Interfacing a Managed Local Fabric to the GRID

LCG Review

Tim Smith IT/FIO

Contents


- **Fabrics, GRIDs and the interface**

- **Interfacing**
  - User Management
  - Security
  - Worker Nodes
  - SW distribution
  - Gateway Nodes
- **Milestones**

# Converging Development Fronts

- **Tier-0 Fabric Automation**
  - Production
    - Software and Configuration Managers LXBATCH
  - Development/Prototypes
    - State and HW management systems, Monitoring, Fault tolerance

- **LCG-0, LCG-1,... LCG-2**
  - Deploying entire SW stack (including fabric!)
  - GRID Services and Operations

- **Fabric – GRID Integration**
  - Establishing the boundaries and interfaces

GRID Services Architecture

The Dream

Reality

GRID Application Layer

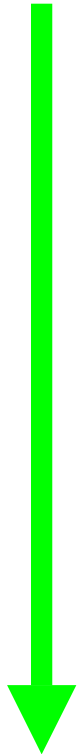Collective Services

Underlying GRID Services

Fabric Services
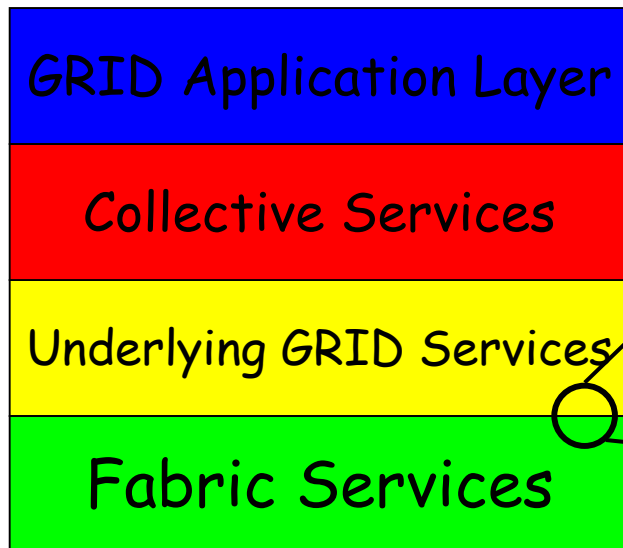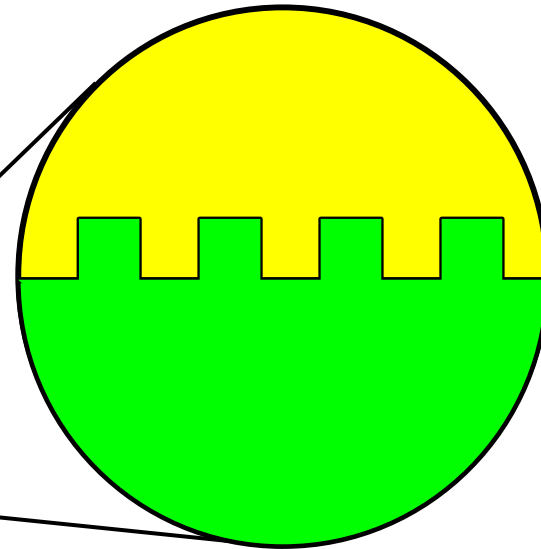
- Extra SW and Services
- Matching of Procedures
- Matching of Environment

Service Lifecycle Focuses

Prototype

Production

- Proliferation, Elaboration
  - Focus on functionality
  - Performance and scalability
    - Risks
      - Destabilisation
      - Workload
- Simplification, Automation
  - Focus on uniformity, minimisation
  - Process and procedure
  - Availability and reliability
  - Stability and robustness

User Management

- Passwd file handling > Certificate handling
  - Authentication: Gathering from VOs
  - Authorisation
    - Mapping to local identities (real or pool)
    - Building gridmap files from widely gathered info sources
- Current integration issues
  - Named accounts > Temporarily assigned accounts
    - 12,000 personal on LXPLUS
    - LCG-1: 50 static pool accounts
    - LCG-2: 80 dynamic pool accounts
  - Accounting and Auditing
    - Feedback on usage
    - Blocking / cleaning out dead-wood
- Open Issues
  - VOMS / AuthZ integration
  - Registration – interface centralised to local procedures
  - User interaction - Notification of service change or incidents
  - User + Service Manager familiarity with new services

