



Overcoming Challenges in Securities Processing



Arun Mitra, Vice President



Overcoming Challenges in Securities Processing

Next Generation Technology for the Securities Industry

A brief insight into the near future

Some Challenges

Challenge 1: Modeling data with multiple semantics

Challenge 2: Addressing the scalability problem

Challenge 3: Handling huge volumes of datasets for long running batch processes

Challenge 4: Converging to a common language for business users and developers

Challenge 1

Modeling data with multiple semantics

One size does not fit all: An RDBMS is
not always the answer

Data Modeling Challenges

- Not every data fits the relational model
- Not all data are modeled alike
- Relational model brings impedance mismatch if the application is based on objects
- Domain specific modeling, close to the domain
- Some data are better modeled as documents, some as graphs, some as key/value pairs

Alternative ways to store data

- Document Based Storage
 - Client Agreements
 - Term Sheets of Structured Products
- Graph Based Storage
 - Settlement Standing Instructions
 - Market Price
- Key Value Storage
 - Instrument
 - Custodian Intermediary Chain
 - Reconciliations
 - Data Scrubbing

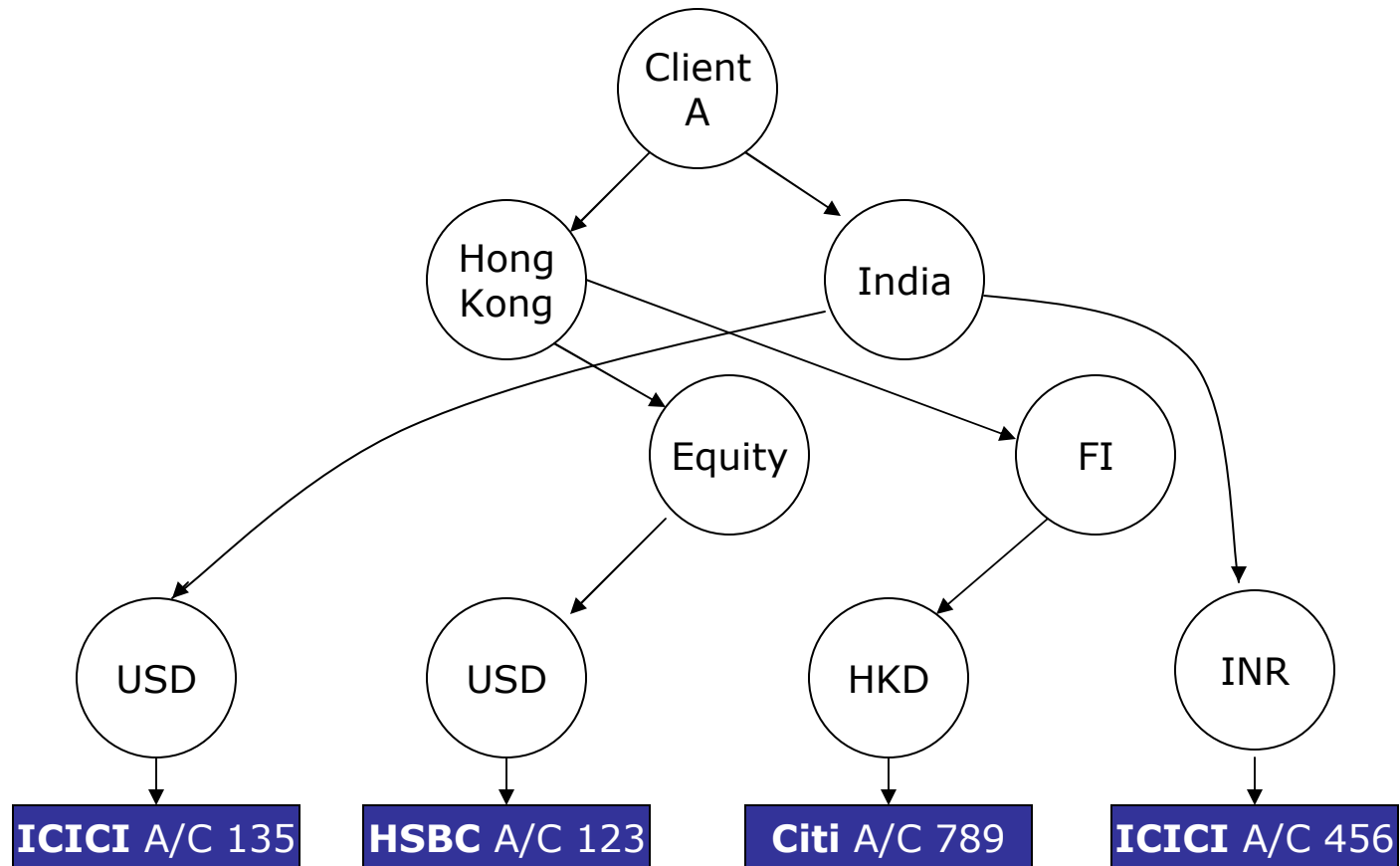
Document Based Storage

