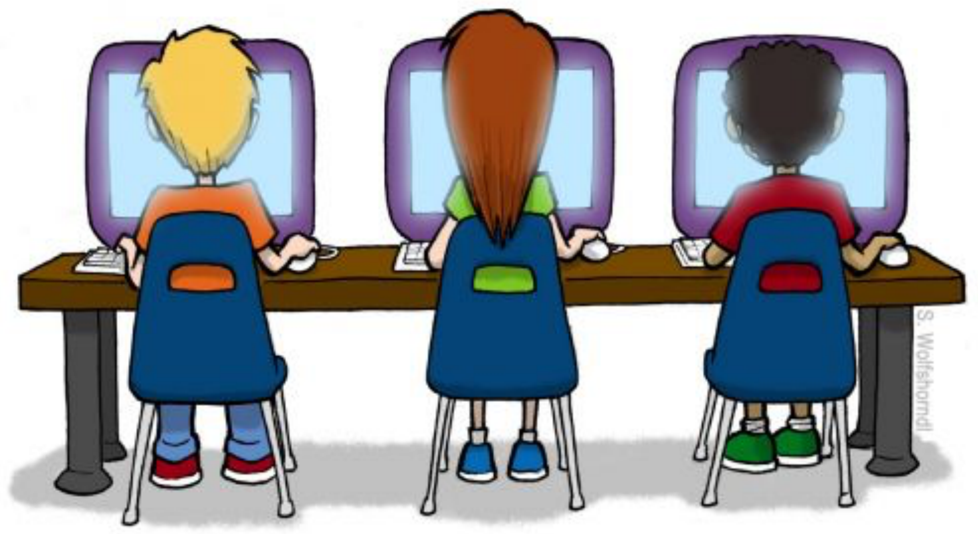
How to Choose the Best In Memory Solution for Your Apps

Agenda

- IMC Introduction
- IMC Myths
- IMC Product Categories
- Q & A

Apache Ignite: We Are Hiring!

- Very Active Community
- Great Way to Learn Distributed Computing
- How To Contribute:
  - https://ignite.apache.org/community/contribute.html#contribute
  - https://cwiki.apache.org/confluence/display/IGNITE/How+to+Contribute



