Review and Future Directions

Grid Computing (M)

Lecture 20

UNIVERSITY
of
GLASGOW

Lecture Outline

- Aims and objectives
- Intended learning outcomes
- Review of lectured material
  - Background
  - Large scale systems architecture
  - Security
  - Resource management
  - Scalability and heterogeneity
  - Systems modelling and simulation
- Review of tutorial material
  - Globus, Condor, Permis, OGSA-DAI
- Future Directions in Grid Computing

Course Aims and Objectives

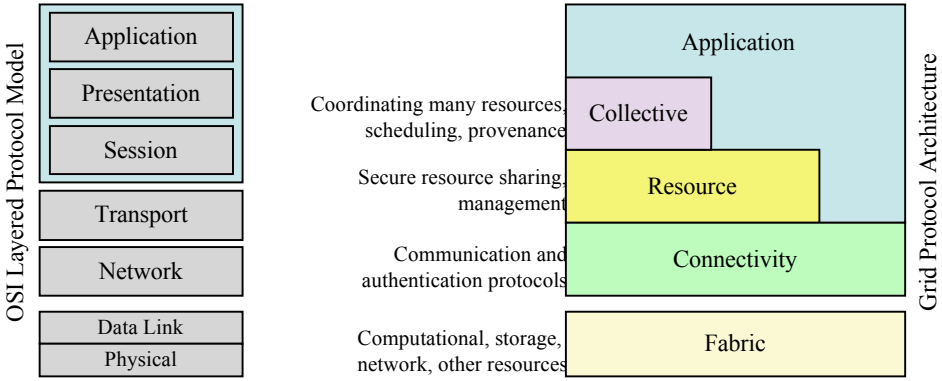
- To provide the participants with:
  - Detailed understanding of the key problems and issues that arise when attempting large-scale distributed computation, within organisations & across organisational boundaries
  - Insight into the architectural implications of Grid-scale computation
  - Awareness of current research issues in:
    - Grid architecture and infrastructure
    - Scalable distributed computation
    - Integration of applications across autonomous organisations
  - Practical experience of current Grid technologies and associated standards
  - Skills in utilising current Grid tools and technologies
  - Appreciation of the weaknesses of existing tools and technologies, and potential areas for improvement

Intended Learning Outcomes

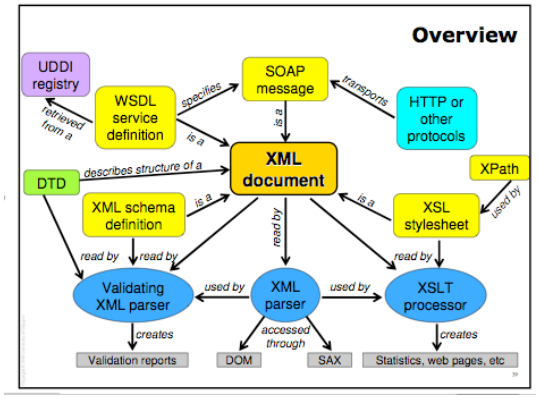
- By the end of this module, participants should be able to:
  - Critically discuss and reason about large-scale distributed system architectures, infrastructures and technologies
  - Articulate research challenges in multi-organisational distributed computing, including Grid computing
  - Design and implement Grid computing applications using Globus or similar toolkits
  - Justify the applicability, or non-applicability, of Grid technologies for a specific application

Background Material

- Networking and communications
  - IP networks: routing, addressing, QoS, congestion, security
  - Transports: TCP, UDP, etc.
  - Sockets API
- Remote procedure calls
  - Java RMI as motivating example
  - Parameters, naming, exceptions
- Distributed systems architecture
  - Designing distributed algorithms
  - Performance and autonomy
- Mark up languages and XML
  - Well-formed & valid documents
  - DTDs, schemas and namespaces
- Web services
  - SOAP, WSDL, UDDI