- Term sheet

```
{
  termsheet:{
    name:'100% Capital Protected Units PLUS in USD on Gold',
    riskcategory:'Complex Product',
    productcategory:'Capital Protection Products',
    securitycode:{
      swisssecurityno:4572134,
      isin:'CH0045721344'
    },
    issuer:'Credit Suisse, Zurich',
    underlying:'1 Fine Troy Ounce of GOLD',
    issueprice:'100%',
    issuedate:20090326,
    observdates:{
      observdate:[
        20090612,
        20100612,
        20110612
      ]
    },
    redemptiondate:20120326
  }
}
```

Graph Based Storage

- Settlement Standing Instruction



Key Value Storage

- Instrument

Key - Security:[Security Code:XS0431753275, Code Type:ISIN]



Serialized form of Values

Security Name: SOHO CHINA LTD 3.75% 2014/JUL/02
Security Type: Convertible Bond
Coupon Rate: 3.75
Issue Date: 20090702
Maturity Date: 20140207
Initial Coupon Date: 20100201
Coupon Frequency: Semi Annual

New paradigm in data modeling

- **NoSQL (Not Only SQL)**
 - Document based storage
 - CouchDB (<http://couchdb.org>)
 - MongoDB (<http://mongodb.org>)
 - Key value storage
 - Redis (<http://code.google.com/p/redis>)
 - Riak (<http://riak.basho.com>)
 - Cassandra (<http://cassandra.apache.org>)
 - Graph based
 - Neo4J (<http://neo4j.org>)

Why NoSQL?