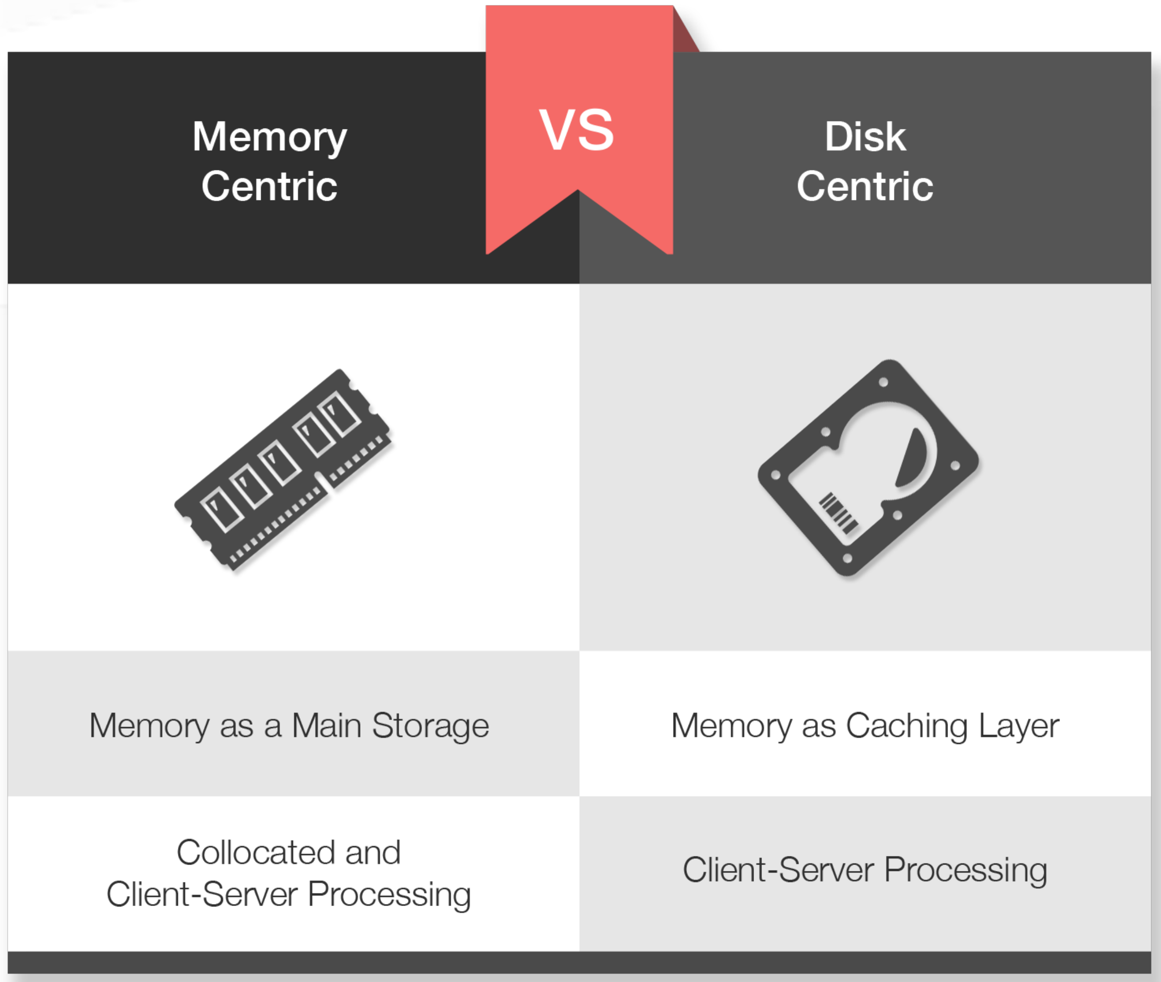
In-Memory Computing

uses high-performance, distributed memory systems
to compute and transact on large-scale data sets
in real-time, orders of magnitude faster
than disk-based systems.

Memory Centric vs. Disk Centric

- **Disk First Architecture**
  - Disk as primary storage, memory for caching
  - Client-Server processing
  - Latency: milliseconds

- **Memory First Architecture**
  - Memory as primary storage, disc for backup
  - Collocated processing
  - Latency: nanoseconds to microseconds



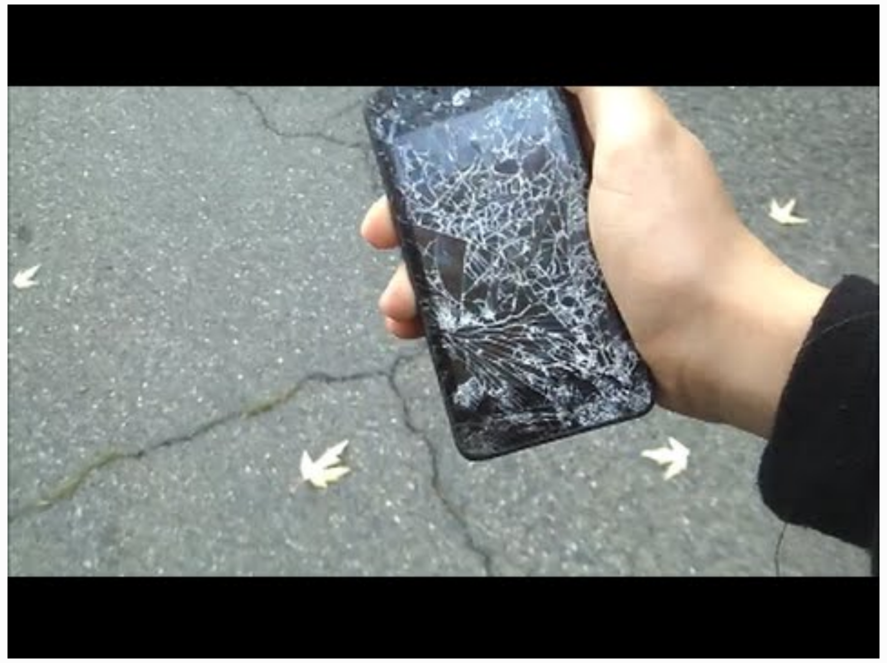
Memory Centric	VS	Disk Centric
Memory as a Main Storage		Memory as Caching Layer
Collocated and Client-Server Processing		Client-Server Processing

