Grid InterNetworking

Michael Welzl  http://www.welzl.at

DPS NSG Team  http://dps.uibk.ac.at/nsg
Institute of Computer Science
University of Innsbruck, Austria

CAIA guest talk
Swinburne University
Melbourne AUS
12 February, 2008
Outline

- Problem scope (Grid introduction)

- Grid InterNetworking
  - Identifying a research gap
  - Some issues and proposed solutions

- Conclusion
  - Practical problems
  - Future perspectives
  - Final words
Problem scope

Shrinking the problem space
Introducing the Grid

- History: parallel processing at a growing scale
  - Parallel CPU architectures
  - Multiprocessor machines
  - Clusters
  - (“Massively Distributed“) computers on the Internet

- GRID
  - logical consequence of HPC
  - metaphor: power grid
    just plug in, don’t care where (processing) power comes from, don’t care how it reaches you

- Common definition:
The real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi institutional virtual organizations

Scope

- Definition quite broad ("resource sharing")
  - Reasonable - e.g., computers also have harddisks
  - But also led to some confusion - e.g., new research areas / buzzwords: Wireless Grid, Data Grid, Semantic / Knowledge Grid, Pervasive Grid,
    [this space reserved for your favorite research area] Grid

- Example of confusion due to broad Grid interpretation:

  "One of the first applications of Grid technologies will be in remote training and education. Imagine the productivity gains if we had routine access to virtual lecture rooms! (..) What if we were able to walk up to a local ‘power wall‘ and give a lecture fully electronically in a virtual environment with interactive Web materials to an audience gathered from around the country - and then simply walk back to the office instead of going back to a hotel or an airplane?"


⇒ Clear, narrower scope is advisable for thinking/talking about the Grid

- Traditional goal: processing power
  - Grid people = parallel people; thus, main goal has not changed much
The next Web?

- Ways of looking at the Internet
  - Communication medium (email)
  - Truly large kiosk (web)

- The Grid way of looking at the Internet
  - Infrastructure for Virtual Teams

- Most of the time...
  - the “real and specific goal“ is High Performance Computing
  - Virtual Organizations and Virtual Teams are well defined
    i.e. not an „open“ system, e.g. security is a big issue

- Virtual Teams
  - Geographically distributed
  - Organizationally distributed
  - Yet work on a common problem

But Web 2.0 is already here :-)
Virtual Organizations and Virtual Teams

- Distributed resources and people
- Linked by networks, crossing admin domains
- Sharing resources, common goals
- Dynamic

Source: Globus presentation by Ian Foster
The Grid and P2P systems

- Look quite similar
  - Goal in both cases: resource sharing

- Major difference: clearly defined VOs / VTs
  - No incentive considerations
  - Availability not such a big problem as in P2P case
    - It is an issue, but at larger time scales
      - (e.g. computers in student labs should be available after 22:00, but are sometimes shut down by tutors)
    - Scalability not such a big issue as in P2P case
      - …so far! ⇒ convergence as Grids grow

- coordinated resource sharing and problem solving in dynamic, multi institutional virtual organizations (Grid, P2P)
Austrian Grid E-science Grid applications

- Medical Sciences
  - Distributed Heart Simulation
  - Virtual Lung Biopsy
  - Virtual Eye Surgery
  - Medical Multimedia Data Management and Distribution
  - Virtual Arterial Tree Tomography and Morphometry

- High-Energy Physics
  - CERN experiment analyses

- Applied Numerical Simulation
  - Distributed Scientific Computing: Advanced Computational Methods in Life Science
  - Computational Engineering
  - High Dimensional Improper Integration Procedures

- Astrophysical Simulations and Solar Observations
  - Astrophysical Simulations
  - Hydrodynamic Simulations
  - Federation of Distributed Archives of Solar Observation

- Meteorological Simulations

- Environmental GRID Applications
Example: CERN Large Hadron Collider

- Largest machine built by humans: particle accelerator and collider with a circumference of 27 kilometers

- Will generate 10 Petabytes ($10^7$ Gigabytes) of information per year ... starting 2007 (?)

- This information must be processed and stored somewhere

- Beyond the scope of a single institution to manage this problem
  - Projects: LCG (LHC Computing Grid), EGEE (Enabling Grids for E-sciencE)

Source: Globus presentation by Ian Foster
Complexity

- Grid poses difficult problems
  - Heterogeneity and dynamicity of resources
  - Secure access to resources with different users in various roles, belonging to VTs which belong to VOs
  - Efficient assignment of data and tasks to machines (“scheduling“)
Grid requirements

- Computer scientists can tackle these problems
  - Grid application users and programmers are often not computer scientists

- Important goal: ease of use
  - Programmer should not worry (too much) about the Grid
  - User should worry even less
  - Ultimate goal: write and use an application as if using a single computer
    (power grid metaphor)

- How do computer scientists simplify?