User Management

Prototype
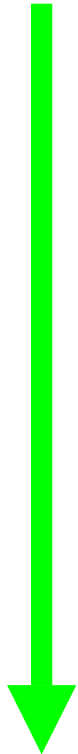
Production

- Focus on adding users
  - avoiding being a blockage to new user uptake
- Risks
  - Multiple authorisations confusion
    - c.f. multiple groups of the late 1990s
- Avoiding accumulated dead-wood and dormant accounts
  - minimise security exposures and recooperate resources

Security

- Security
  - Extending: local practices to encompass grid demands
    - Instead of processing and raising alarms …
      - Collection and storage of sufficient history of raw files
      - No history of attack patterns
  - Adapting: Tracking of incidents and blocking of compromised accounts – but now anonymous accounts and certificates to be blocked
    - Audit requirements
    - Incident response procedures
    - Revocation procedures
  - Risks:
    - Exposure: Incident propagation requires coordinated approach

SW distribution

- **OpSys and Common Applications**
  - http://www.quattor.org
  - SPMA and NCM
    - HTTP as SW distribution protocol
    - Load balanced server cluster
    - Pre-caching of SW packages on the node possible
  - Examples
    - LSF upgrade; 1000 nodes, 10 minutes, no interruption
    - Kernel upgrade; multiple version support, reboot later
    - Security upgrades; weekly, big KDE patch
  - Risks:
    - EDG toolkit – long term support

SW distribution

- **GRID + Application middleware**
  - Packaging approaches: RPM / tar / PACMAN
    - Automating a workstation orientated approach
    - Complying to enterprise management requirements
    - Configuration complexity
    - Bulky, supplier orientated
      - 1050 RPMs for LXBATCH; 220 extra for simple WN
  - Work In Progress
    - Tuning dependencies
    - Trimming unnecessary/conflicting SW and services
      - gcc, python, tomcat
  - Risks: Push aside the production knowledge
    - Harder user support, SW maintenance, incident handling

SW distribution

- ## Applications – Experiment SW
  - ### Rapid release cycles
  - ### Balancing User and Administrator preferences
    - Experiments desire for control of installation, validation and publication
    - Efficient local access; leverage SW distribution tools
    - Local Disks vs Shared file systems
  - ### Pragmatic approach in LCG-1
    - No intelligent cache yet so either
      - Copy at the start of every job
      - Shared file systems
  - ### Risks:
    - Duplication; Wasted resources in creation and housing
    - Hidden demands on reliability and stability of shared file systems

Batch Worker Nodes

- Solved some scalability issues like
  - NFS mounts of CE on WNs
    - Job Managers to address Gass-cache issues
- Interfacing to a mature batch scheduler
  - Build on lowest common denominator approach
    - Rudimentary use of batch scheduler power
    - Shifting scheduler decisions higher up the chain
  - Expose non-homogenous HW as multiple queues
- Open Issues
  - Wide area network access to/from batch nodes
    - Only 30% of 1000 LXBATCH nodes left on routed network
- Risks:
  - Drop in efficiency while learn to share with new queues

GRID Service Nodes

- Functionality deployed: RB, CE, SE, UI, …

- Open Issues
  - Coping with realistic loads – jobs; long running, complex, chaotic
  - Scalability
  - Redundancy
  - Operations
    - Learn to capture state and restart on other nodes
    - Learn how to upgrade without service interruptions
    - Do the services "age" or "pollute" the host nodes?

Milestones

- Fabric
  - Production: *SPMA and NCM*
    - July and November
  - Development: *SMS, Monitoring, FT, HMS*
    - *October, November, September, October*
    - On track for Q1 2004
- GRID Integration
  - August: *10 LXBATCH nodes integrated in LCG-1 (isolated from main LXBATCH)*
    - 5 in LCG-0, manually configured
    - Iterating on automation and productionising
  - October: *100 LXBATCH nodes…*
  - December: *Full integration of LXBATCH*
    - Delayed until early 2004

Conclusions

- A production fabric has
  - Inertia ... as a virtue!
  - Charted QoS
  - Scalability
  - Procedures and Manageability

- Cautious introduction of GRID services
  - Retain qualities and add functionality!