Scalable Software. Real-time Big Data Analytics.

The Future of Real-time Big Data:

In-Memory Data Grids

Jacek Kruszelnicki, Numatica Corporation

E-mail: jacek@numatica.com
Phone: 781 756 8064

Presenter

Numatica
Scalable Software. Real-time Big Data Analytics.

Jacek Kruszelnicki
President & CEO, Numatica Corporation

"Intellectuals solve problems. Geniuses prevent them."
-- Albert Einstein

- 20+ years of shipping scalable, distributed enterprise software
- Vendor – neutral. Merit-based. Biased towards simplicity/elegance.
- Author, mentor & conference speaker
- Founder and past persident of New England WebLogic User Group
- M.S. in Computer Science, Adjunct Professor at Northeastern University

Agenda

Definitions:

- What is Big Data?
- What is Real-time Big Data Analytics?
- What is High-velocity data stream computing

Business Drivers

Big Data Universe: Hadoop, NoSQL and Distributed Data Grids

Distributed Data Grids:

- What are Distributed Data Grids
- Charateristics of a Distributed Data Grid

Major players, and the winner is....

From Technlogies to Platforms

The Ultimate: Unified Big Data processing

Not on Agenda

Numatica
Scalable Software. Real-time Big Data Analytics.

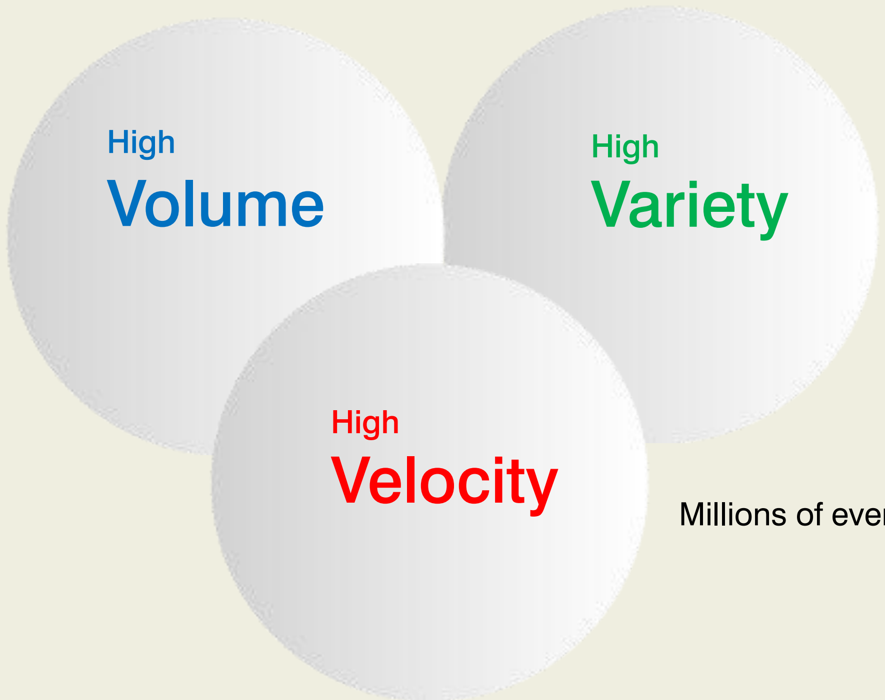
- Not an IMDG tutorial
- Not In-depth comparison of IMDGs (things change too fast)
- Data modeling, partitioning strategy (hub/spoke, data affinity, etc) (separate presentation)
- Analysis of competing frameworks suitable for Real-Time B.D.A. (separate presentation)
- CAP Theorem discussion ☺

What is Big Data?

Numatica
Scalable Software. Real-time Big Data Analytics.