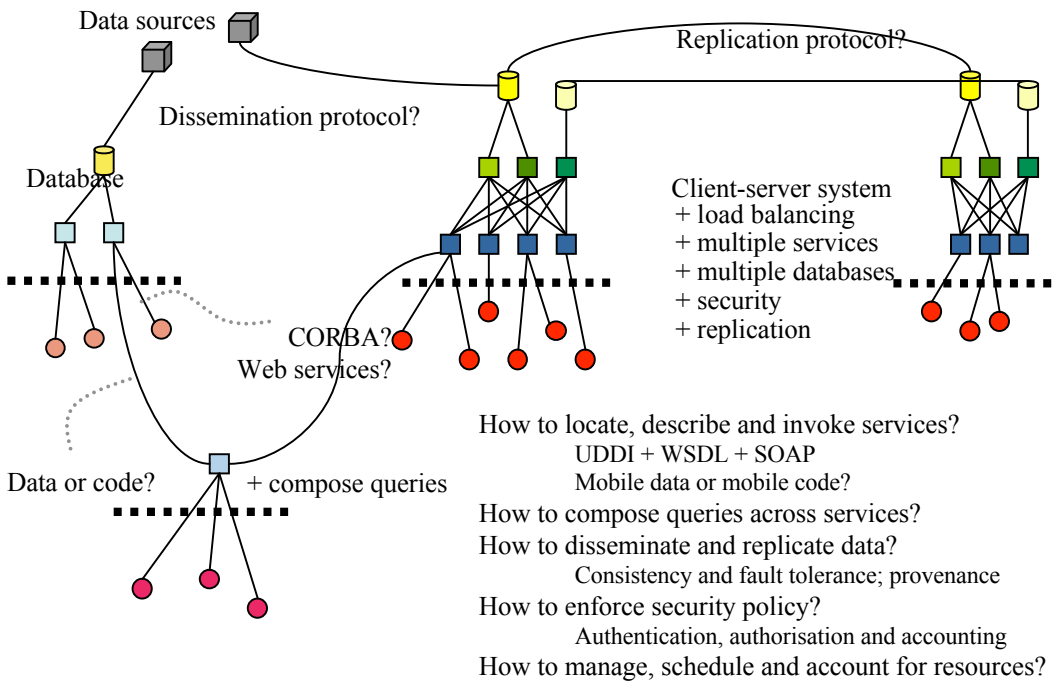


Reference model for Grid Computing differs from the OSI model



Large Scale Systems Architecture

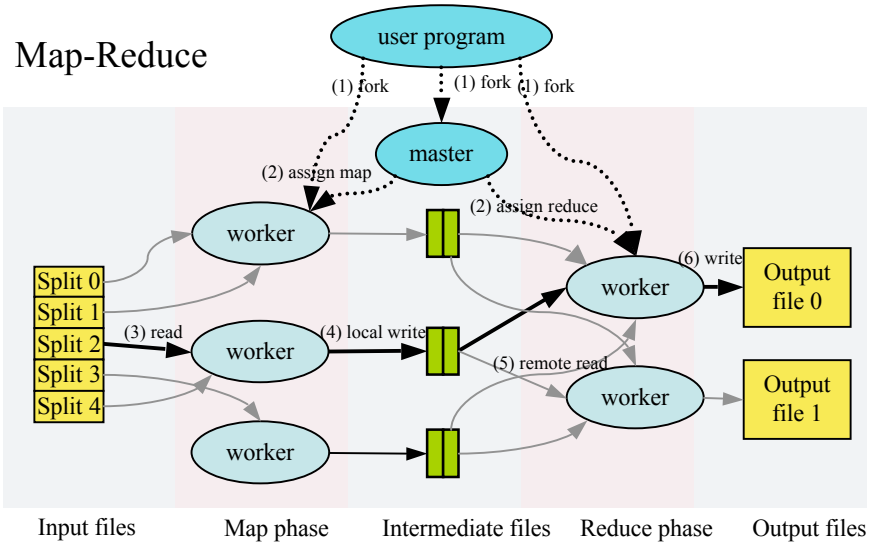
- Key concerns in grids:
  - Autonomy
  - Scalability
  - Heterogeneity
  - Security
  - Fault tolerance

