Designing job-, resource-, and project-management tools for the Grid ecosystem

Erik Elmroth
Umeå University, Sweden

PPAM 2007
Gdansk, Poland
September 10-12, 2007

The Grid interoperability contradiction

- Grids address resource interoperability and resource heterogeneity
- Contradictory, we have currently major
  - interoperability problems
  - portability problems

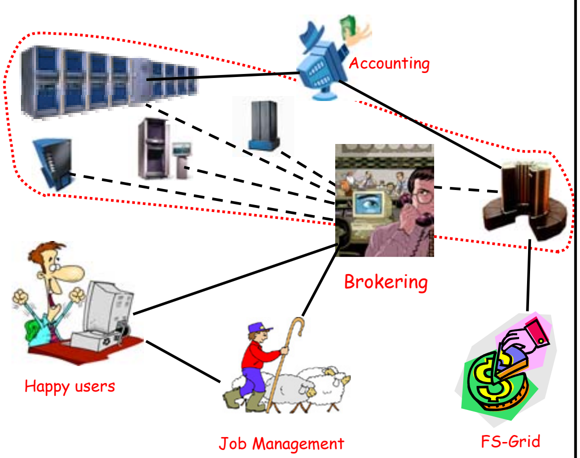
  between different Grids
- ... and the consequences...
  - Constantly reinventing the wheels
  - Slow progress in development of fundamental Grid services
  - Development of portable high-level Grid applications virtually impossible

Eco- vs. Ego-systems

- Healthy Grid *eco*system:
  - collection of generic components (for different niches) developed by the Grid community
  - competition, innovation, evolution, and diversity lead to natural selection
- Unhealthy Grid *ego*system:
  - trying to occupy too many niches with a single component
  - too strong coupling between components
  - probably caused by
    - a need for rapidly developed infrastructure
    - a symbiosis of "lack of accepted standards" and a strong "not invented here" mentality

GIRD – Grid Infrastructure Research & Development at Umeå University, Sweden www.gird.se

- Generic infrastructure components for resource & project management
- Interoperable, standards-based
- Focusing both business and e-Science



The GIRD approach to Grid computing

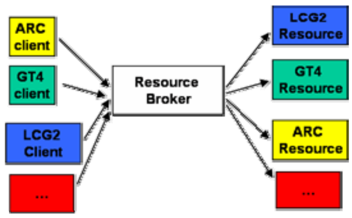
- Small, well-defined, single-purpose components
  - Each occupy a single niche in the Grid ecosystem
- Focus on interoperability
  - Use (emerging) standard
    - Formats
      - For intercomponent interactions
      - Internally
    - Interfaces
    - Functionality
  - Ease of integration with existing middleware
    - Few, small and well-defined integration points
- Service-oriented architectures and good programming practices (e.g., minimize software dependencies, define mechanisms rather than behavior/policies, re-use instead of re-invent, customizability, simplicity, interface abstraction, etc)

Erik Elmroth, Umeå University, Sweden http://www.gird.se

Sample ongoing projects

Generic Grid Computing Research

1. SweGrid Accounting System (SGAS)
   - Included in Globus Toolkit 4
2. Grid-wide fairshare scheduling
   - Hierarchical three-party QoS support (user, resource-owner, VO-authority)
3. Job submission and resource brokering
   - Standards-based, cross-middleware (ARC, LCG2, GT4)
4. Multi-tier job management framework
   - High-level job management
5. Generic Grid workflows
   - Extends on work with Univ. Birmingham, Alabama
6. Resource and project portal
   - Jointly by HPC2N, PDC, and NSC
7. Grid interface-generation for numerical software libraries
   - SLICOT-interfaces for NetSolve and web-portals



Erik Elmroth, Umeå University, Sweden http://www.gird.se

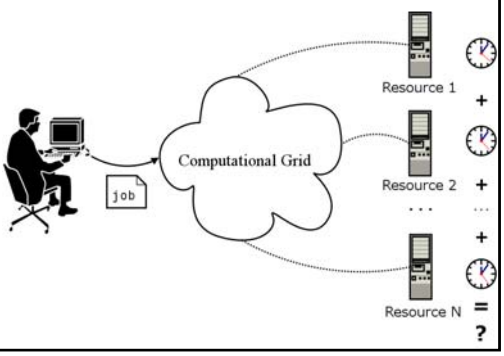
1. Enforcing resource allocations with the SweGrid Accounting System (SGAS)