Myth #1: Too Expensive

- **Facts:**
  - Memory price declined over last 30 years
    - went slightly up in past 2 years, but insignificant
  - Memory can be used as a caching layer
    - disk is a super set of memory
  - Memory can be extended to disk with swap store
    - disk only for cold data

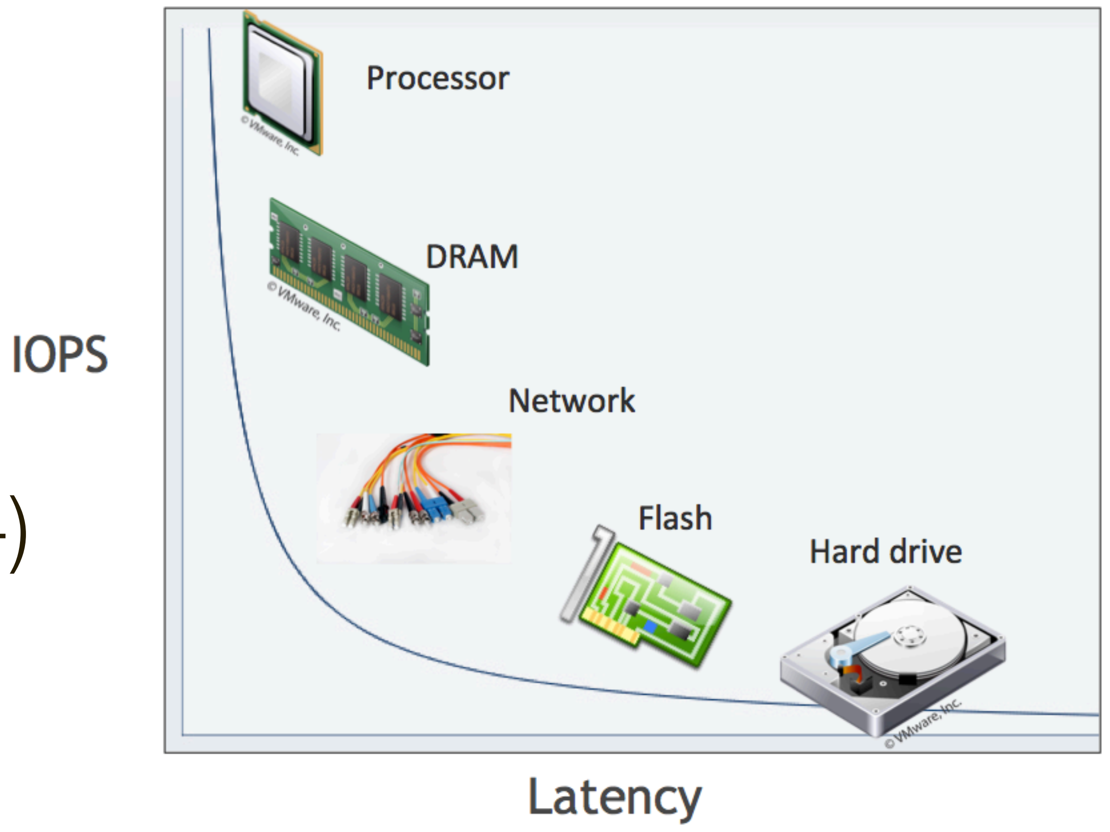
Myth #2: Not Durable

- **Facts:**
  - IMC systems have durable backups and disk storage
    - active or passive replicas,
    - transactional read-through and write-through
  - Mature IMC systems provide tiered storage
    - disk to store superset of data or cold data
    - memory to store hot data
  - Operational vs. Historical datasets
    - 99% of operational datasets < 10TB

