Migrating Scientific Applications from Grid and Cluster Computing into the Cloud

Issues & Challenges

T S Mohan, PhD
Principal Researcher, E&R

ISGC2011 - 24 March 2011
Overview

Dilemma of Scientists
- Use existing Grid / Cluster Applications, Libraries and Infrastructures
- Start from Scratch & Develop Application for the Cloud

Migrating Scientific Applications into the Cloud
- Challenges, Issues and Trends

Our Methodology and Approach
- Relevant to certain classes of Scientific Applications
- Best Practices
What is the Difference?

Grid or Cluster based Scientific Applications

Vs

Scientific applications leveraging existing Cloud Services

- Grid/Cluster Scientific Applications
  - Use Existing Libraries
  - Use Existing VOs and Infrastructure
  - Stagnant Development
  - Limited
  - Not efficient
  - Incompatible
  - Non-Interoperable

- Cloudonomics
  - OPEX vs CAPEX
  - Governmental Funding...

- Scientists & Core Competencies
  - MindSet Change
  - Models to use
Can Scientists used to Grids Leverage the Cloud Effectively?

Cloudonomics
- ‘Pay per use’ – Lower Cost Barriers
- On Demand Resources – Autoscaling
- CAPEX vs OPEX – No CAPEX & Only OPEX.
- SLA driven operations – Much Lower TCO
- Better Availability and Reliability

Technology
- ‘Infinite’ Elastic availability – Compute/ Storage/ Bandwidth
- Automatic Usage Monitoring and Metering
- Jobs / Tasks Virtualized and Transparently ‘Movable’
- Integration and interoperability ‘support’ for hybrid ops
- Transparently encapsulated & abstracted IT features.
Cloud Computing – Software Development Lifecycle
For Certain Kinds of Scientific Applications for the Cloud

Cloud Oriented Development Life Cycle

- Requirements
- Architect
- Customize
- Design & Align Policies
- Assemble and implement
- Test, Verify, Validate
- Deploy
- Monitor

At what level (Code) do we migrate?
What kind of cloud do we migrate into?
What is Migration into the Cloud?

- Cloud - Integration / Adoption / Migration
- Migration : Code, Design, Architecture, Usage

\[ P \rightarrow P'_C \rightarrow P'_{OFC} \]

where

- \( P \) is the Application before migration
- \( P'_C \) is the Application after migration either into a cloud or hybrid cloud
- \( P'_{OFC} \) is the Application after migration and optimized to leverage the advantages of the cloud

- Technical Challenges
  - Portability of Architecture, Design, Code, Usage as well as Public, Private & Hybrid Cloud Environment we wish to migrate into.

- Only certain classes of Scientific Applications are appropriate
  - Data parallel Applications
  - Unpredictable Run times to completion— either too much demand on computational resources or none - no fixed time runs.
  - No ‘unstructured message passing’ or too much dependence on Virtual Organizations
  - Secure sharable data – or data that is already in public
The Middle Path!

Mindset Change!

Migrate core parts of existing Grid Applications / Libraries to leverage Cloud Benefits

Leverage Grid Patterns for Cloud Compute, Storage & Communications

Cloudonomics!

Our Neutral Approach – The 7 Step Migration Model Applicable Across Cloud Services (Private &/or Hybrid) AWS, Azure, GAE, Eucalyptus, Hadoop oriented Apps
The 7-Step Migration Model © Infosys Research

1. Conduct Cloud Migration Assessments
2. Isolate the Dependencies
3. Map the Messaging & Environment
4. Re-architect & Implement the lost Functionality
5. Leverage Cloud Functionalities & Features
6. Test the Migration
7. Iterate and Optimize
“The 7-Step Migration Model” in Brief

© Infosys Research

Assess
- Cloudonomics
- Migration Costs
- Recurring Costs
- Database data segmentation
- Database Migration
- Functionality migration
- NFR Support

Isolate
- Runtime Environment
- Licensing
- Libraries Dependency
- Applications Dependency
- Latencies Bottlenecks
- Performance bottlenecks
- Architectural Dependencies

Map
- Messages mapping: marshalling & de-marshalling
- Mapping Environments
- Mapping libraries & runtime approximations

Rearchitect
- Approximate lost functionality using cloud runtime support API
- New Usecases
- Analysis
- Design

Augment
- Exploit additional cloud features
- Seek Low-cost augmentations
- Autoscaling
- Storage
- Bandwidth
- Security

Test
- Augment Test Cases and Test Automation
- Run Proof-of-Concepts
- Test migration strategy
- Test new testcases due to cloud augmentation
- Test for Production Loads

Optimize
- Optimize – rework and iterate
- Significantly satisfy cloudonomics of migration
- Optimize compliance with standards and governance
- Deliver best migration ROI
- Develop roadmap for leveraging new cloud features
The 7-Step Migration Model Explained

• Conduct Cloud Migration Assessments
  • Cloudonomics: Migration Costs and Long-Term Recurring costs
  • Database data segmentation, security & migration
  • Functionality migration and Non-Functional Requirements support

• Isolate the Dependencies
  • Runtime Environment
  • Virtual Organizations and Policies
  • Licensing
  • Libraries, Application and Code
  • Communication latency & Performance dependencies
  • Architectural dependencies
The 7-Step Migration Model Explained .. Contd

- Map the Messaging and Environment
  - Marshalling /demarshalling & mapping parameters of communication -
    Adapters
  - Environment parameters approximation maps
  - Libraries and runtime mapping – exploiting data parallelism where
    possible
- Keep it as loosely-coupled and coarse-grained
- Re-Architect and Implement the lost functionality
  - Approximate the lost functionality in terms cloud support
  - Not all grid features can be migrated – Need Mindset Change for that!
  - Re-architect bridges and new code
- Leverage Cloud Functionalities and Features
  - Augment application to exploit cloud features
  - Low cost - big impact cloud features enhance usage
The 7-Step Migration Model Explained .. Contd

• Test the Migration
  • Augment test cases and test automation
  • Run Proof of Concepts & then scale to production size
  • Augment test cases suite with cloud functionality usage
  • Validate both the migration strategy and augmentation

• Optimize
  • Optimize the migration strategy – rework & iterate
  • Satisfactory Cloudonomics for a broad range of usage?
  • Standards compliance & Governance issues
  • Develop a roadmap for leveraging new cloud features
  • Iterate to deliver best ROI
Best Practices for Migrating into a Cloud

• Best Practices
  • Design for failures – underlying cloud systems are commodity items
    • Avoid Single Points of Failure in your Applications
  • Develop loose coupling between applications / code / services
  • Exploit key cloud features: scaling, elasticity, network locality and location independence, anonymity, support for hybrid clouds, etc
  • Build security, reliability and other non-functional requirements at every level and layer
  • Rethink architectural constraints to avail cloud benefits
  • Iterate and Optimize
  • Beware of Vendor-Lock ins, Data Security Issues, SLAs and Pricing Honeypots

• Migrating into the Cloud is not a trivial task – Leverage best practices – EMI, DCI etc
• Migrating into the cloud – major activity amongst most software firms and IT departments of various enterprises – Lots of individual experiences – IP challenges
• Migration Complexities are software engineering challenges – Approaches exist!
• Optimal Migration yield best ROI for using Cloud offerings
• Data and Application migration is pretty popular while more challenging is the Architecture and Design migration for PAAS and SAAS platforms – it is still evolving.
THANK YOU!