Erik Elmroth & Peter Gardfjäll, UmU
Lennart Johnsson, Olle Mulmo &
Thomas Sandholm, KTH

Grid accounting - tracking Grid resource usage

Maintaining a (consistent) Grid-wide view of the Grid resources utilized by VO members

- Measure and control users' total resource usage on the Grid
  - Assuming absence of central point of control
  - Resource owners should retain local control

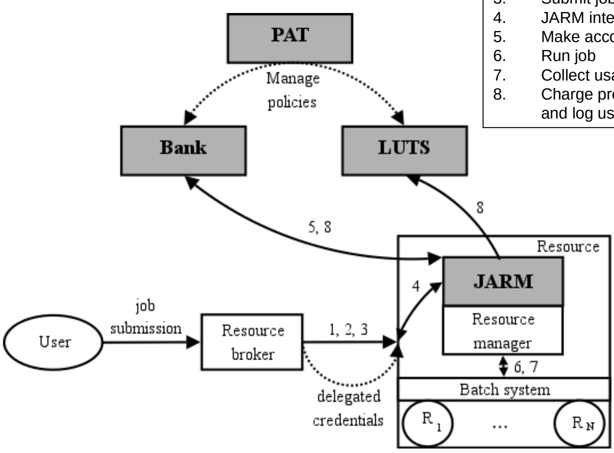


SweGrid Accounting System (SGAS)

- Decentralized resource allocation enforcement system
- SGAS performs soft real-time enforcement of allocations
  - *Real-time enforcement*: Resources can deny access if project quota has been used up
  - *Soft*: Enforcement is subject to local resource policies
    - Strict enforcement not always appropriate
- WSRF-compliant implementation using Globus Toolkit 4 Java WS core
- Developed with an emphasis on easy integration into different Grid middleware
  - Single-point-of-integration
  - In SweGrid: deployed on top of ARC middleware
  - Globus Toolkit 4 WS-GRAM is now prepared for SGAS support

SGAS component interactions

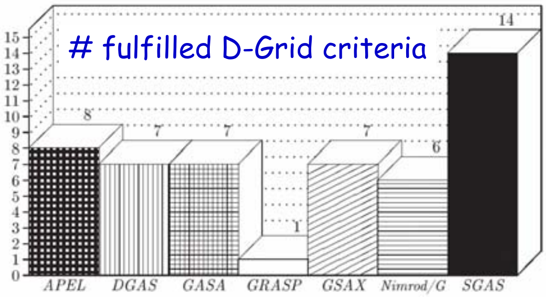
1. Contact resource
2. Authenticate/authorize (delegate credentials)
3. Submit job request
4. JARM intercepts request
5. Make account reservation
6. Run job
7. Collect usage info
8. Charge project account and log usage info



①

SGAS available for production use

- Software availability (v 2.0):
  - Download at www.sgas.se
  - Included in GT4
    - Tech preview
- In use in SweGrid & NDGF
- Considered by other Grids



- "Test winner" in German D-Grid investigation, October 2006 (compared to APEL, DGAS, GASA, GRASP, GSAX, Nimrod/G)

"The four approaches SGAS, GASA, DGAS, APEL inherit the most promising concepts, whereas within these four, there is an advantage for SGAS. SGAS has its special strength in interoperability, ability for integration, portability, accounting beyond one community, supporting standards, security, fault tolerance, precision, administration, and verification" (excerpt from abstract by Rückemann-Müller-von Voigt)

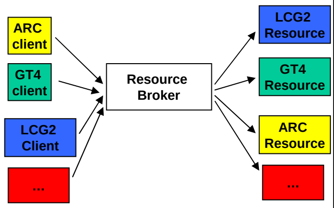
3. An Interoperable, Standards-based Grid Resource Broker and Job Submission Service

Erik Elmroth & Johan Tordsson, UmU
Umeå University, Sweden

③

JSS - An interoperable, standards-based Grid resource broker and job submission service

- Web Service (GT4 WS-Core) based job submission service (JSS) and Grid resource broker
  - Decentralized broker not assuming global control
  - Schedule to minimize either job start time or job completion time
  - Exchangeable modules and resource selection algorithms
- Uses existing and emerging Grid standards (internally and externally)
  - JSDL, GLUE, WSAG, WSRF

