Experiences of a production deployment: Multi-tenant software on multi-tenant hardware

Gary Kendall CD02

Dan Creswell

Lone Crusader



Overview

- The problem Structured Credit
- The technology
 - Excel
 - Web Services
 - JINI
- Implementation
 - Sun Grid
 - Case Study
- Deployment infrastructure



Structured Credit



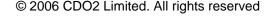
What are Credit Products?

- Corporate Bonds
 - Lending a company money
 - Coupon depends on the credit worthiness of the company
- Credit Default Swap (CDS)
 - Buy or Sell protection on a company
 - No money is actually lent to the company
 - Form of insurance



What about CDOs?

- Like Bonds and CDS but...
 - Contain a portfolio of companies
 - Payout dependent on some loss parameters
 - Number to default
 - Percentage of losses
 - Range of loss (e.g. 3%-6%)
 - Very hard to price!





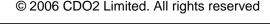
Pricing Credit Derivatives

- Build probability distribution of default
 - Bond prices
 - CDS prices
 - Interest rates
- Predict future cash flows to get a price
 - Easy for a single company but...



Pricing CDOs

- How do I build a probability distribution for a large number of companies?
 - Numerical integration over all of the companies' probability distributions
 - Simulation over all the possible outcomes
- Very CPU intensive





Technology



Excel

- You'll never take the spreadsheet away from the trader...
 - Flexible
 - No programmer required
- But...
 - Not scalable
 - Error prone



Web Services

- The power of XML
 - Easy integration with Excel
 - Open up server side power
- But...
 - How do you make it truly scalable?





JINI and JavaSpaces

- The perfect Grid infrastructure
 - Service discovery: resources can join or leave transparently
 - State persistence: data and calculations can be shared amongst users
 - Transactional behaviour: node failure is handled transparently
 - Open source: different vendors' implementations can be used without code changes



Solution

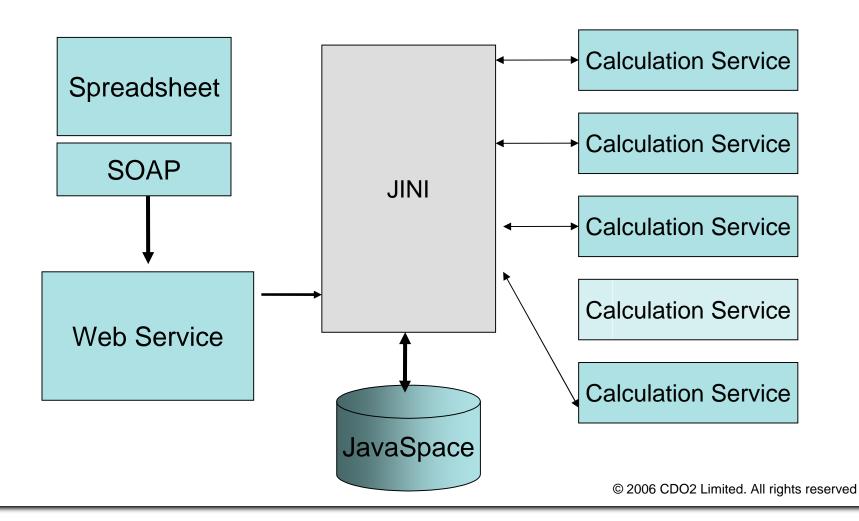


Putting the pieces together

- Excel
 - SOAP calls implemented as Excel functions
- Web Services
 - Proxy for JINI requests
- JINI
 - Calculations performed across the network
 - Market data and results shared seamlessly



CDOSheet Architecture





Technology Choices



Spreadsheet

- Microsoft Excel
 - Universally accepted
 - Ready installed client base
 - Web Services support via SOAP



Web Services

- Tomcat
 - Widely adopted
 - Pure Java
- AXIS
 - Generates WSDL from Java code
 - Easy integration with JINI





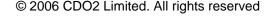
JINI and JavaSpaces

- Sun JSK
 - Core JINI Services
 - Lookup and discovery
 - Transaction Management
- Blitz
 - Scalable JavaSpace implementation
 - Open source licensing
 - Commercial support available