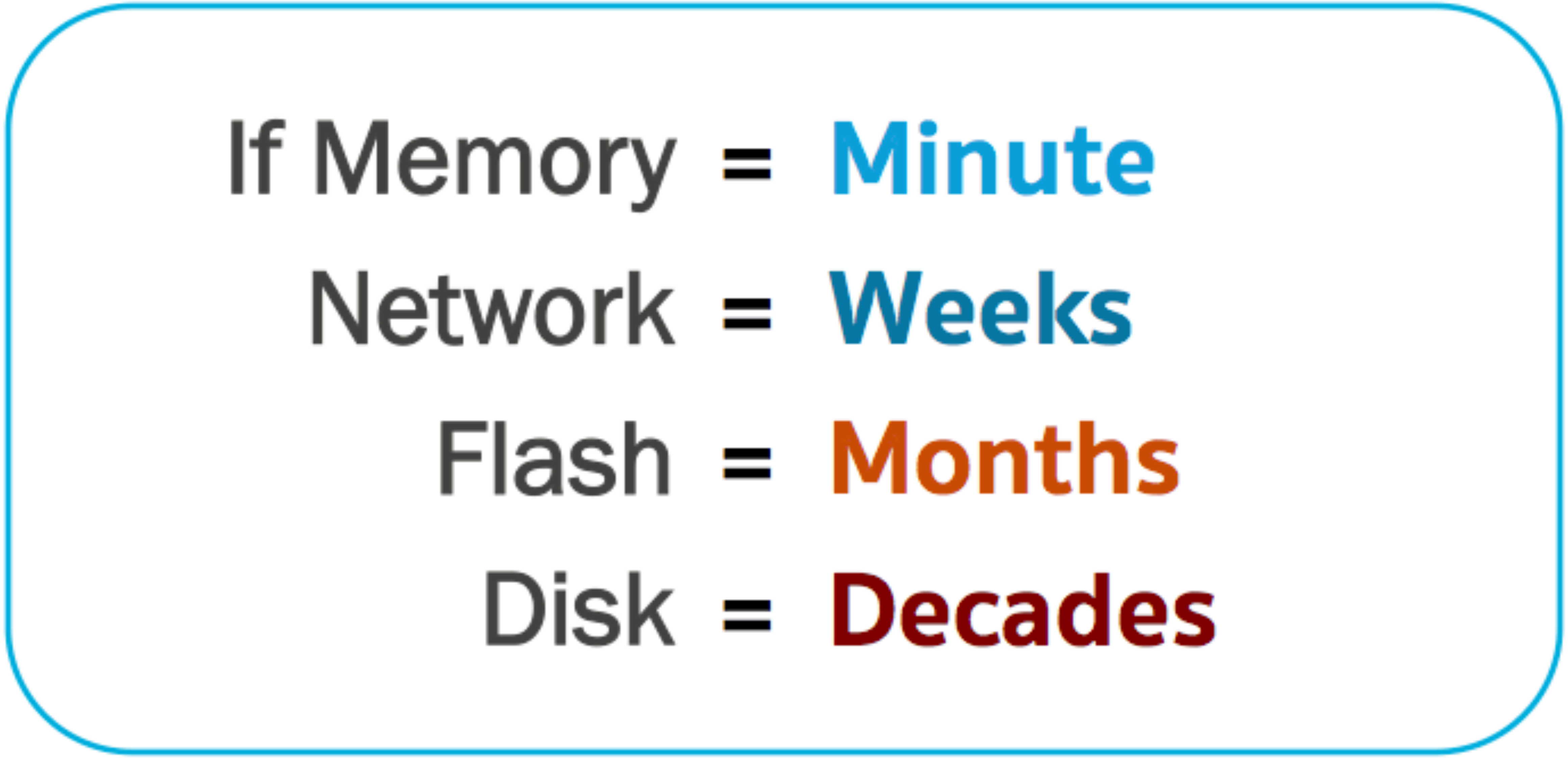


Myth #3: Flash Is Fast Enough

- **Facts:**
  - Flash on PCI-E is still... a block device.
    - Still going through OS I/O, I/O controller, etc.
  - DRAM - nanoseconds
  - 10GbE - microseconds (~50)
  - Flash or SSD - microseconds (between 20-500+)
  - Spinning Disk - milliseconds (between 4-7)