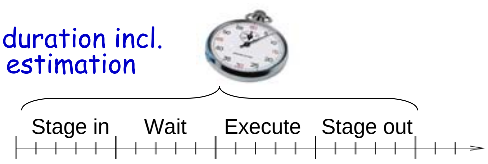


- Interoperable with multiple middlewares
  - Job submission possible to any (supported) middleware, on both client and resource side
  - Cross-middleware submissions
  - Simple integration with additional middlewares
    - Typically, plugins and format converters constitute < 10% of total code

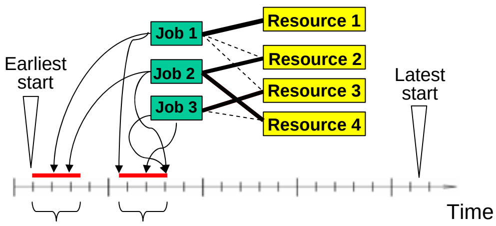
③

JSS brokering functionality

- Features include
  - A priori estimation of job duration incl. benchmark-based runtime estimation



  - Advance reservations
  - Coallocation

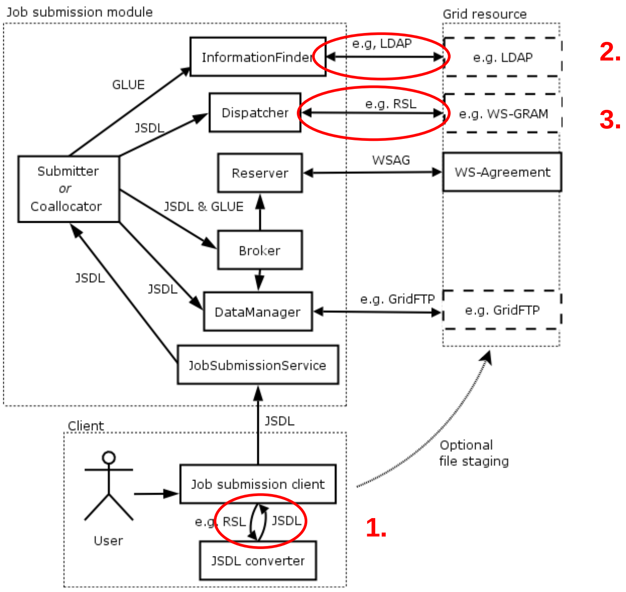


- High performance
  - 250 jobs/min (< 1s response time)

③

JSS architecture

Middleware Integration



4. Grid Job Management Framework

Erik Elmroth, Peter Gardfjäll, Arvid Norberg, Johan Tordsson and P-O

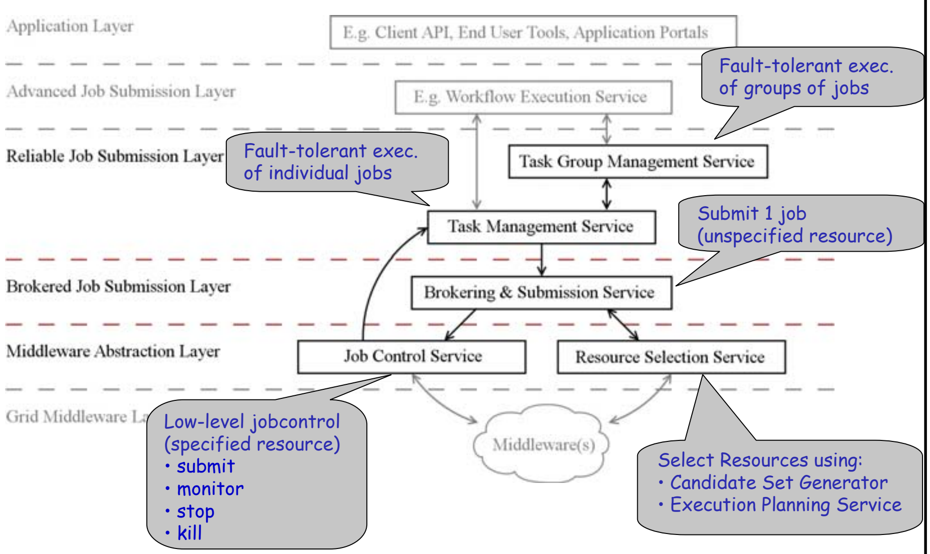
Umeå University, Sweden

④

GJMF - Grid Job Management Framework

- Multi-service framework for Grid job management
  - Flexible & customizable architecture
  - Hierarchical layers of functionality
    - Job control, resource selection, fault tolerant execution, simplified management of groups of jobs
  - Each service add value and can be used individually
- Focus on existing and emerging Grid standards
  - JSDL, WSRF, OGSA-RSS, OGSA-BES
- Low overhead
  - Brokered fault-tolerant submission of job groups: 0.2 s slower per job (compared to GT4 WS-GRAM)