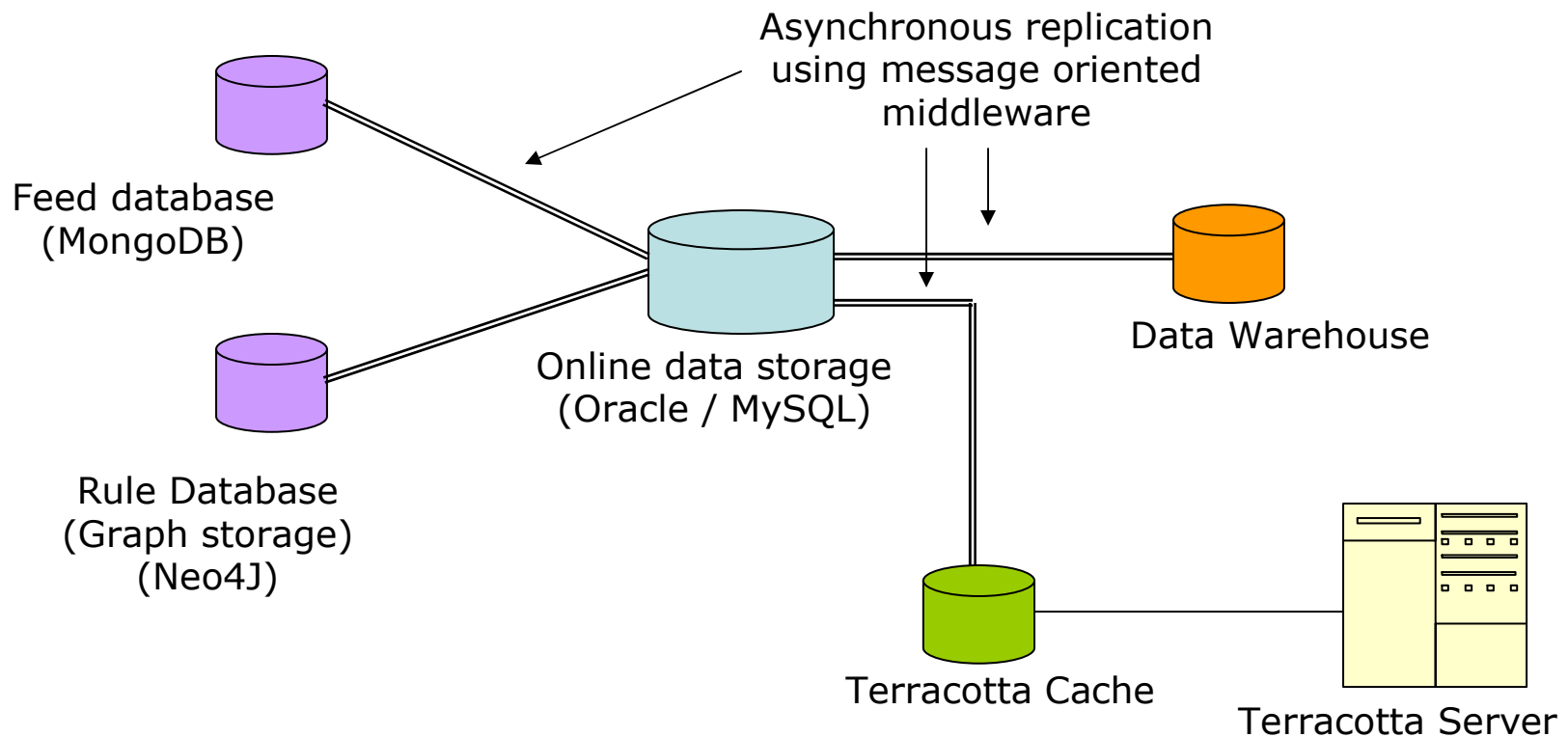
- Store data close to the domain layer
- No schema, hence no headache of schema migration
- No normalization – hence no joins
- Read and write scalable by design
- Fault tolerant through distribution and/or replication
- Lots of industry traction today

Do I throw away my RDBMS?

NO...

An RDBMS can coexist along with NoSQL stores. They address somewhat orthogonal use cases. So long we have been force fitting everything into the relational model.

Hybrid Data Modeling Architecture



<http://debasishg.blogspot.com/2010/01/new-way-to-think-of-data-storage-for.html>

Challenge 2

Addressing the scalability problem

Scalability of the data tier, application
tier and the Web tier

and the infrastructure

The Bottlenecks

- RDBMS is the single point of bottleneck
- Latency kills throughput
- Synchronous blocking database writes do not scale
- Web session clustering is still a challenge

In-memory data grid for Scaling Out

- POJO based programming model
- In process transactional distributed caching
- Asynchronous write-behinds to database
- Linear scalability
- Data availability through clustering
- Terracotta, Gigaspaces, Coherence ...