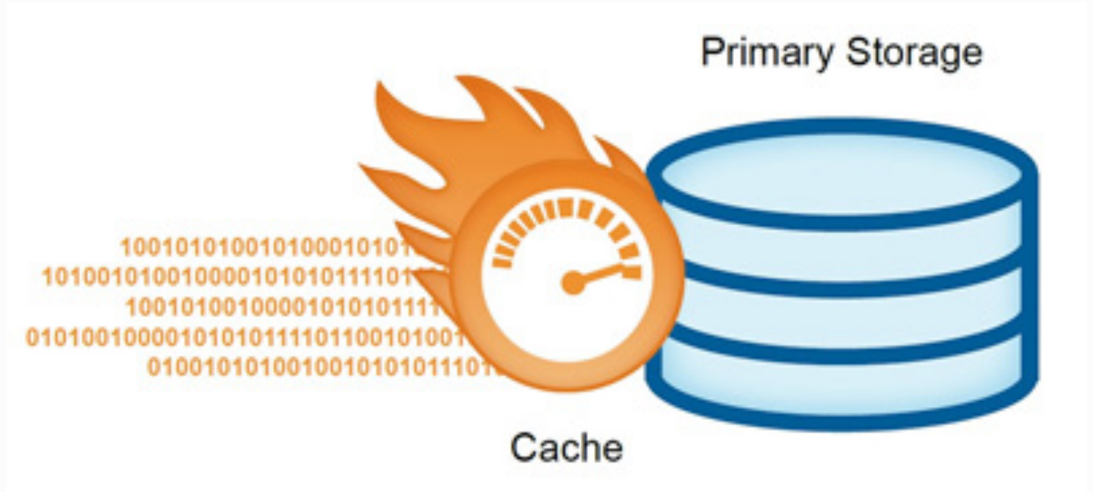
Mapping nanoseconds to our universe



If Memory	=	Minute
Network	=	Weeks
Flash	=	Months
Disk	=	Decades

Myth #4: Only For Caching

- **Facts:**
  - Caching is important use case, but limited
    - Easiest adoption and a “low-hanging fruit”
  - In-Memory Data Grids & Databases for today
    - Main system of records are in-memory
  - Memory-Centric Systems for tomorrow
    - Memory and disk are tightly integrated
    - Can store more data than fits in memory



IMC Product Categories

- **In-Memory "Options"**
  – Oracle Database 12c, Microsoft SQL Server, Cassandra
- **In-Memory Caches**
  - Redis, Memcached
- **In-Memory RBDMS**
  – VoltDB, MemSQL, Apache Ignite (use case)
- **In-Memory Data Grids**
  – Hazelcast, Coherence, Geode, Apache Ignite (use case)
- **Memory-Centric Platforms**
  – Apache Ignite, GridGain

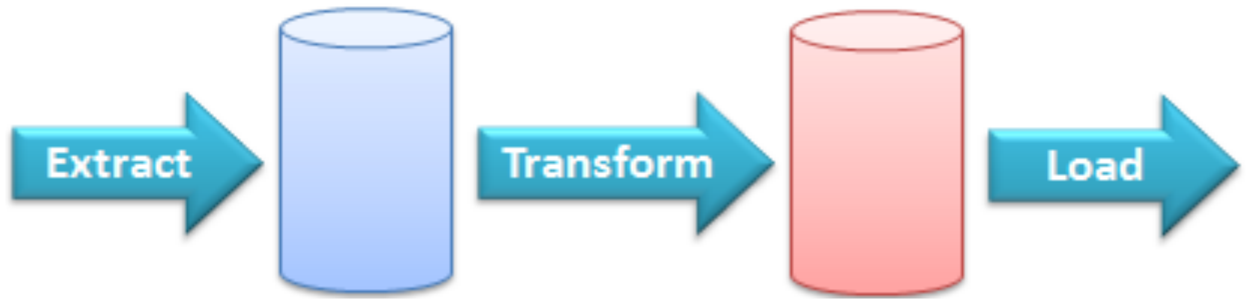
Category: In-Memory “Options”

- Feature onto an **<u>EXISTING</u>** database
- Ideal when only configuration change is possible:
  - No API changes
  - No code changes
  - No data migration
- Limited benefits
  - “marketing” for basic caching
  - not distributed
  - not horizontally scalable