GJMF architecture overview



4. Lightweight Grid workflow execution engine

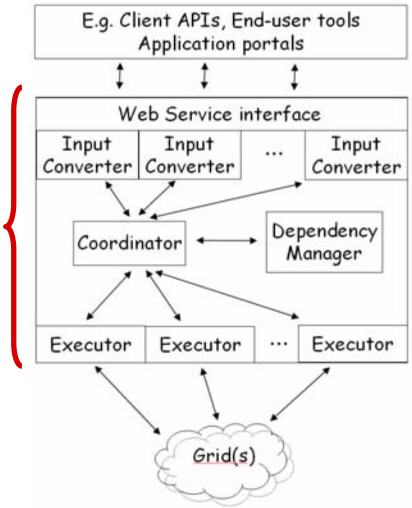
Erik Elmroth, Francisco Hernandez, and Johan Tordsson

Umeå University, Sweden

⑤

Workflow execution engine (cont.)

- Lightweight execution engine
  - Focus on workflow execution *only*
  - Focus on Grids *only*
  - Enables modular design of the next generation workflow tools
- Simple DAG w/fl language internally
- Engine implemented as Web Service
  - WSRF to model workflows
    - State management and monitoring for "free"
  - GT4-based implementation
  - Currently supports Karajan, GT4, ARC, GJMF
- Client prototype recently developed



2. A Decentralized System for Grid-wide Fairshare Scheduling

Erik Elmroth & Peter Gardfjäll, UmU

Umeå University, Sweden

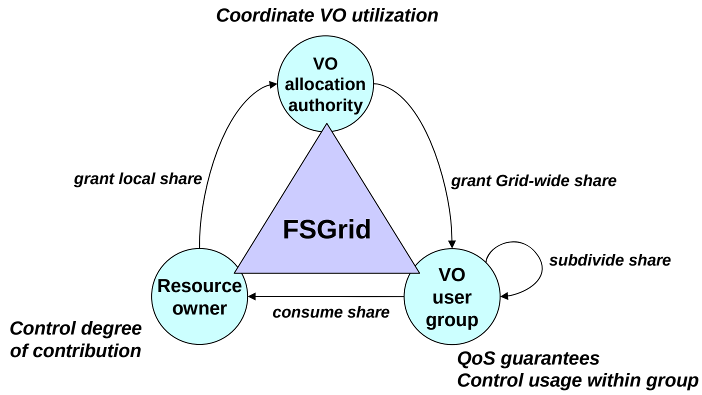
②

FSGrid - motivation

- Coordinating Grid utilization to achieve fairness and efficient use of aggregate capacity
- How can we divide the aggregate computing capacity of a Grid between research groups in a manner that
  - ... is fair and provides QoS guarantees to users
  - ... preserves site-autonomy
  - ... is decentralized
  - ... is simple to deploy in the existing system
- Decentralized Grid-wide fairshare scheduling

②

Resource allocation model – share policies



FairShareGrid system provides support for:

- Resource owners to control the usage of the local resource between different VOs, projects, and users on
- VOs, projects, and users to control the usage of grid-wide allocations among themselves