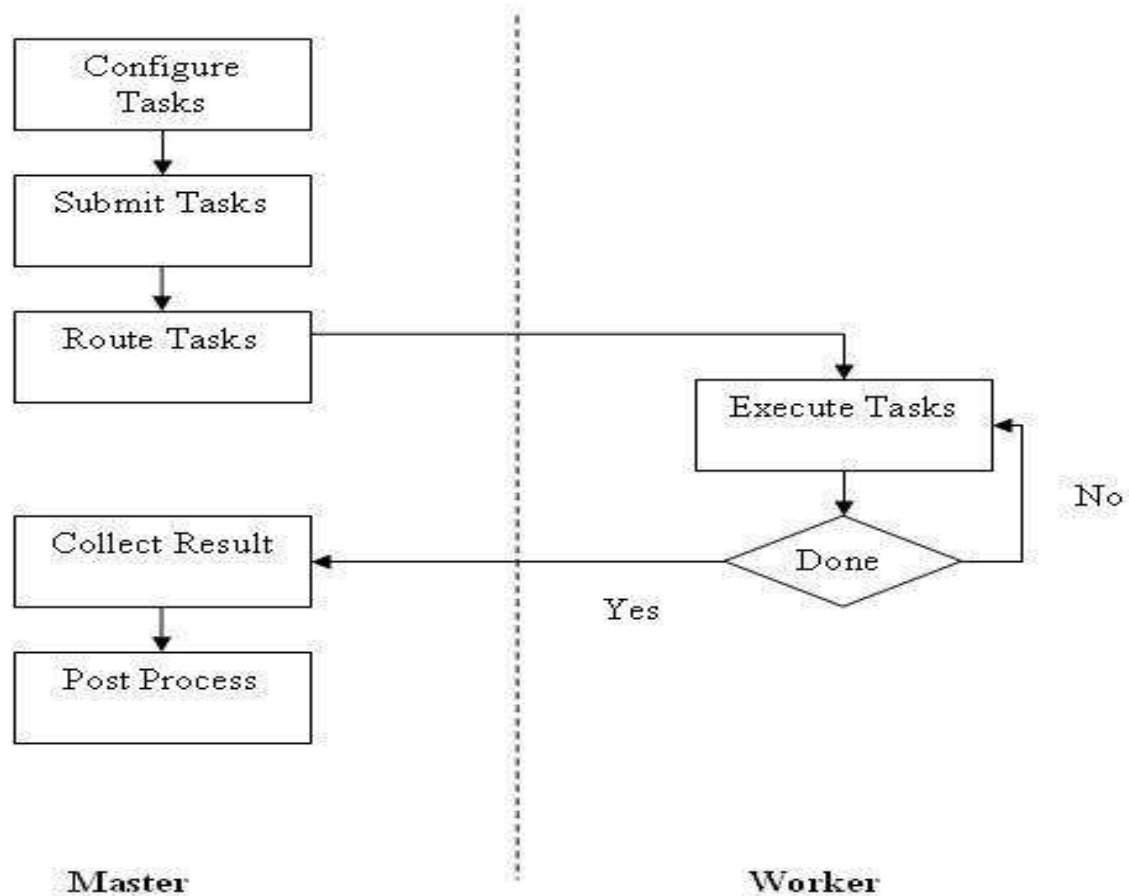
Distributed Workflow: in-memory data grid

- Offers a ready-to-use distributed service and batch process infrastructure
- Tasks that are *independent* and can be executed in *parallel*
 - Map/Reduce style batch jobs e.g. Mark to Market can be split across offices and then consolidated in the reduce phase