Fast Data & Big Data

- Fast Data
  - OLTP mostly
  - Smaller Operational Data Set
  - High Throughput (ops/sec)
  - Low Latencies
  - Consistent or Transactional

- Big Data
  - OLAP mostly
  - Larger Historical Data Set
  - Read-Mostly
  - Throughput Not Important
  - Low Query Latencies
  - Good-enough for interactive analytics



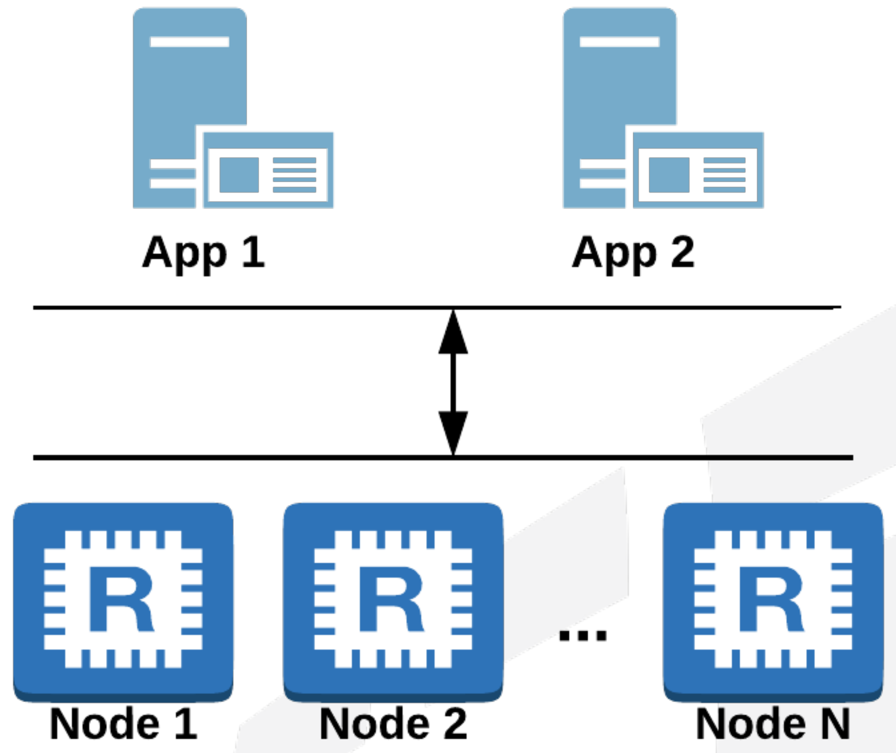
Fast Data & Big Data

- Fast Data
  - Streaming
    - Apache Flink
    - Apache Kafka
    - Apache Apex
  - In-Memory Data Grids
    - Apache Ignite
    - Apache Geode
  - In-Memory Database
    - VoltDB
    - MemSQL
  - NoSQL
    - MongoDB
    - Apache Cassandra