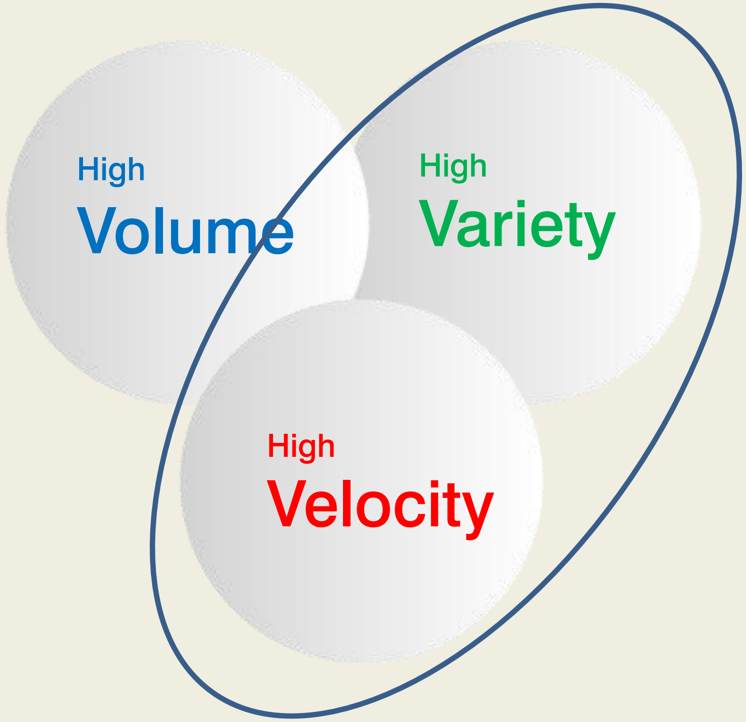
Data sets too large and complex to process using standard DB tools and data processing applications

Few dozen terabytes to
Many petabytes of data
in a single data set.



Structured, semi-structured
At-rest, In-flight
Variety of formats

Millions of events per second



Focus on Velocity + Variety

Rapidly changing data (at rest or in-motion):

- **Business activities**
  Examples: trades, orders
- **External stimuli**
  Examples: market data, current customer location
- **Operational monitoring**
  Example: application log entries

Business Drivers

Numatica
Scalable Software. Real-time Big Data Analytics.

Operational Intelligence:

- Analyze stream of business activities and external stimuli on-the fly
- React to them (preferably) instantaneously

- First-mover competitive advantage (new market conditions, customer wishes, etc)
- Real time data stream processing is critical, otherwise business value is lost.

Real-time Big Data Analytics examples:

- Dynamic pricing (e-commerce)
- High-frequency trading
- Network security threats
- Credit card fraud prevention
- Factory floor data collection, RFID
- Mobile infrastructure, machine to machine (M2M) applications
- Prescriptive or Location-based applications
- Real-time dashboards, alerts, and reports

Growth (per Gartner)

Numatica®
Scalable Software. Real-time Big Data Analytics.

- In-Memory Computing Racing Towards Mainstream Adoption
- Market to Reach $1 Billion by 2016
- Issues so far:
  - Lack of standards,
  - Scarcity of skills,
  - Relative architectural complexity,
  - Monitoring and management challenges

Real-time Data Processing

- Advanced queries or complex transactions on huge datasets
- Get results in (near) real-time

In-motion:

- Rapidly changing, massive stream(s) of data (events)
- Multiple sources, formats
- Needs to be processed in-flight
- Events persisted or not, depending on value

At rest:

- Data already persisted, change notification
- Re-ingest, re-process (deltas only?)

Typically, a combination of both

Big Data Universe

Hadoop

Spark
Shark (Spark on Hive) 40x

Still needs Hadoop: HDFS, YARN, Pig, Hive, HBase, JobTracker. TaskTracker,...

Open-source, batch-oriented

MPP DBs

- Teradata
- Vertica
- Greenplum

Proprietary, complex, expensive

NoSQL DBs

Column: Cassandra, Hbase, Redshift
Document/K-V: MongoDB, CouchDB, Riak
Graph: Neo4j, etc.

No tx, no referential integrity, triggers, foreign keys

In-Memory Data Grids

- Coherence (commercial
- Hazelcast (OSS)
- Terracotta (commercial)
- Gemfire (commercial)
- GridGain (commercial)
- Gigaspaces XAP (OSS)

The Origins of In-Memory Data Grids

Numatica
Scalable Software. Real-time Big Data Analytics.