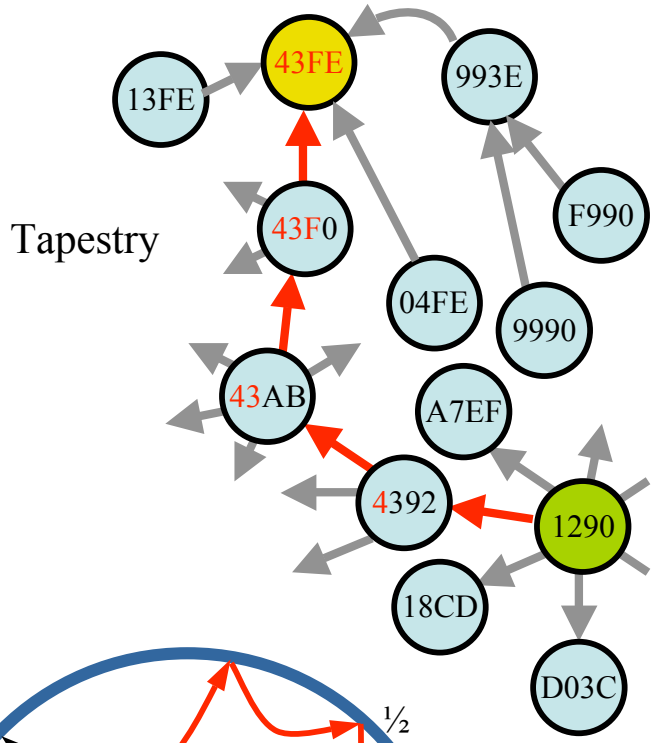


- Traditional client-server model of grid computing

- Challenges and alternative architectures
  - Importance of naming and addressing architectures
  - Discovery and routing
  - Availability, robustness and coupling
  - Timing and ordering
  - Automatic parallelism

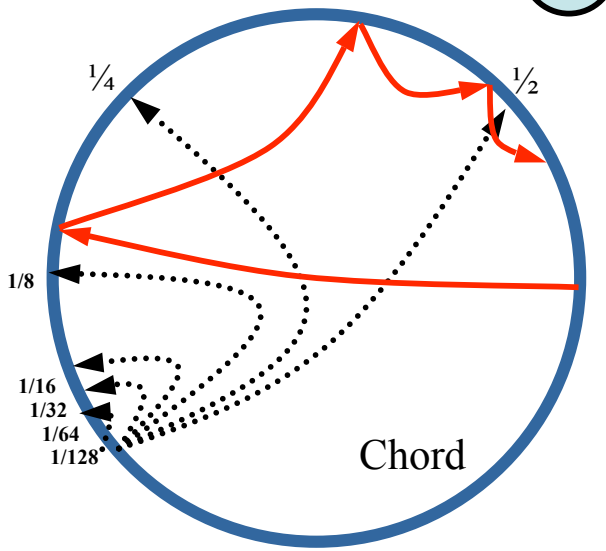
Peer-to-Peer Systems





- Peer-to-peer systems and distributed hash table algorithms
  - Deployment considerations
- New approaches to scalability, naming and object location

- OceanStore



Security

- Virtual organisations and the importance of grid security
  - Authentication, authorisation and accounting
  - Confidentiality, privacy
  - Integrity
  - Fabric management
  - Trust

- Public key infrastructure
  - Symmetric vs asymmetric cryptography; RSA
  - X.509 PKI model
- Policies and role based authorisation
  - Limitations of Globus security: Grid map file
  - Newer solutions e.g. Permis, SAML, Shibboleth

Resource Management: Principles

- Taxonomy and Survey of Grid Resource Management paper
  - Overall aims:
    - Scalability, responsiveness, fault-tolerance, stability
    - Site autonomy and security
  - Overall system architecture:
    - How to organise machines? Flat, hierarchical, cells
    - How to describe resources or operations on them?
    - How to name resources? Organize those names?
    - How to store and discover resource information?
    - How to schedule requests? Allocate jobs to resources?

- Examples:
  - WS-GRAM
  - Community Scheduler Framework

Resource Management: Scheduling

- Job scheduling and management
  - Outline technical problems
  - Effects of autonomy
- Load balancing
  - Resource provision and advertisement
  - Static and dynamic load balancing
  - Queuing models for job submission
  - Heavy tailed distributions
  - Job migration
  - Execution environments
  - Examples
- Workflow management
  - Survey + examples
- Static load balancing fundamentally a solved problem
  - Well developed theory and models
  - Many robust implementations
- Dynamic load balancing difficult
  - Basic problems solved; open issues remain; few implementations
- Grid computing systems currently address limited scope problems
  - No need for complex solutions
  - Likely to change in future: growth of parallelism

Resource Management: Fault Tolerance

- Reliability of a computational grid
- Failure modes
- Fault tolerance
  - Checkpoint and retry
  - Avoiding systematic software failures
    - Error detection
    - Damage confinement
    - Techniques for error recovery
  - Implications of scale and autonomy
  - Economics
- Other failure modes
  - Multi-organisational security
  - Reliable multicast data distribution

- Current grids make widespread use of checkpoint and restart of long-running computational jobs
  - Protection from hardware failures
  - Minimal programmer effort