②

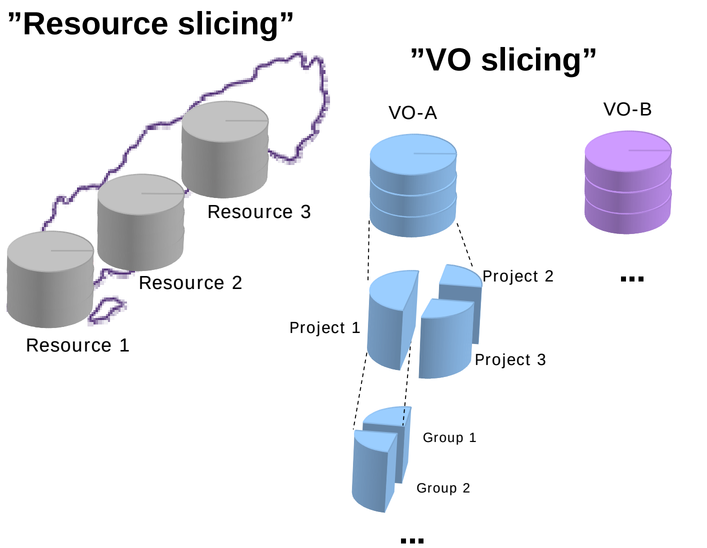
Fairshare scheduling

- A standard-technique used on individual computers since decades
- (Logical) **division of resource capacity**
  - Users granted **target shares**
  - Entitled portion of delivered utilization
- Scheduler adjusts job prio according to past usage
  - **job prio := f(target share, job submitter historical usage)**
  - History decay to increase impact of recent usage
- Goal: fairness over time

We apply fairshare scheduling on a Grid-wide scale

②

"Capacity slicing"



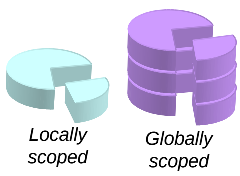
Erik Elmroth, Umeå University, Sweden

http://www.gird.se

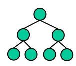
②

Share policy model

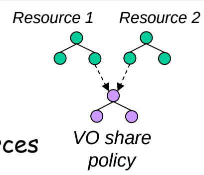
- *Locally scoped* share policies
  - Divides local capacity ("resource slicing")
- *Globally scoped* share policies
  - Divides total VO capacity ("VO slicing")



- Hierarchical policy structure
  - Share tree, recursive subdivided
  - Each node: subshare (in percentage) of parent share



- Supports remote policy references
  - A node may "mount" a remote policy tree
  1. Delegation of subpolicy definition
  2. VO policy distribution and coordination of resources

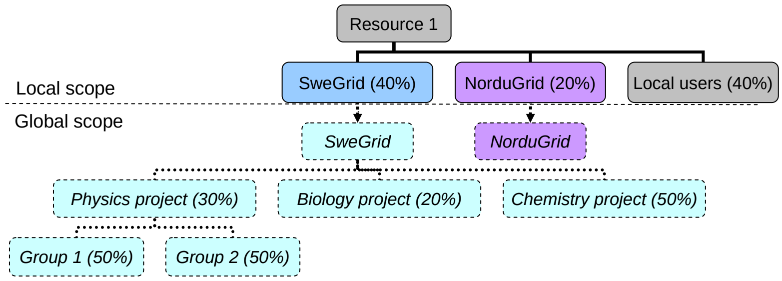


Erik Elmroth, Umeå University, Sweden

http://www.gird.se

②

Share policy illustration

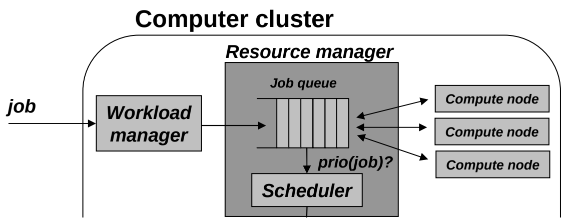


- Share policy enforcement
  - Carried out locally by steering utilization towards target shares
  - Local shares – enforced locally (local usage data)
  - Global shares – collective enforcement (Grid-wide usage data)
  - Top-down enforcement
  - Decentralization! No central coordinator

Erik Elmroth, Umeå University, Sweden http://www.gird.se

②

FSGrid operation context

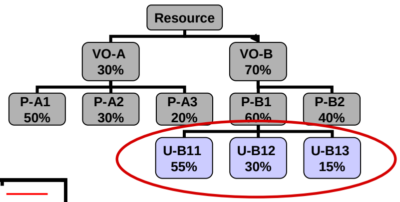


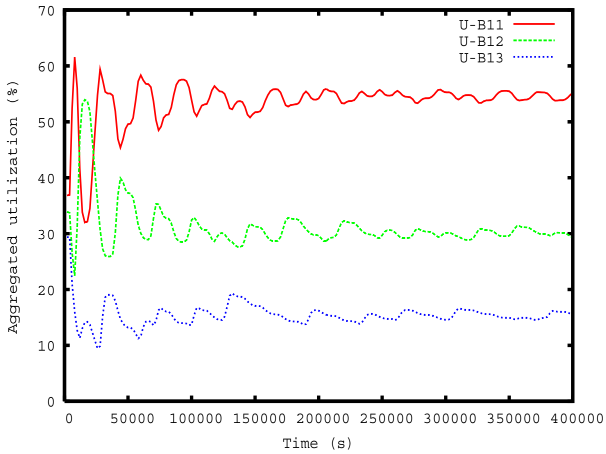
Erik Elmroth, Umeå University, Sweden http://www.gird.se