Local Cache

Distributed Cache (K-V Store) + Distributed List, Queue, Topic

+

Computational Grids
Reliable jobs execution,
Scheduling
Load balancing

=

In-Memory Distributed Data/Compute Grid

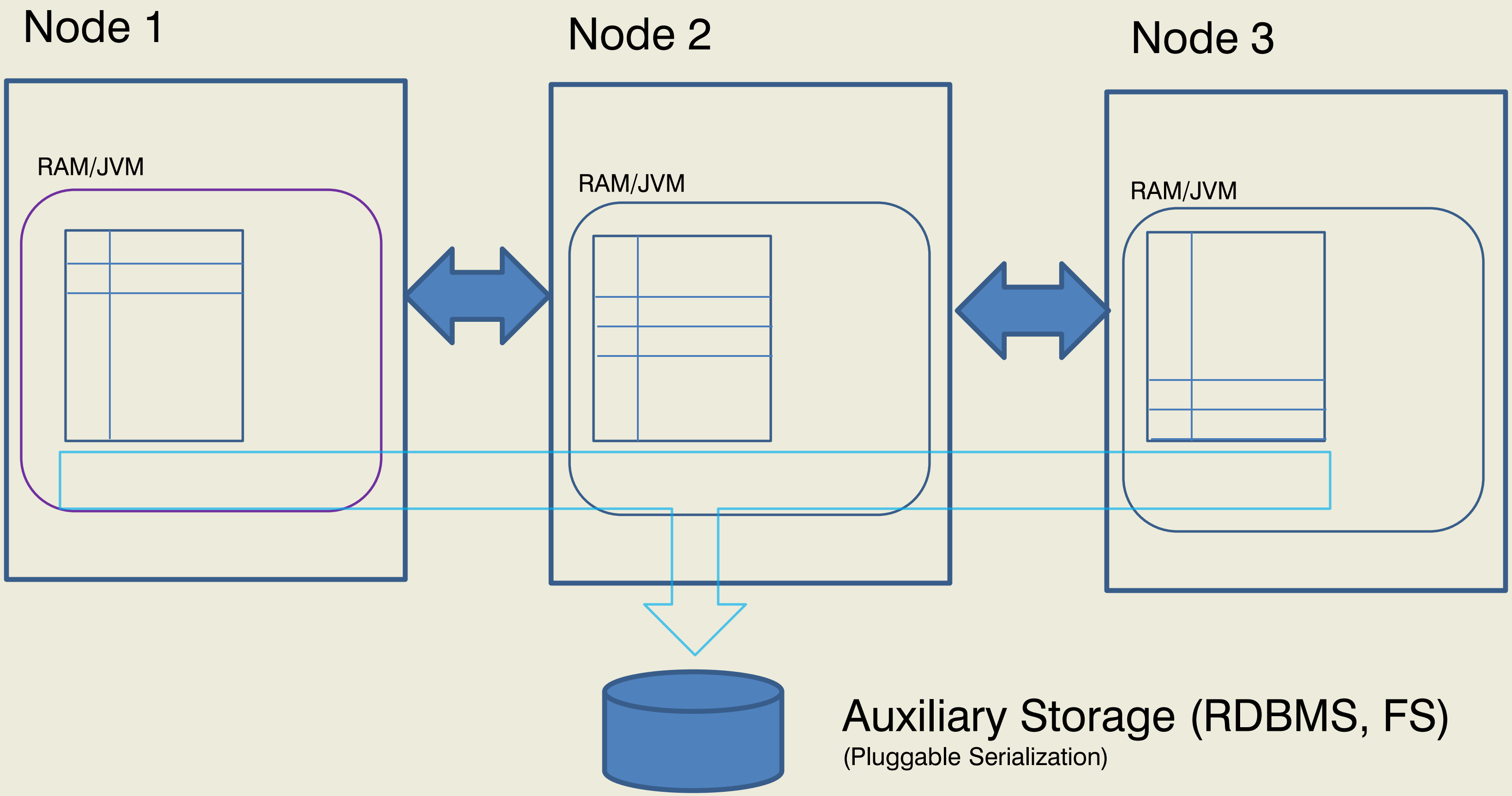
- RAM is the new disk
- DISK is the new tape

SQL (not Turing complete) MapReduce (low granularity, invasive) , often not enough

- From: Disk IO optimization
- To: Data management over network (data in a grid)

The Concept

Numatica
Scalable Software. Real-time Big Data Analytics.