- Abstraction.

- We build layers.

- In a Grid, we typically have Middleware.
Toolkits

- **Most famous: Globus Toolkit**
  - Evolution from GT2 via GT3 to GT4 influenced the whole Grid community
  - Reference implementation of Open Grid Forum (OGF) standards

- **Other well-known examples**
  - **Condor**
    - Exists since mid-1980's
    - No Grid back then - system gradually evolved towards it
    - Traditional goal: harvest CPU power of normal user workstations
      ⇒ many Grid issues always had to be addressed anyway
    - Special interfaces now enable Condor-Globus communication ("Condor-G")
  - **Unicore** (used in D-Grid)
  - **gLite** (used in EGEE)

- **Issues that these middlewares (should) address**
  - Load Balancing, error management
  - Authentication, Authorization and Accounting (AAA)
  - Resource discovery, naming
  - Resource access and monitoring
  - Resource reservation and QoS management
Evolution: moving towards an architecture

- **OGSI / OGSA: Open Grid Service Infrastructure / Architecture**
  - Open Grid Forum (OGF) standards
  - OGSA = service-oriented architecture; key concept for virtualization
    use a resource = call a service
  - OGSI = Web Services + state management
    • failed: too complex, not compliant with Web Service standards

Source: Globus presentation by Ian Foster
Current SoA

- Standards are only specified when mechanisms are known to work
  - Globus only includes such working elements

- Lots of important features missing

- Practical issues with existing middlewares
  - Submitting a Globus job is very slow (Austrian Grid: approx. 20 seconds)
    ⇒ significant granularity limit for parallelization!
  - Globus is a huge piece of software

- Currently, some confusion about right location of features
  - On top of middleware? (research on top of Globus)
  - In middleware? (other Middleware projects)
  - In the OS? (XtreemOS)

⇒ Upcoming slides concern mechanisms which are mostly on top and partially within middleware
Automatic parallelization in Grids

- **Scheduling**: important issue for “power outlet“ goal!
  - Automatic distribution of tasks and inter-task data transmissions = scheduling

- Grid scheduling encompasses
  - **Resource Discovery**
    - Authorization Filtering, Application Requirement Definition, Minimal Requirement Filtering
  - **System Selection**
    - Dynamic Information Gathering
    - System Selection
  - **Job Execution**
    - (optional) Advance Reservation
    - Job Submission
    - Preparation Tasks
    - Monitoring Progress
    - Job Completion
    - Clean-up Tasks

- So far, most scheduling efforts consider embarassingly parallel applications - typically parameter sweeps *(no dependencies)*
Grid workflow applications

- **Dependencies** between applications (or large parts of applications) typically specified in **Directed Acyclic Graph (DAG)**
  - Condor: DAG manager (DAGMan) uses .dag file for simple dependencies
  - “Do not run job ‘B’ until job ‘A’ has completed successfully”
  - DAGMan scheduling: for all tasks do...
    - Find task with earliest starting time
    - Allocate it to processor with Earliest Finish Time
    - Remove task from list

- **GriPhyN (Grid Physics Network)** facilitates workflow design with **“Pegasus“** (Planning for Execution in Grids) framework
  - Specification of abstract workflow: identify application components, formulate workflow specifying the execution order, using **logical names** for components and files
  - Automatic generation of concrete workflow (map components to resources)
  - Concrete workflow submitted to **Condor-G/DAGMan**
Grid Workflow Applications /2

- Components are built, Web (Grid) Services are defined, Activities are specified
- Several projects (e.g. K-WF Grid) and systems (e.g. ASKALON) exist
- Most applications have simple workflows
  - E.g. Montage: dissects space image, distributes processing, merges results
Grid InterNetworking
Research gap: Grid-specific network enhancements

Original Internet technology

Enriched with customised network mechanisms

Bringing the Grid to its full potential!

Today‘s Grid applications

EC-GIN

EC-GIN enabled Grid applications

Driving a racing car on a public road

Applications with special network properties and requirements

Traditional Internet applications (web browser, ftp, ..)

Real-time multimedia applications (VoIP, video conference, ..)
Grid-network peculiarities

• Special behavior
  - Predictable traffic pattern - this is totally new to the Internet!
    - Web: users create traffic
    - FTP download: starts ... ends
    - Streaming video: either CBR or depends on content! (head movement, ..)
  • Could be exploited by congestion control mechanisms
    - Distinction: Bulk data transfer (e.g. GridFTP) vs. control messages (e.g. SOAP)
    - File transfers are often “pushed“ and not “pulled“
    - Distributed System which is active for a while
      • overlay based network enhancements possible
        - Multicast
        - P2P paradigm: “do work for others for the sake of enhancing the whole system (in your own interest)“ can be applied - e.g. act as a PEP, ...
  • sophisticated network measurements possible
    - can exploit longevity and distributed infrastructure

• Special requirements
  - file transfer delay predictions
    • note: useless without knowing about shared bottlenecks
  - QoS, but for file transfers only (“advance reservation“)
What is EC-GIN?