②

1. Correctness

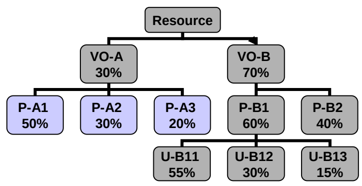
P-B1 usage



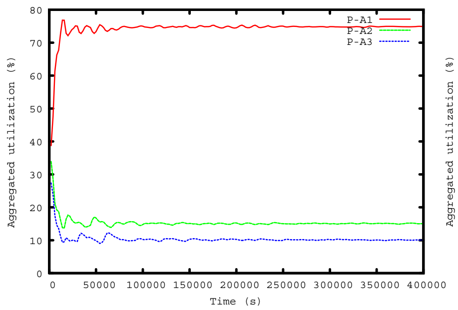


P-B1 users' utilization

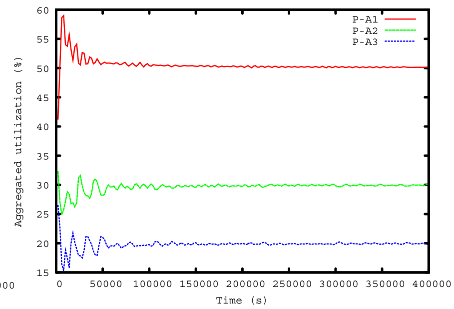
2. Imbalanced workload



Only local usage data



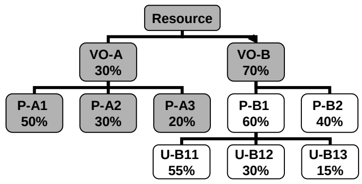
Grid-wide usage data

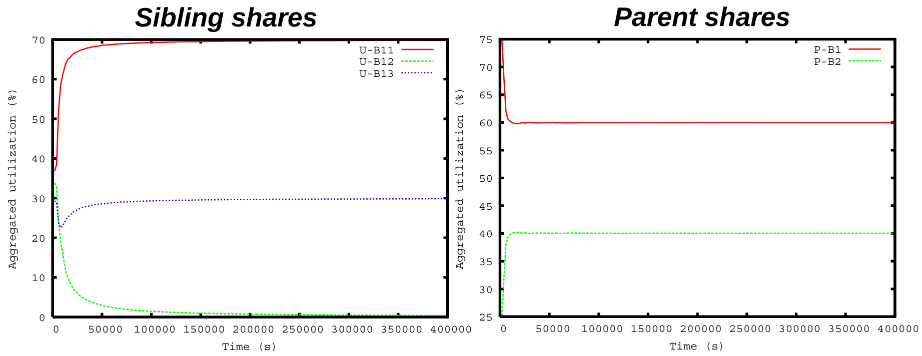


P-A2 and P-A3 only submit jobs to half of the resources

②

3. Subgroup isolation





U-B12 becomes idle

- Performs subgroup isolation
- Idle share made available to (and only to) active sibling entries

②

Summary - FSGrid properties

- Enforces target shares over time
  - In a top-down, least-favored-first manner
  - Local and globally scoped shares
  - Hierarchical share policies, fairness on multiple levels
- Handles imbalanced workload
- Performs subgroup isolation
  - Unused shares are divided over share tree siblings
- Easy integration with prio-based schedulers
  - Shields scheduler from policy details
  - Can control impact on overall scheduling

FairShareGrid system provides support for:
- Resource owners to control the usage of the local resource between different VOs, projects, and users on
- VOs, projects, and users to control the usage of their grid-wide allocations among themselves

The *if, when & where* for Grid jobs

- If: SGAS
- When: FS-Grid
- Where (and how): JSS, GJMF, Workflow engine

Concluding remarks

- Need for reusable and composable components – ecosystem idea
- Our approach proven feasible (and may co-exist with other approaches):
  - Small, well-defined, single-purpose components
  - Leverage standards for improved interoperability and ease of composition of components
  - Middleware integration via very few, small, and well-defined integration points
  - Service-oriented architectures
- Academic and industrial use (e-science & e-business)