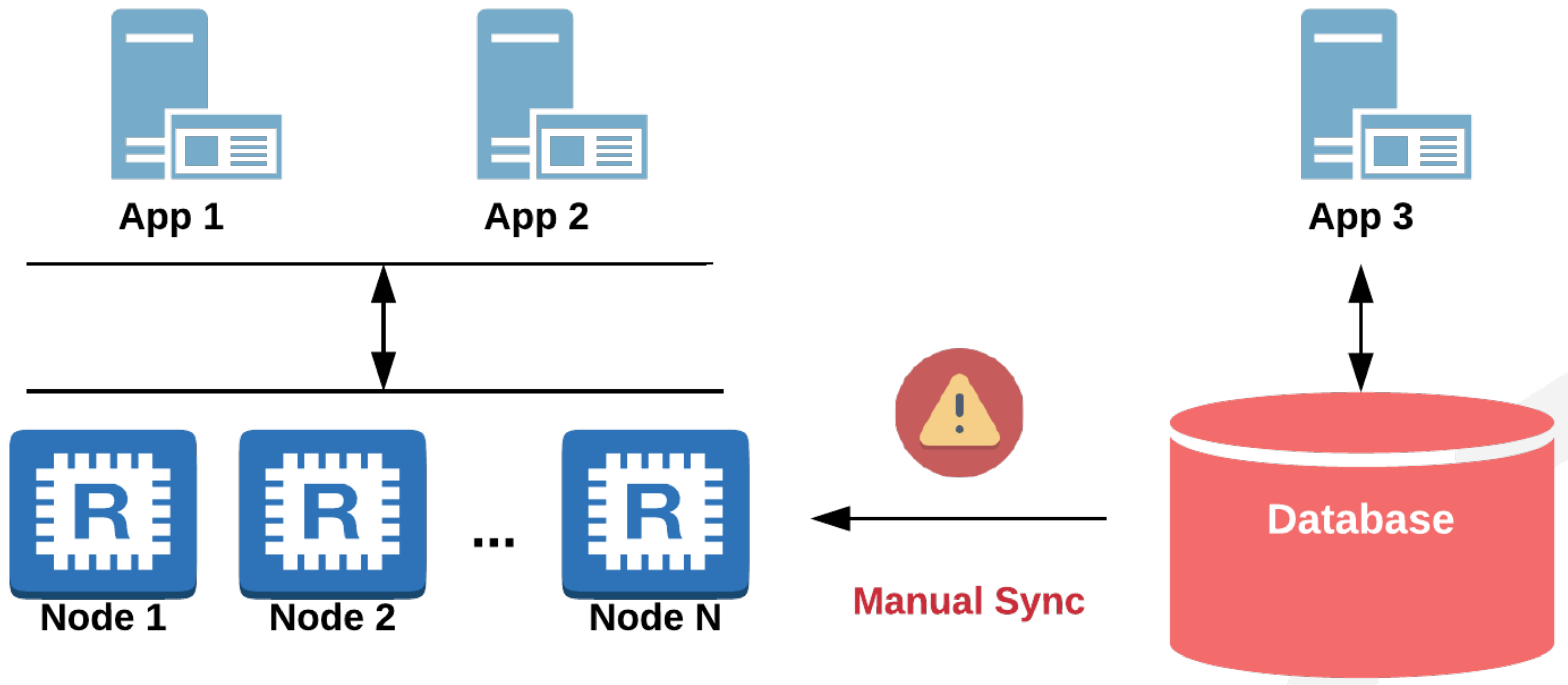
- Big Data
  - Apache Hadoop
    - MapReduce
    - HDFS
    - HBase
  - Apache Spark
    - Machine Learning
    - Graph Processing
    - SQL
  - Warehouse/DB Vendors

Category: **In-Memory Caches**

- Distributed In-Memory cache
  - Redis
  - Memcached
- Main Features
  - Shared Cache
  - Beyond Local RAM Capacity
  - Easy of maintenance



Where Distributed Caches Fail?



Category: In-Memory Databases

- In-Memory Databases
  - MemSQL
  - VoltDB
- Main Features
  - High-Throughput
  - Low Latencies
  - Full SQL Support
    - However, SQL is the only API
  - Disk Persistence
    - Disk is just a copy of memory

- **Complete replacement of existing databases! Good or Bad?**

Category: In-Memory Data Grids

- In-Memory Data Grids
  - Apache Geode, Hazelcast, Oracle Coherence
  - Apache Ignite (use case)
- Main Features
  - High Throughput & Low Latencies
  - Transactions
  - Collocated Processing
  - Data Querying Capability
  - Disk Persistence
    - Read & Write-through to databases
    - Keep your existing database

