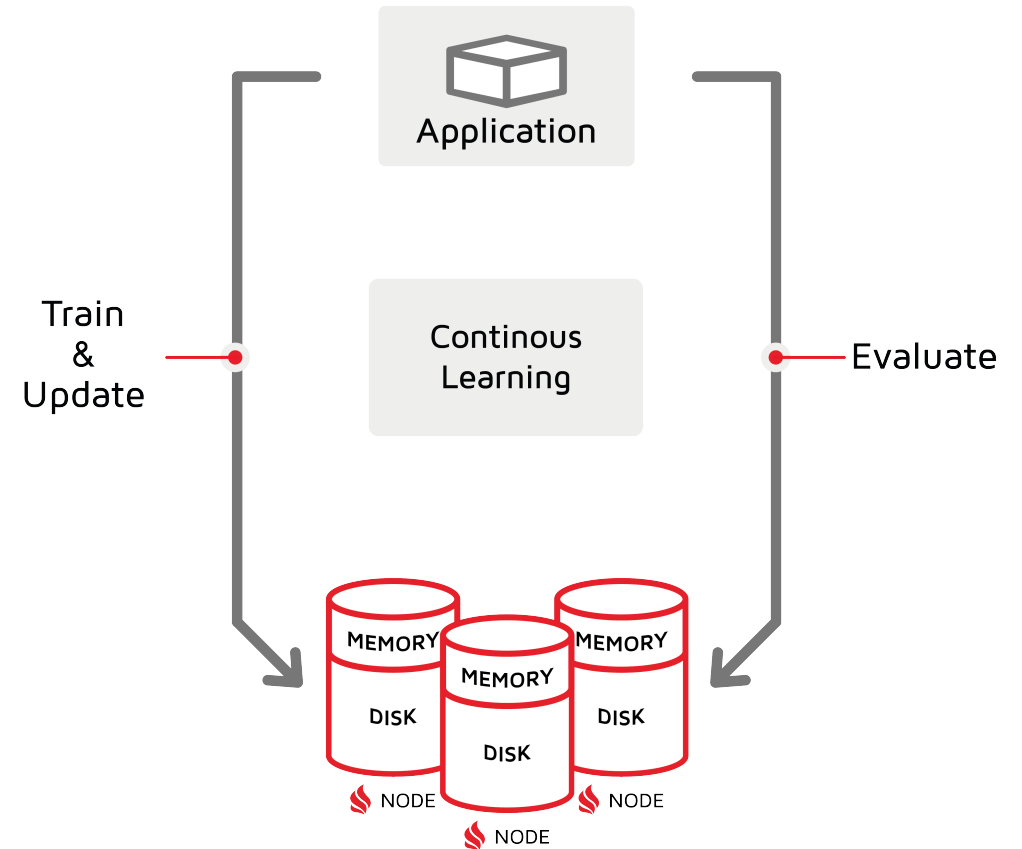
Major Upgrade of Ignite Machine Learning



Continuous Learning With Real-time and Batch Data



- New “model update” APIs let refine an existing model on the fly with new data samples
- Ignite ML is deeply integrated into Ignite multi-tiered storage:
 - Optional ETL and no data shuffling during training
 - (Re-)Train across petabytes of in-memory and on-disk data



Advanced Toolkit for Machine Learning Experts

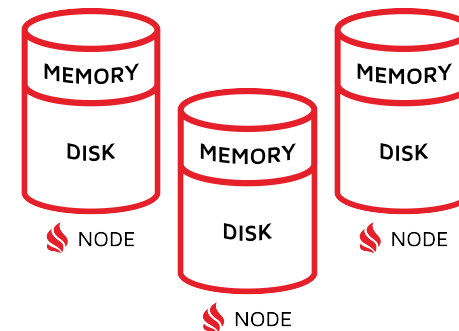


- Pipelining APIs
 - <https://apacheignite.readme.io/docs/pipeline-api>
- Evaluators
 - <https://apacheignite.readme.io/docs/evaluator>
- Model cross-validation
 - <https://apacheignite.readme.io/docs/cross-validation>
- Models ensembling
 - <https://apacheignite.readme.io/docs/ensemble-methods>

Model Importing from Spark and XGBoost



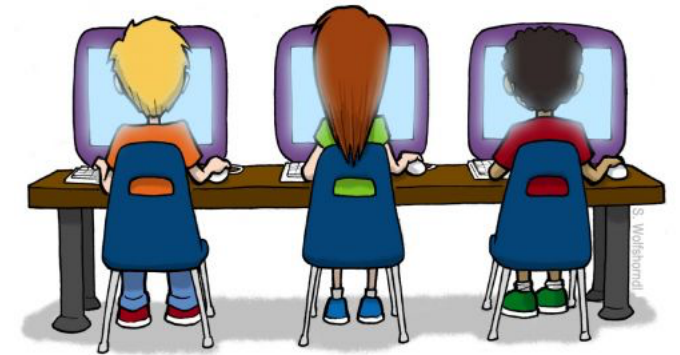
- Train (*if convenient*) in Spark or XGBoost and run at scale with Ignite
 - <https://apacheignite.readme.io/docs/model-import-from-apache-spark>
 - <https://apacheignite.readme.io/docs/model-import-from-gxboost>



Join Apache Ignite Community!



- Rapidly Growing Engineering Community
- Great Way to Learn Distributed Systems, Computing, SQL, ML, Transactions
- How To Contribute:
 - <https://ignite.apache.org/community/contribute.html>
- Join Ignite Meetups:
 - <https://ignite.apache.org/meetup-groups.html>



Q&A