- Data partitioning:
  - Logical (on key) % LOGICAL_PARTITION_COUNT (Hazelcast = 271)
  - Physical (one or more logical)
- Partition ownership, replicas reassigned on membership change
- Can determine physical location by key, to dispatch logic

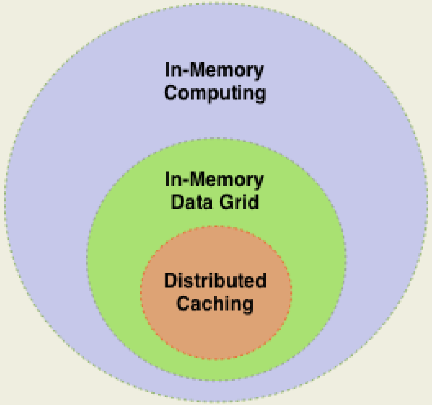
Real-time Big Data Universe

Hadoop family

Spark/Shark

Apache Storm

In-Memory Data Grids => Data/Compute Grids



- Coherence (commercial
- Hazelcast (OSS)
- Terracotta (commercial)
- Gemfire (commercial)
- GridGain (OSS, commercial)
- Gigaspaces XAP (OSS)

Traits of a Real-time Analytics platform

Numatica
Scalable Software. Real-time Big Data Analytics.

- **Low Transactional Latency**
  In-memory transaction speeds, minimize network trips, response in ms
- **Low Data Latency**
  Real-time data, rather than stale data weeks or months old
- **High Throughput, High Scalability**
  Horizontal and vertical
- **Stream processing/messaging/CEP "Continuous Query"**
- **Resilience**
  Stateless, decentralized, peer-peer - no single point of failure,
  Partitioned with backup replicas. nodes can fail without data loss and join without any data corruption
- **Elastic, simple cluster management**
  No"Mesos on top of ZooKeeper, etc."
- **Programmatic, performant SQL-like querying (data affinity)**
- **Txs , including XA (when needed)**

Major players - comparison

Numatica
Scalable Software. Real-time Big Data Analytics.

Feature	Hazelcast 3.2	GemFire 7	Oracle Coherence 12.1.3
Distributed data/collections	DistributedMap MultiMap Dist. Queue Dist. Events	No Multimap	No Multimap
Distributed Concurrency	Dist. Lock Dist. Atomic Ref Dist. Atomic Dist Semaphore	No Dist. Atomic Ref. No Dist. Semaphore	No Dist. Atomic Ref. No Dist. Semaphore
Distributed Computing	Executor Service, Key Affinity, “Map/Reduce”	Convoluted, Invasive	Convoluted, Invasive
Resilience	In-memory replicas Recovery WAN replication	Yes	Yes
Elasticity/Cluster Management	Yes	Yes	Yes
Transactions	Local + XA	Local + XA	Local + XA
Distributed Querying	SQL (subset) Predicate Continuous Query	Yes	Yes
License	Apache 2.0 + Commercial	Commercial	Commercial

And the winner is...

Numatica
Scalable Software. Real-time Big Data Analytics.

hazelcast

- The most intellectually elegant distributed in-memory data/compute grid.
- Minimalism as design aesthetic:
  Non-intrusive,
  No dependencies
  2.6MB single jar library.
- Implements Java APIs (Map, List, Set, Queue, Lock) in a distributed manner
- Distributed Execution Framework (extension of Java's Executor Service)
  Distributed Queries (SQL/Predicate), Data affinity (execution on specific node execution)
- Cluster management Java API included (dead simple)
  Auto-discovery of nodes and intelligent synchronization
- Apache License 2, commercial extensions + support

Distributed Execution Framework (1)

Numatica
Scalable Software. Real-time Big Data Analytics.

```
import java.util.concurrent.Callable;

public class Echo implements Callable<String>, Serializable {
	String input = null;

	public Echo() { }
	public Echo(String input) {
		this.input = input;
	}
	public String call() {
		return Hazelcast.getCluster().getLocalMember().toString() + input;
	}
}
```

```
ExecutorService executorService = Executors.newSingleThreadExecutor();
Future<String> future = executorService.submit (new Echo("myinput"));
...
String result = future.get();
```