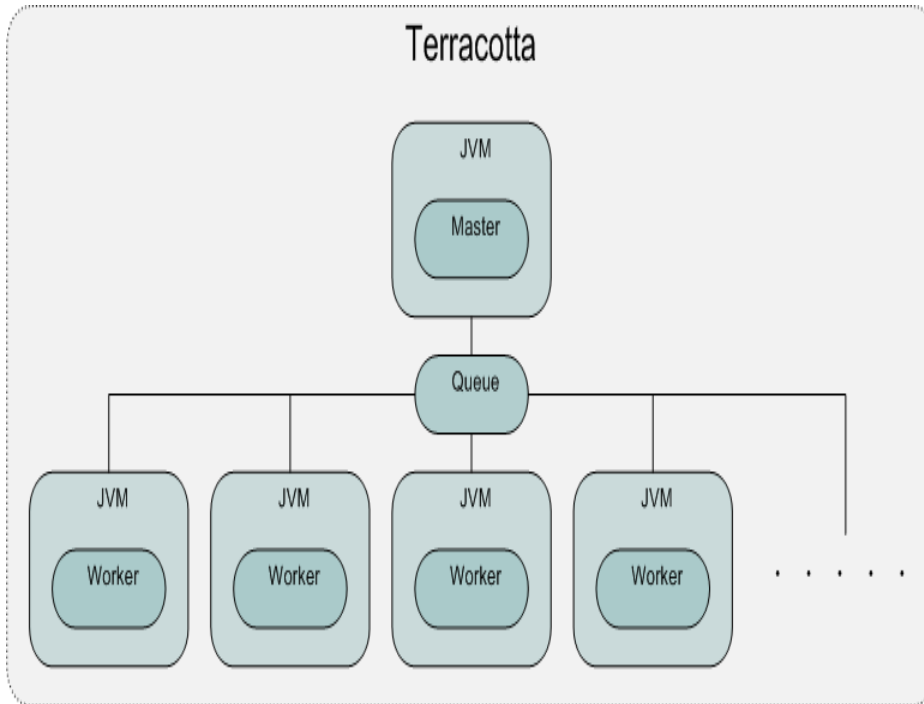
Master/Worker Pattern

- **Master**
 - Initiates processing by putting a request in shared space
- **Worker**
 - Picks up the task from shared space and does the work
- **Involves multiple JVMs**
 - Terracotta makes them appear like single unit

Master/Worker Workflow



Master/Worker with Terracotta



- JVM instrumented by TC
- Shared work queue
- Add additional masters
- Easy to monitor
- More flexible allocation of tasks

Managing Infrastructure for Scalability

Deploying on the cloud

Models of Deployment

- **Amazon EC2 approach**
 - Hypervisor level isolation amongst tenants
 - Isolation at a lower level
 - Freedom of choice on technology stack
- **Salesforce approach**
 - Database level isolation amongst tenants
 - Easier to implement, but sharing of database may be a concern to tenants

Usage Model

- Web applications deployed on the cloud e.g. the SaaS application
- Client applications such as local backup tools that store data in the cloud
- Direct/abstracted access, such as a local folder synchronized with cloud storage (e.g., Dropbox), or VPN access to a cloud-based server.

Business Delivery Model

- **SaaS**
 - Use providers' applications running on cloud infrastructure and access through a thin client like web browsers (e.g. Salesforce.com)
- **PaaS**
 - Deploy onto the cloud infrastructure consumer created applications using the languages and tools supported by the provider (e.g. Google App Engine model)
- **IaaS**
 - Provision processing, storage, networks where the consumer is able to deploy his own application (e.g. Amazon EC2 model)

Deployment Models

- Public
- Community
- Private
- Hybrid

Security Issues

- Requirements
 - Confidentiality
 - Availability
 - Integrity
 - Non-repudiation

- Making secure at
 - Database level
 - Internet level
 - Program level
 - Server level

Security Issues – How to Handle

- Encryption
- Controlled access
- Audit tools for monitoring
- Policy enforcement on data storage and access

Challenge 3

Handling huge volumes of dataset for long running batch processes

Reconciliation
Mark to Market
Valuation
Monthly Closing
Etc...

Large scale Analysis of Structured Data

- Reconciliation of data from multiple sources
- Data cleansing and scrubbing
- Need to process large data sets
- Need low latency streaming data access
- Mostly *write-once-read-many* access models
- Need to be portable across heterogeneous hardware and software platforms

Hadoop

- Distributed processing framework from Yahoo! Inspired by Google Map/Reduce
- Open Source
- Parallelizes data processing across multiple nodes in a cluster, speeding up computations and hiding latency through increased concurrency