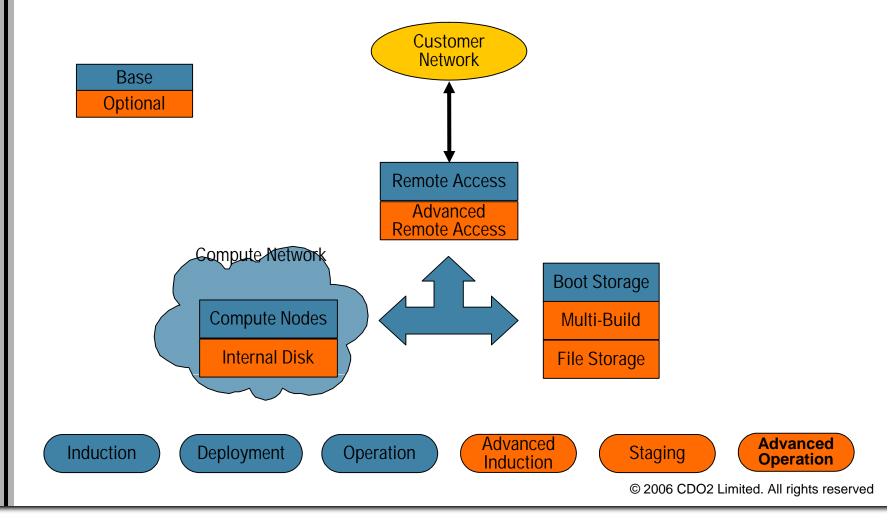
Compute infrastructure

- Sun Grid Commercial Compute Utility
 - Predictable pricing (\$1 per CPU hour)
 - Scalable business model
 - Resilient infrastructure
 - Flexible deployment model
 - Enterprise class support





Sun Grid Commercial Compute Utility





Availability Requirements

- High Availability
 - Reduce single points of failure
 - Minimize downtime on failure
 - Manual intervention tolerable
 - Reduce scope for error



Deployed Availability

- Redundant network
 - Dual path from ISP down
- Redundant power
 - Dual electricity substation + diesel backup
- Redundant disks
 - Dual path Fibre Channel RAID array
- Redundant servers
 - Dual servers for core services (manual failover)



Case Study



Sachsen LB Europe

- Dublin based subsidiary of East German Landesbank
- Manages €1bn plus CDO portfolio
- Desktop IT infrastructure
- Needed independent and reliable pricing and risk



Sachsen Case Study

- Set up trial subscription to CDOSheet and market data to evaluate system
- Initial production deployment with 10 CPUs per hour
- Upgraded to 100 CPUs per hour as business grew and demand for risk analysis increased



System topology









Sachsen LB Europe plc









Summary

- Grid computing solution delivered as Software as a Service (SaaS)
- Multi-tenanted grid environment
- Scalable business model
- Incorporates third party web services (e.g. market data)
- Secure internet deployment



Deployment Infrastructure

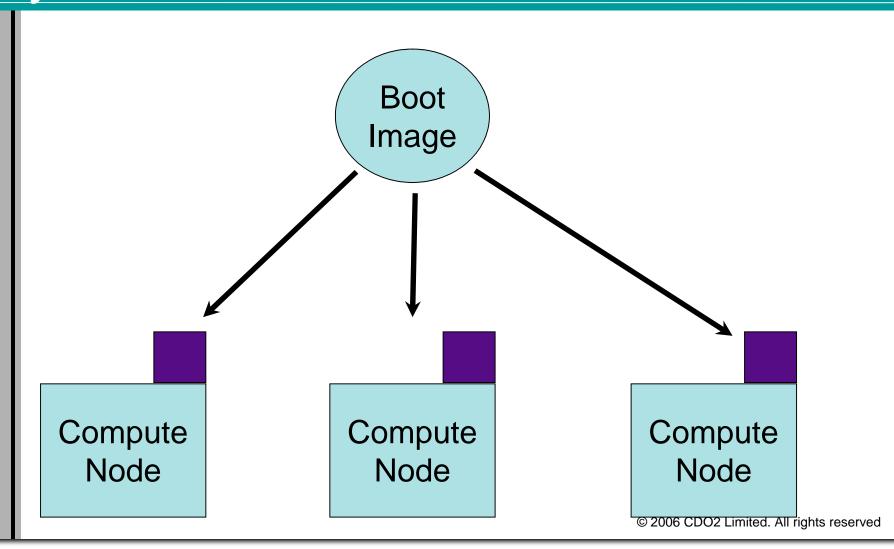


Systems Management

- So many machines
 - So few administrators
- Multiple Customers
 - Each with different software loads
- Constantly Changing
 - Customer trials
 - On-demand compute model
 - Differing compute power per customer



System Model





Systems Model

Control JavaSpace

Lookup

Control TxnMgr

Compute Node



Agent Startup

- Command-line
 - Id (possibly unique)
 - Unicast Lookup URL
 - Locate JavaSpace
 - Locate TransactionManager



Agent Requests

- Deploy Package
 - Target
 - machine to deploy on
 - Classpath
 - Codebase
 - Classname
 - Instantiated with newInstance
 - Implements ServiceLifecycle
 - Duration



Agent Requests

- Stop
 - Target
 - A polite request to ServiceLifecycle



Agent Behaviour

- Deployment Request is taken under transaction
- Transaction remains open for lifetime of deployment request
- Compute node failure causes transaction lease to expire



Agent Requests II

- Show
 - Lists outstanding (or returned) Deployment Requests
- Cancel
 - Remove an outstanding (or returned)
 Deployment Request



Agent Behaviour

- Successful completion commits the transaction
- ServiceLifecycle issues causes commit of transaction and posting of DeploymentError
 - Can't leave request in space otherwise we get repeated failure and error report



Future Work

- Determine the best way to handle groups of compute nodes
- Simplest failure model
 - Recovery Oriented Computing
 - Still some fine tuning in respect of exact behaviour in context of various failures
 - Serious system failure causes brutal exit
 - Possible need for multiple lookup URLs

