Clouds Provide Grids with Higher-Levels of Abstraction and Explicit Support for Usage Modes

S Jha*, A Merzky, G Fox
In a Nutshell

• First attempt to characterise Clouds in relation to Grids from the perspective of system semantics and interface abstractions

• Clouds provide interfaces that are syntactically simple, semantically restricted and high-level

• Introduce:
  • Centrality of Usage Mode: Principal pattern of usage, access
  • Affinity: System's internal design principle to support usage modes
    • elements of Model-of-Computing, QoS, SLA
• Clouds: Emphasis on Usage Mode and Affinity!
  • Simplicity of Clouds
Outline

- A perspective on the status of Grids
- Some background formalism
- Use semantic ordering in an attempt to tighten fuzzy relationship between Grids and Clouds:
  - Semantic Ordering:
    - Implications of this ordering: Explicit support for Usage Modes
    - Our perspective on Clouds, Grids & all that Jazz..
- Observations/Musings
Perspective on the status of Grids

• What is the status of Grids (or Grid vision)?
  • As always: It depends...

• Lack of applications uptake & deployment correlated with challenges in the deployment, provisioning and management of resources
  • Be careful of over-simplification of the causes..

• “It’s the complexity, stupid”
  • Programmatic, deployment and management

• Level of exposed detail is too great!
  • e.g., Web-Services are just too fine-grained on which to deploy Grids and build applications
  • Need abstractions to hide levels of detail and provide functionality in a simple way
Some Formalism

- **Resource**: A physical or virtual entity of limited availability
  - Physical: storage, network..
  - Virtual: usually services
- **Services**: An entity which provides a (specific) capability or which allows an action on a resource
  - High-level service: act primarily on physical resources
  - Low-level service: act primarily on virtual (ie services)
- **System**: A set of services and resources which form an integrated whole
  - Systems are inherently hierarchical (high-level, low-level)
Some Formalism (2)

- **Semantics of Systems**: The set of capabilities, or features, available within a system
  - Semantics of a system (high-level) can be more powerful than the semantics of the individual (low-level) systems
- **System Interfaces**: A means of accessing the capabilities of a system
  - APIs provide programmatic access to interfaces
  - App. Hosting Envn. provide user-level abstractions to API
- **Virtualisation**: Layer between systems and applications that translates concurrent access into seemingly exclusive access to the virtual system
Relating the Concepts..
Yet more formalism...

- General versus Narrow Grids:
  - maximal set of semantics vs Limited semantics
  - Possibly domain specific set of semantics: TG vs. Data-Grid
  - Narrow Grids = General grid + high-level services
    - Limited semantics, but greater (ease) use
- Usage Mode: Commonly occurring resource access and deployment patterns for applications
  - Eg parameter-sweep, logical coupling of components..
- Affinity: Inherent system property, describing relationship between resources and computation types possible w/o system details
  - MOC presented with focus on interfaces not on implementation
  - Dominant design guideline to support usage mode
    - indicative of the type of UM supported by systems
    - eg high-throughput affinity, fast-turnaround affinity, bulk-storage
  - Clouds support at least one affinity;
    - Corollary: Interface designed to support at least one usage mode
Observation

- Observation I: System Interfaces expose a set of semantic/features as required by the target applications
- Observation II: Higher-level system tend to support more specific, target applications and usage modes
- Observation III: Narrower a Grid and its system interfaces, the easier its use tends to be
- Clouds: Systems with minimalistic interfaces & no system internals
  - Type of narrow Grids with support for explicit UM
    - eg S3 a type of data-grid with less exposed semantics compared with regular (CERN, LHC) data-grids
- Clouds: Affinity: Limited application scope ==> Support for specific UM
  - Limited system interfaces ==> easy to use
- Grids: Your favourite definition here..
  - Wide application scope ==> Numerous usage modes
    - Rich system interfaces ==> not so easy to use
Semantic Ordering

Application Environments

Clouds

Grids

Operating System

OGC Portal
Workflow Environment

Application Domain Specific Solutions

Storage Cloud
Compute Cloud

Domain Specific Services

Data Grids
Campus Grids

Higher Level Services

General Purpose Grids

Lower Level Services

Compute Elements
Storage
Network

Resources (physical)
Usage Modes

- General purpose Grids are typically constructed bottom-up: aggregating existing heterogenous resources
  - Interfaces designed to provide combined functionality
- Clouds constructed top-down with a limited, specific set of use-cases and modes
  - Interfaces are designed to support these and only these
  - System itself maybe designed with single use case
    - Could be homogeneous
    - Homogeneity does not imply simplicity; issues of scale
    - Many of the same challenges of GP Grids, just that system internals are not exposed at the interface level
- Clouds (as of now) target single usage mode
  - Single usage-modes influence current design guidelines
A perspective on the status of Clouds

- Clouds do not have to be associated with an underlying Cost-model
  - i.e. clouds to not have to be commercial or a type of utility (on-demand) computing
- Clouds can be separated from provisioning details
  - Explicit service, cycle-scavenging..
- “General Purpose” Clouds don't make sense..
  - Role for “other specific” clouds
  - Clouds will not make everyone happy
- Clouds can be built on top of Grids..
  - Disregard whether a business or technical model...
Lessons

- **Resource Providers:**
  - Look at the target user space, and usage modes

- **Application Developers:**
  - Use highest level interfaces

- **Role for OGF:**
  - Standardization at the interface-level
    - SAGA & extensions to data-management systems
  - Standardization of the core-capability level
    - Sch & Res, HPC-BP, BES, DRMAA
Conclusions

• Clouds provide higher-level of abstractions
  • .... provide explicit support for usage modes
  • .... are a logical evolution of the Grid concept
• Many unanswered questions:
  • Models of computing supported ie affinities? How?
  • Couple different clouds:
    • Different providers? Different affinities?
    • Simple interoperation: Models of aggregation?
  • How to provide high-levels abstractions (to support access and/or usage patterns)?
Acknowledgments

- Funding Agencies:
  - UK EPSRC:
    - eSI Theme “Distributed Programming Abstractions”
    - OMII-UK “OMII SAGA” project
  - US NSF
    - Cybertools projects
  - CCT Internal Funds