Distributed Execution Framework (2)

```
import com.hazelcast.core.IExecutorService;
import java.util.concurrent.Callable;
import java.util.concurrent.Future;
import java.util.Set;

public void echoSomewhere(String input, Member member)  {
  Callable<String> task = new Echo(input);
  HazelcastInstance hz = Hazelcast.newHazelcastInstance();
  IExecutorService executorService = hz.getExecutorService("default");

  Future<String> future = executorService.submitToMember(task, member);
  Future<String> future = executorService.submitToKeyOwner(task, key);
  Map<Member, Future<String>> futures =
          executorService.submitToMembers(new Echo(input), members);

  String echoResult = future.get();
```

Execution callbacks, cancellation APIs provided

Code example:

Data Affinity(1)

Numatica
Scalable Software. Real-time Big Data Analytics.

```
HazelcastInstance instance = Hazelcast.newHazelcastInstance(cfg);
Map mapa = instance.getMap("mapa");
Map mapb = instance.getMap("mapb");
Map mapc = instance.getMap("mapc");

// different entries, but the operation will take place on the same member
mapa.put("key1", value);
mapb.get("key1");
mapc.remove("key1");

// lock operation will still execute on the same member of the cluster
instance.getLock ("key1").lock();

// consistent with distributed execution
instance.getExecutorService().executeOnKeyOwner(runnable, "key1");
```

Code example:

Data Affinity(2)

```
public class OrderKey implements Serializable, PartitionAware {
        int customerId;
        int orderId;
        public OrderKey(int orderId, int customerId) {...}
        public Object getPartitionKey() { return customerId; }
}

Map mapCustomers = instance.getMap("customers") ;
Map mapOrders = instance.getMap("orders");
mapCustomers.put(1, customer);
mapOrders.put(new OrderKey(21, 1), order);

public static class OrderDeletionTask implements Callable<Integer>,
PartitionAware, Serializable { .... }

ExecutorService es = instance.getExecutorService();
OrderDeletionTask task = new OrderDeletionTask(customerId, orderId);
Future future = es.submit(task);
int remainingOrders = future.get();
```

Distributed Query

Numatica
Scalable Software. Real-time Big Data Analytics.

```
public class Employee implements Serializable {
    private String name;
    private int age;
    private boolean active;
    private double salary;
}

HazelcastInstance hz = Hazelcast.newHazelcastInstance(cfg);
IMap map = hz.getMap("employeeMap");

Set<Employee> employees = (Set<Employee>) map.values(
            new SqlPredicate("active AND age < 30"));
// or JPA-like criteria
Predicate predicate = e.is("active").and(e.get("age").lessThan(30));
Set<Employee> employees = (Set<Employee>) map.values(predicate);
```

Code example:

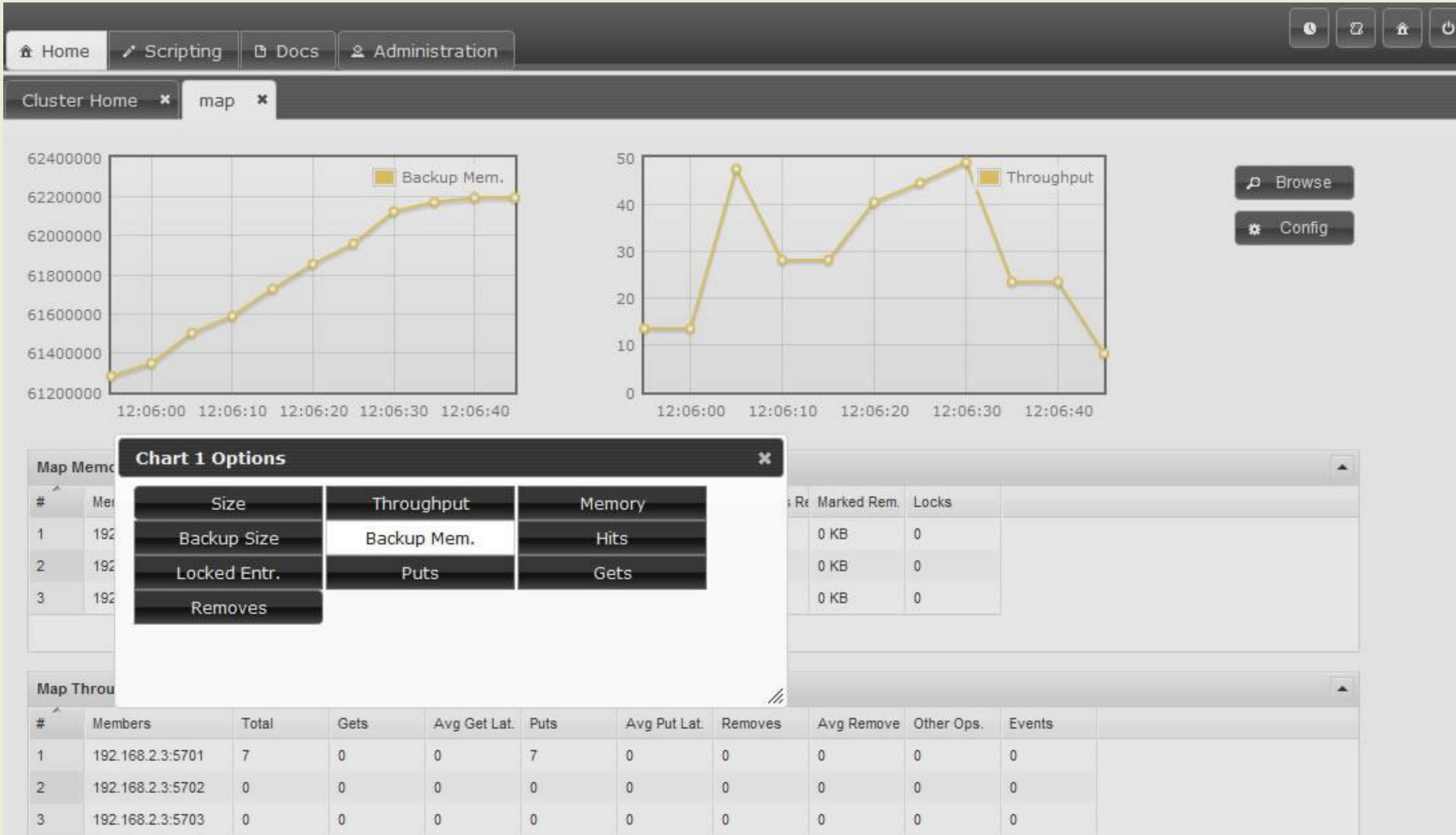
Cluster Management

```
HazelcastInstance hz = Hazelcast.newHazelcastInstance(cfg);
Cluster cluster = hz.getCluster();
cluster.addMembershipListener(new MembershipListener(){
        public void memberAdded(MembershipEvent membersipEvent) {
                System.out.println("MemberAdded " + membersipEvent);
        }
}

Member localMember = cluster.getLocalMember();
System.out.println ("my inetAddress= " + localMember.getInetAddress());
Set setMembers = cluster.getMembers();
for (Member member : setMembers) {
        System.out.println ("isLocalMember " + member.localMember());
        System.out.println ("member.inetaddress " + member.getInetAddress());
        System.out.println ("member.port " + member.getPort());
}
```

Example:

Admin console



More than just Real-time Big Data

Scalable Computing:

- DB Caching
- Web/Session Clustering
- High-speed K-V Data Store
- Distributed queues, topics
- (Fast) SOA Services

Limitations

Numatica®
Scalable Software. Real-time Big Data Analytics.

- RAM currently maxes out at ~ 640GB/server
- Still more expensive than SSDs and HDDs
- Garbage collection

- Cost and capacity limitations will disappear over time
- Off-heap memory and specialized JVMs (Azul, etc.)

From Technologies to Platforms

Numatica
Scalable Software. Real-time Big Data Analytics.

In-Memory Data/Compute Grids not enough:

- Backing DB(s)
- Enterprise Message Store(s)

Example:

GemFire

GreenPlum DB

Holy Grail: Unified Big Data Processing

Numatica
Scalable Software. Real-time Big Data Analytics.