- European project: *Europe-China Grid InterNetworking*
  - STREP in IST FP6 Call 6
  - 2.2 MEuro, 11 partners (7 Europe + 4 China)
  - Networkers developing mechanisms for Grids

<table>
<thead>
<tr>
<th>Anticipated Results &amp; Impacts</th>
<th>Milestone (date)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network level analysis of Grid applications</td>
<td>M15</td>
</tr>
<tr>
<td><em>Increased Grid knowledge in network community</em></td>
<td></td>
</tr>
<tr>
<td>ns-2 code for Grid network simulation</td>
<td>M24</td>
</tr>
<tr>
<td><em>Fostering EC-GIN related research</em></td>
<td></td>
</tr>
<tr>
<td>Grid-specific network enhancements</td>
<td>M30</td>
</tr>
<tr>
<td><em>Snowball effect in research community</em></td>
<td></td>
</tr>
<tr>
<td>Economic Grid traffic mgt and security mechanisms</td>
<td>M30</td>
</tr>
<tr>
<td><em>Reinforcing commercial use of Grids</em></td>
<td></td>
</tr>
<tr>
<td>Customised GINTONIC implementations</td>
<td>M31</td>
</tr>
<tr>
<td><em>Performance improvement of partner Grids</em></td>
<td></td>
</tr>
</tbody>
</table>
Research Challenges

- Research Challenges:
  - How to model Grid traffic?
    - Much is known about web traffic (e.g. self-similarity) - but the Grid is different!

- How to simulate a Grid-network?
  - Necessary for checking various environment conditions
  - May require traffic model (above)
  - Currently, Grid-Sim / Net-Sim are two separate worlds
    (different goals, assumptions, tools, people)

- How to specify network requirements?
  - Explicit or implicit, guaranteed or “elastic“, various possible levels of granularity

- How to align network and Grid economics?
  - Combined usage based pricing for various resources including the network

- What P2P methods are suitable for the Grid?
  - What is the right means for storing short-lived performance data?
Open issue: abstract-concrete WF mapping

Tasks = \{T1, T2, T3, T4\}
Resources = \{R1, R2, R3, R4\}
Data transfers = \{D1, D2, D3, D4\}

Unnoticed by scheduling algorithms!
Large stacks

1. Physical Layer
2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

Source: http://img.dell.com/images/global/topics/power/ps1q02-broadcom.gif
Open issue: layering inefficiency

- Stateless
- Connection state
- Stateless
- HTTP 1.0
- TCP
- Stateless
- HTTP 1.1

Stateful
Grid Service
Web Service
SOAP
WS- RF

Breaking the chain

Connection state

Can’t reuse connections!

Could reuse connections, but doesn’t!

Reuses connections

Can’t reuse connections!
NWS: The Network Weather Service

- Most common tool for performance prediction
  - Important for making good scheduling decisions

- Distributed system consisting of
  - Name Server (boring)
  - Sensor - actual measurement instance, regularly stores values in...
  - Persistent State
  - Forecaster (calculations based on data in Persistent State)

- Interesting parts:
  - Sensor
    - Measured resources: availableCpu, bandwidthTcp, connectTimeTcp, currentCpu, freeDisk, freeMemory, latencyTcp
  - Forecaster
    - Apply different models for prediction, compare with actual measurement data, choose best match
NWS critique

• **Architecture** (splitting into sensors, forecaster etc.) seems reasonable; open source ⇒ consider integrating new work in NWS

• **Sensor**
  - active measurements even though non-intrusiveness was an important design goal
  - does not passively monitor TCP (i.e. ignores available data)
  - **strange methodology:**
    - (Large message throughput) *Empirically, we have observed that a message size of 64K bytes (...) yields meaningful results*“
  - ignores packet size (= measurement granularity) and path characteristics
  - trivial method - much more sophisticated methods available
  - point-to-point measurements: distributed infrastructure not taken into account

• **Forecaster**
  - relies on these weird measurements, where we don’t know much about the distribution (but we do know some things about net traffic IFF properly measured)
  - uses quite trivial models (but they may in fact suffice...)
NWS measurements (Austrian Grid)


- Salzburg-Linz (left): more than 20 MB needed to saturate link
- Within Innsbruck (right), Gigabit link: around 100 MB needed

- NWS supposedly designed to be non-intrusive...
The impact of shared bottlenecks