Recent Grid computing publications (2005-2007), see www.gird.se

1. E. Elmroth and P. Gardfjäll. Design and Evaluation of a Decentralized System for Grid-wide Fairshare Scheduling. *e-Science 2005. First IEEE Conference on e-Science and Grid Computing*, IEEE Computer Society Press, USA,pp. 221-229, 2005.
2. E. Elmroth, P. Gardfjäll, O. Mulmo, and T. Sandholm. An OGSA-based Bank Service for Grid Accounting Systems. *Lecture Notes in Computer Science*, Vol. 3732, Springer Verlag, pp. 1051-1060, 2006.
3. E. Elmroth, P. Gardfjäll, A. Norberg, J. Tordsson and P-O Östberg. Designing general, composable, and middleware-independent Grid infrastructure tools for multi-tiered job management. In T. Priol and M. Vaneschi (Eds.) Towards Next Generation Grids. Springer Verlag, pp. 175-184, 2007.
4. E. Elmroth, P. Gardfjäll, and J. Tordsson. An Advanced Grid Computing Course for Application and Infrastructure Developers. *CCGrid05*, IEEE Computer Society Press, USA, 2005, pp. 43-50, 2005.
5. E. Elmroth, F. Hernandez, and J. Tordsson. A light-weight Grid workflow execution service enabling client and middleware integration. Proceedings of the Grid Applications and Middleware Workshop, PPAM 2007, Springer Verlag, Lecture Notes in Computer Science (accepted).
6. E. Elmroth, M. Nylén, and R. Oscarsson. A User-Centric Cluster and Grid Computing Portal. *International Journal of Computational Science and Engineering*, 2006, (accepted).
7. E. Elmroth and R. Skelander. Semi-automatic generation of Grid computing interfaces for numerical software libraries. *Lecture Notes in Computer Science*, Vol. 3732, Springer Verlag, pp. 404-412, 2006.
8. E. Elmroth and J. Tordsson. An Interoperable Standards-based Grid Resource Broker and Job Submission Service. *e-Science 2005. First IEEE Conference on e-Science and Grid Computing*, IEEE Computer Society Press, USA, pp. 212-220, 2005.
9. E. Elmroth and J. Tordsson A standards-based Grid resource brokering service supporting advance reservations, coallocation and cross-Grid interoperability. *Submitted for Journal Publication*, November, 2006.
10. E. Elmroth and J. Tordsson. Grid Resource Brokering Algorithms Enabling Advance Reservations and Resource Selection Based on Performance Predictions. *Future Generation Computer Systems. The International Journal of Grid Computing: Theory, Methods and Applications*. Elsevier, (accepted).
11. P. Gardfjäll. Capacity Allocation Mechanisms for Grid Environments. *Licentiate Thesis*, UMINF-06.38, Umeå University, October, 2006.
12. P. Gardfjäll, E. Elmroth, L. Johnsson, O. Mulmo, and T. Sandholm. Scalable Grid-wide Capacity Allocation with the SweGrid Accounting System (SGAS). *Submitted for Journal Publication*, revised, August, 2007.
13. Z. Guan, F. Hernandez, P. Bangalore, J. Gray, A. Skjellum, V. Velusamy, and Y. Liu. Grid-Flow: a Grid-enabled scientific workflow system with a petri-net-based interface. *Concurrency and Computation: Practice and Experience*, 18(10), pp. 1115 - 1140, 2006.
14. F. Hernandez, P. Bangalore, J. Gray, Z. Guan, and K. Reilly. GAUGE: Grid Automation and Generative Environment. *Concurrency and Computation: Practice and Experience*, 18(10), pp. 1293 - 1316, 2006.
15. T. Sandholm, P. Gardfjäll, E. Elmroth, L. Johnsson, and O. Mulmo. A Service-Oriented Approach to Enforce Grid Resource Allocations, *International Journal of Cooperative Information Systems*, Vol. 15, No. 3, pp. 439-459, 2006.
16. J. Tordsson. Decentralized Resource Brokering for Heterogeneous Grid Environments. *Licentiate Thesis*, UMINF-06.39, Umeå University, November, 2006.