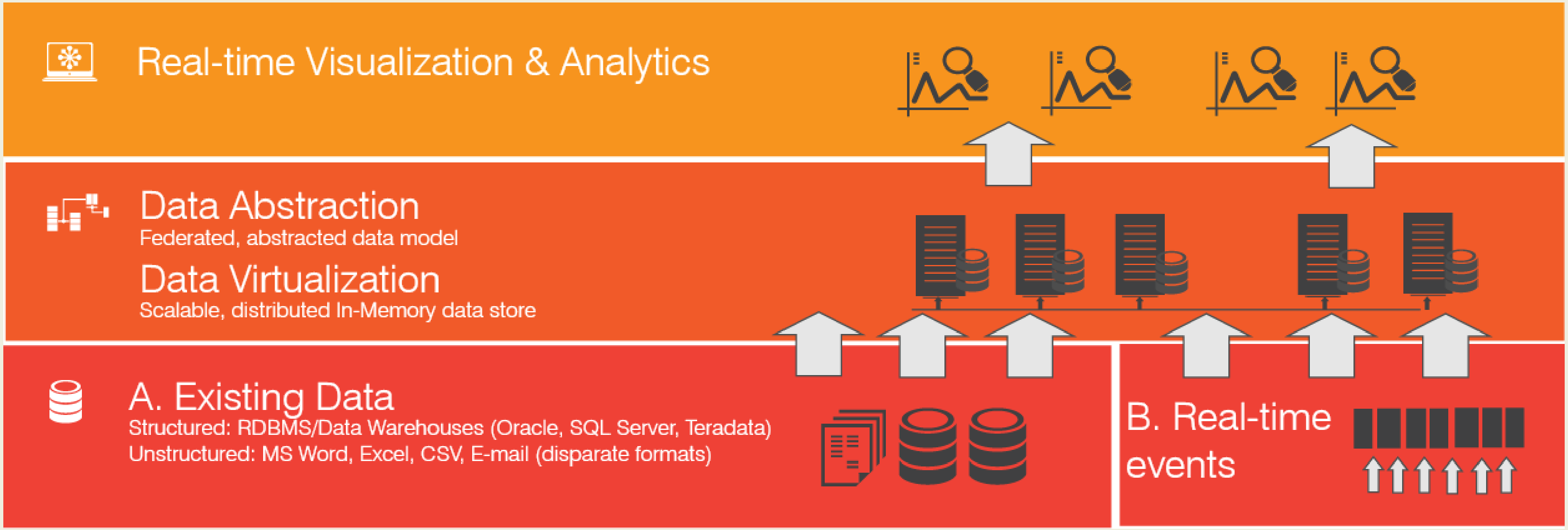
- Multi-Source Data Harvesting, Ingestion, Transformation
- Data-type agnostic:
  - Highly Structured (RDBMS data)
  - Semi-structured (documents, key-value tuples)
  - In-Motion and At Rest

- Distributed Computing, not just Data
  - Tx and non-Tx
  - Batch and/or Real-time Processing

- The fewer moving parts the better

Data Warehousing ->	Hadoop/MapReduce ->	Unified Big Data Processing
Relational + Batch	Batch	All of the above

Unified Architecture

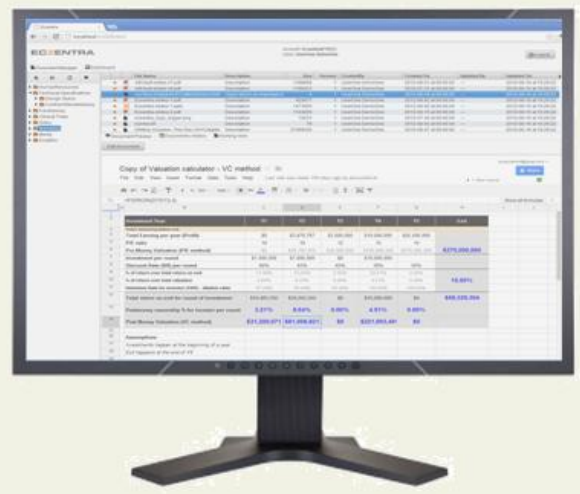


Summary

Numatica
Scalable Software. Real-time Big Data Analytics.

- One order of magnitude faster
- Data And Logic colocated, affinity-able (with careful design)
- Flexibility:
  - Turing complete
  - SQL/Predicate queries
  - MapReduce framework on top
- Scalable and Elastic, simply (automatic repartitioning)
- Current limitations (cost, GC) fading fast

Q & A



http://www.numatica.com

Appendix

Berkeley Big-data Analytics Stack (BDAS)