- Example problem:
  - C allocates tasks to A and B (CPU, memory available); both send results to C
  - B hinders A - task of B should have been kept at C!
- Path changes are rare - thus, possible to detect potential problem in advance
  - generate test messages from A, B to C - identify signature from B in A's traffic
- Another issue in this scenario: how valid is a prediction that A obtains if a measurement / prediction system does not know about the shared bottleneck?
EC-GIN Large File Transfer Scenario (LFTS)

Multipath file transfer
(A→B + A→C→B) beneficial

Multipath file transfer not beneficial
due to shared bottleneck

Questions: when does this make sense, how to expose this functionality, how to authenticate and authorize...?
Shared bottleneck detection with SVD

Muhammad Murtaza Yousaf, Michael Welzl, Bulent Yener (2007); under submission

- **Input:** end-to-end forward delays of multiple flows
- **Analysis**
  - Multivariate Analysis Method SVD (Singular Value Decomposition)
    - Matrix operation which yields clustered values for correlating flows
    - Calculate differences between values, consider changes between clusters as outliers, apply simple outlier detection method
- **Output:** clusters of flows which share a bottleneck
  - Very precise, easy to calculate, can cluster multiple flows at the same time (other work uses pairwise cross correlation)
Extending the Padhye equation to \( N \) flows


- Fair amount of work done, but so far, no (easily usable) approximation exists which also takes loss into account

- Useful in a Grid for multiple reasons:
  - Prediction of GridFTP throughput (multiple TCP flows)
  - Protocol with tunable aggression (MulTFRC)
    - Because, if a Grid application uses two flows which share a bottleneck, flow 1 may be 3.7 times as important to it as flow 2 (e.g. if flow 2 is from replication)
    - For fairness: if we take a break, we may earn “aggression points“

\[
\]

\[
\text{we get } E[W] \text{ depending on } n, p \text{ and } b
\]
Other current EC-GiN work

- **INRIA + UIBK** working on scheduling of advance reservations for bulk data transfers *(using high-speed congestion control mechanisms)*

- **WP2** dedicated to modeling (and ns-2 simulation code)
  - Lead: ULANC; currently collecting measurements from everywhere...
  - Traffic model from INRIA
  - UIBK developed a Grid-Net scheduling simulator using ns-2

- **ISCAS** developed a high-speed SOAP engine

- **UniZH** working on P2P incentive mechanisms for the Grid + security

- ...stay tuned for more!
Conclusion

- Practical problems
- Future perspectives
- Final words
Problem: How Grid people see the Internet

- **Abstraction** - simply use what is available
  - still: performance = main goal

- **Existing transport system**
  (TCP/IP + Routing + ..) works well

- **QoS makes things better, the Grid needs it!**
  - we now have a chance for that, thanks to IPv6

- **Quote from a paper review:**
  "In fact, any solution that requires changing the TCP/IP protocol stack is practically unapplicable to real-world scenarios, (...)"

- **How to change this view**
  - Create awareness - e.g. GGF GHPN-RG published documents such as "net issues with grids", "overview of transport protocols"
  - Develop solutions and publish them! (EC-GIN, GridNets)
A time-to-market issue

Typical Grid project

Research begins

Real-life coding begins

Thesis writing

Real-life tests begin

Result: thesis + running code; tests in collaboration with different research areas

Typical Network project

Research begins

(Simulation) coding begins

Thesis writing

Ideal

Result: thesis + simulation code; perhaps early real-life prototype (if students did well)
Machine-only communication

- Trend in networks: from support of Human-Human Communication
  - email, chat

- via Human-Machine Communication
  - web surfing, file downloads (P2P systems), streaming media

- to Machine-machine Communication
  - Growing number of commercial web service based applications
  - New “hype“ technologies: Sensor nets, Autonomic Computing vision

- Semantic Web (Services): first big step for supporting machine-only communication at a high level

- **So far, no steps at a lower level**
  - This would be like RTP, RTCP, SIP, DCCP, … for multimedia apps: not absolutely necessary, but advantageous
The long-term value of Grid-net research

- A subset of Grid-net developments will be useful for other machine-only communication systems!

- Key for achieving this: change viewpoint from “what can we do for the Grid“ to “what can the Grid do for us“ (or from “what does the Grid need“ to “what does the Grid mean to us“)
**Conclusion**

- Grid applications show special requirements and properties from a network perspective
  - and it is reasonable to develop tailored *Internet* technology for them.

- There is another class of such applications...

- **Multimedia.**

- For multimedia applications, an immense number of network enhancements (even IETF standards) exist.

- For the Grid, there is nothing.

- **This is a research gap; let's fill it together!**
  - submit a paper to GridNets 2008 in Beijing! :-(

**Reminder:** if done right, such research is also applicable to other systems with machine-only traffic
More information: http://www.ec-gin.eu

Thank you!

Questions?