- More advanced techniques currently not widely used
  - Design diversity and security issues will become important to reliability as grids are used for more critical tasks
  - Reliable multicast protocols will be important as grid computing adopts peer-to-peer protocols, multicast

Resource Management: Economics

- Charging models
  - How to bill users?
  - Externalities and indirect costs
  - Cost and price

- Congestion charging
  - Theory of shadow pricing
  - Pricing multiple resources
    - Vickrey auction
  - Resource negotiation

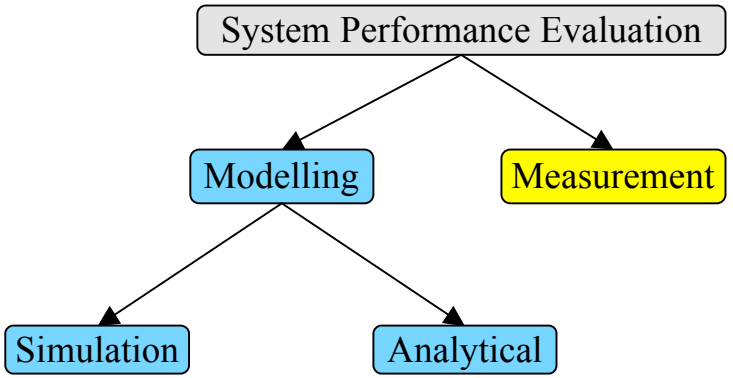
Scalability and Heterogeneity

- What is scale? Why should I care?
- Four principal scale dimensions of interest:
  - Number of nodes
  - Size of distributed database
  - Size of objects
  - Communication delays
- General approaches to scaling
  - Move functionality into "leaves"
  - Partition problem into domains and introduce gateways
  - Caching
  - Asymptotic convergence
  - Smooth out control response
  - Paradigm shift
- Case studies:
  - DNS
    - Simple, clean, lightweight, relaxed
    - Partitioning of database; caching
  - FAST TCP
    - Problems due to large bandwidth-delay product networks
    - Limits of TCP congestion control
    - Problems with Grid-FTP
    - A better TCP: Vegas, FAST, etc.
  - Delay tolerant networking
    - Violate assumptions behind TCP
    - Paradigm shift: store-and-forward message switching; thin waist shifts from IP to Bundle Layer
    - Naming and addressing
    - Sensor networks and remote areas

Systems Modelling and Simulation

- Modelling
  - Disruptive, dangerous, expensive, impractical, ad-hoc, etc.

    ...but essential!
- Simulation
  - E.g. ns2
  - Algorithmic abstraction of system
  - Discrete event simulation: pitfalls
- Analytical modelling
  - Mathematical model of system
  - Trivial Markov modelling example



- Measurements
  - Collection: passive, active, in-line
  - Analysis: on-line, off-line
  - Examples: repositories, p2pmon

Review of Tutorials

- The Globus toolkit (GT4)
- Grid services
  - Servers, WSDL, etc.
  - Client implementation using SOAP
- Globus security
  - X.509 certificates
  - Grid map file
  - GSI and the job start-up process
- Permis
  - Globus security call-out in SAML
  - Permis role-based access control
  - XML security policies
- Condor
  - Job advertisement
  - Job submission
  - Batch queuing system
- Grid portals
  - Portlets, etc.
  - GridSphere
- OGSA-DAI
  - XML-based queries across multiple databases
  - XSLT presentation transforms
- Example applications

Questions or Comments on Review?

Future Directions in Grid Computing?

- Vast and ongoing investment in computational grids
- Technology developing rapidly
  - Versions of Globus, Permis, Condor, etc., used ⇒ obsolete within months

- Two trends we can expect to see:
  - Ongoing convergence of grid services and web services
    - Globus and similar grid toolkits vanish; replaced by generic commercial(?) web services implementations
      - Fundamental concepts remain very similar
    - Grid security might live on – has somewhat specialist needs – or might be subsumed into industry standard security
      - Likely depends on deployment of thin-clients
  - Increasing use of peer-to-peer systems, sensor and delay tolerant networks
    - Heavy push in the computer science research community
    - Vastly more interesting grids can be developed; needs evangelism

Discussion on Future Directions?

The End...

- Tutorial on Friday: demonstration of programming assignments
- Exam format:
  - Answer 1 mandatory wide ranging question; 2 out of 4 optional and more narrowly focussed questions
  - Past papers on the web site
- No revision lecture
  - Email questions to appropriate lecturer
- Please return module evaluation forms!