Apache Ignite – Memory Centric Platform

Financial Services

Telco

Travel & Logistics

E-Commerce

Pharma & Healthcare

IoT

SQL | Key/Value | Transactions | Compute | Services | Streaming | ML

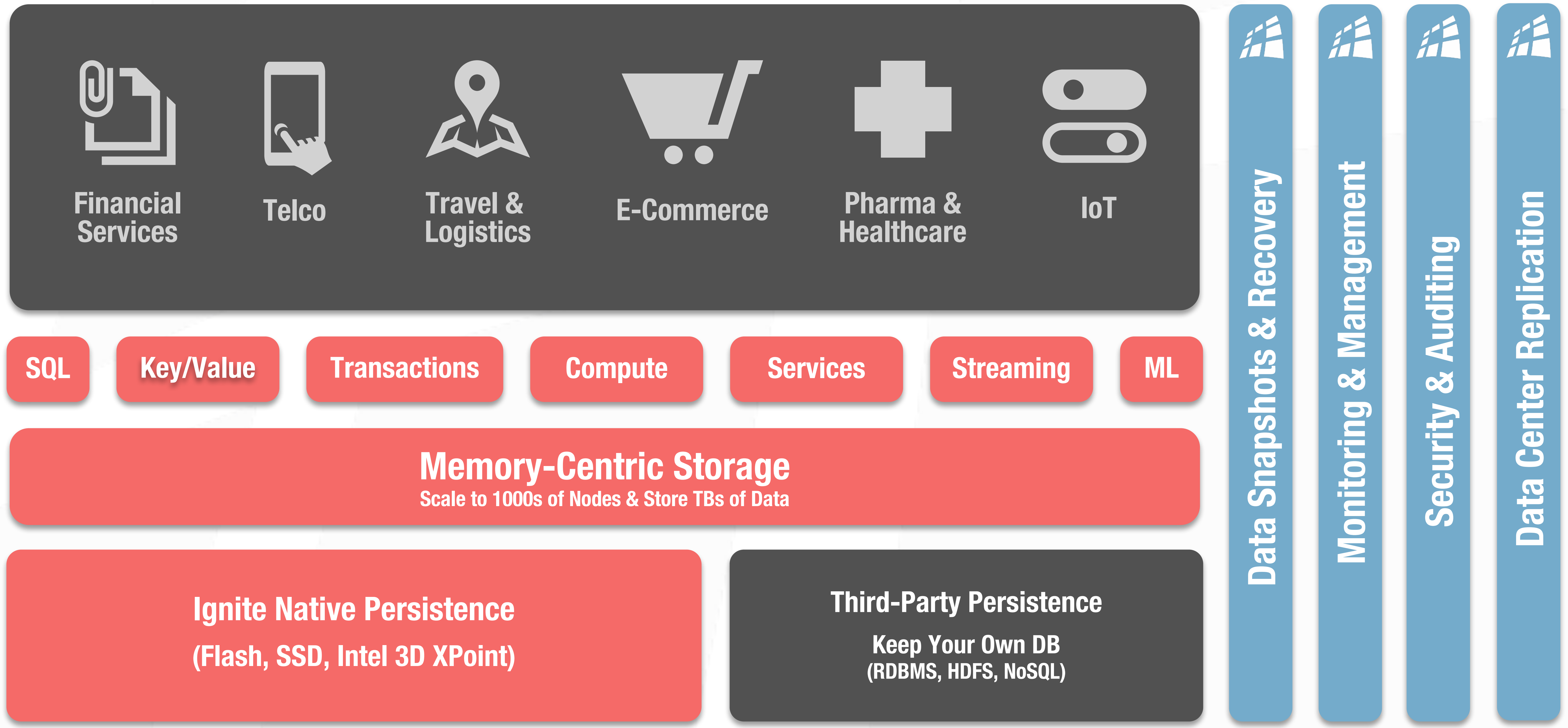
Memory-Centric Storage
Scale to 1000s of Nodes & Store TBs of Data

Ignite Native Persistence
(Flash, SSD, Intel 3D XPoint)

Third-Party Persistence
Keep Your Own DB
(RDBMS, HDFS, NoSQL)

Memory-centric distributed database, caching, and processing platform

GridGain In-Memory Computing Platform



How Ignite Compares

Feature	RDBMS	NoSQL	IMDG	**Ignite**
Scale Out	X	✓	✓	✓
Availability	X	✓	✓	✓
Consistency	✓	X	✓	✓
In-Memory	✓	X	✓	✓
Persistence	✓	✓	X	✓
SQL	✓	X	X	✓
Key-Value	X	✓	✓	✓
Collocated Processing	X	X	✓	✓

GridGain

Any Questions?

Follow the conversation.
http://www.gridgain.com

#apacheignite
#gridgain
#dmagda