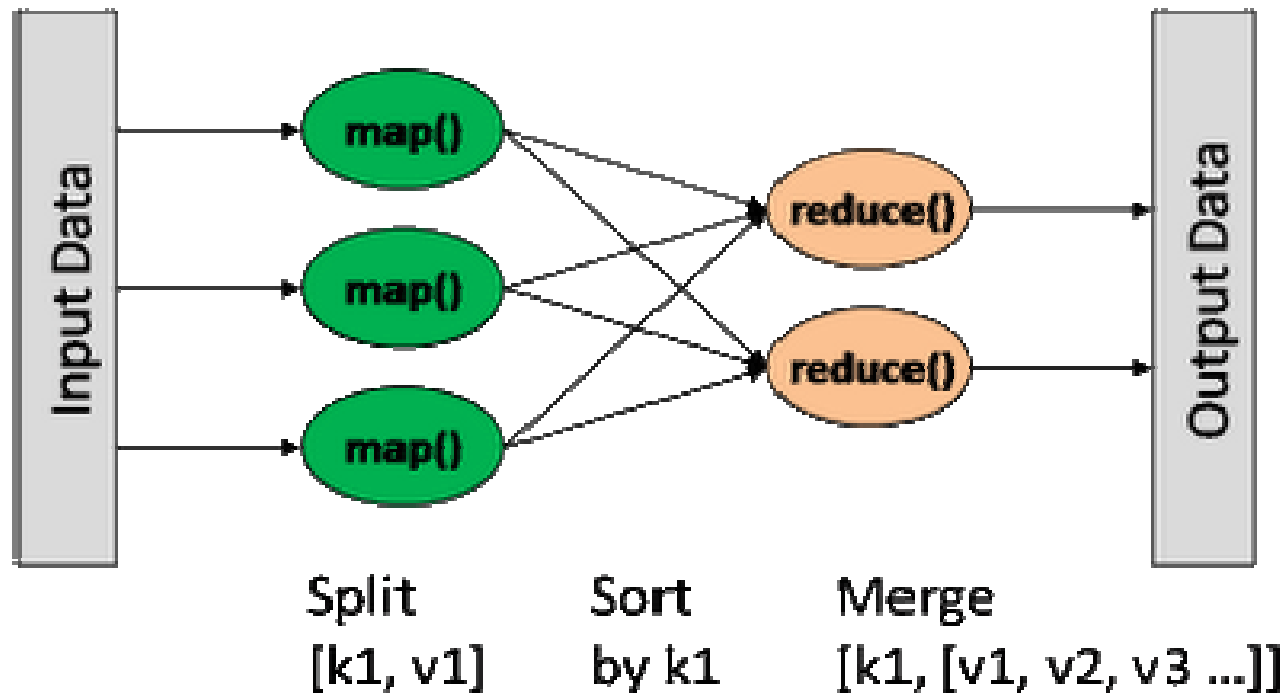
Features

- Resilient to node failures
- Can handle large data sets
- Fast
- Horizontally scalable – just add more machines

Features

- Built on top of HDFS, Yahoo's distributed file system
- Users write map/reduce jobs that process data stored in HDFS
 - 'map' function transforms a piece of data into key/value pairs
 - 'reduce' merges same keys into a result

△ Hadoop Map/Reduce



Challenge 4

Converging to a common language for Business Users and Developers

Using Domain Specific Languages

What is a Domain Specific Language

"A language of limited vocabulary implemented for a specific domain"

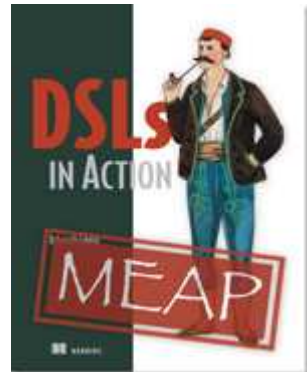
This actually runs!

- Buy 100 shares of IBM at limit-price 100 USD
- Sell 100 bonds of GOOG at minimum-price 80 USD
- Settle broker Chase trades in JPN market safe-keep security with our account ABC-123 and cash externally at BOJ account BOJ/1/BC

Advantages

- Minimize gap between developers and business analysts
- Business analysts can actually verify the domain logic before the code gets shipped
- Results in better software, which is easily maintained – business rules more explicit

△ DSLs in Action



by
Debasish Ghosh
CTO, Anshinsoft

- Coming up from Manning (<http://www.manning.com/ghosh>)
- An extensive discussion on DSL design in securities trading applications
- MEAP available NOW, print version this summer