

Status and Integration of AP2 - Monitoring and Online Steering

Daniel Lorenz - University of Siegen

Stefan Borovac, Markus Mechtel - University of Wuppertal

Ralph Müller-Pfefferkorn – Technische Universität Dresden

HEPCG Workshop, Hamburg

14.-15. 6. 2007



GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



Overview

- 3 Partners in AP 2
- University of Siegen
 - Online steering tool RMOST
- University of Wuppertal
 - Job Execution Monitor
- Technische Universität Dresden
 - User-centric monitoring of jobs and their resource usage



Job Execution Monitor and Expert System



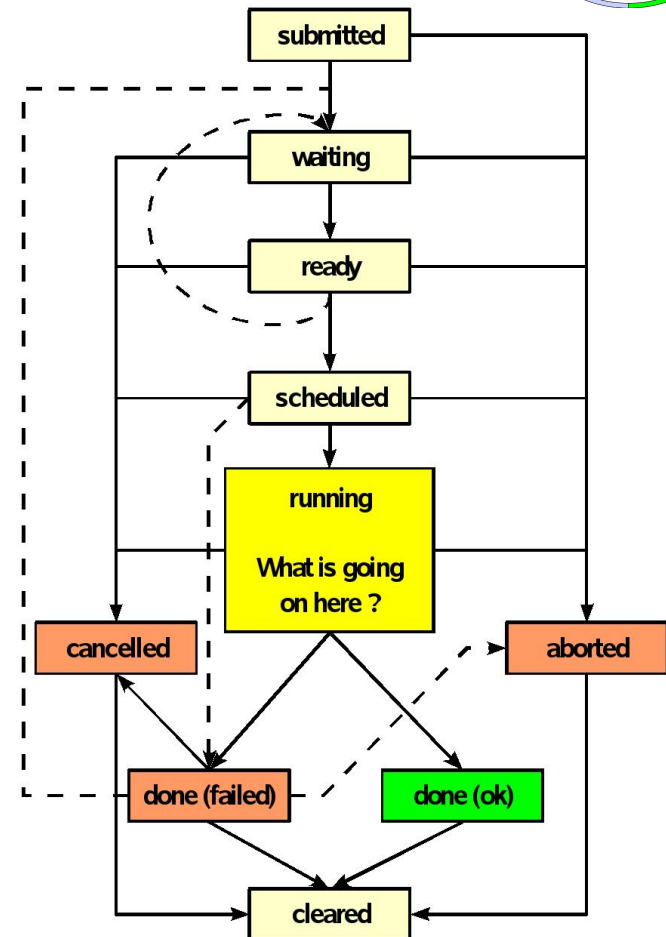
University of Wuppertal



Job Monitoring (1)

Motivation

- Thousands of jobs/day in the LHC Computing Grid (LCG)
- Jobs either fail or finish successfully
- 30% of LCG jobs fail
- LCG software observes the grid infrastructure only
- Job status is unknown to the user
- Error detection is done manually only and hard to perform
- Up to now, hardware based monitoring only (GridICE, ?)





Job Monitoring (2)

Conclusion

- Detect errors within running jobs
- Identify sources of failures
- Classify errors (expert system)



Job Execution Monitor

JEM provides data on

- Errors during execution
- Additional information about job execution (script tracing)
- Current status of hard- and software
- Information about site problems



No additional costs



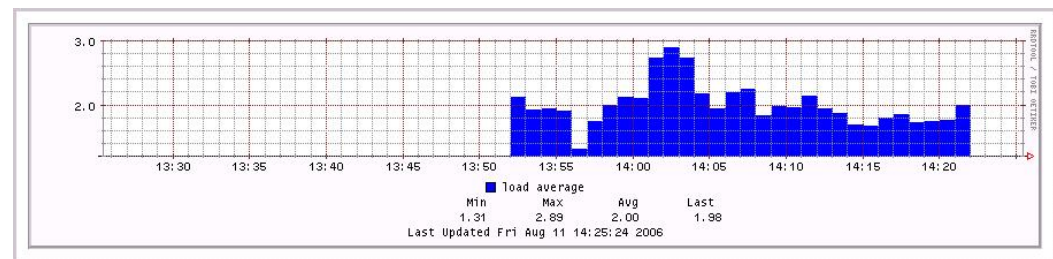
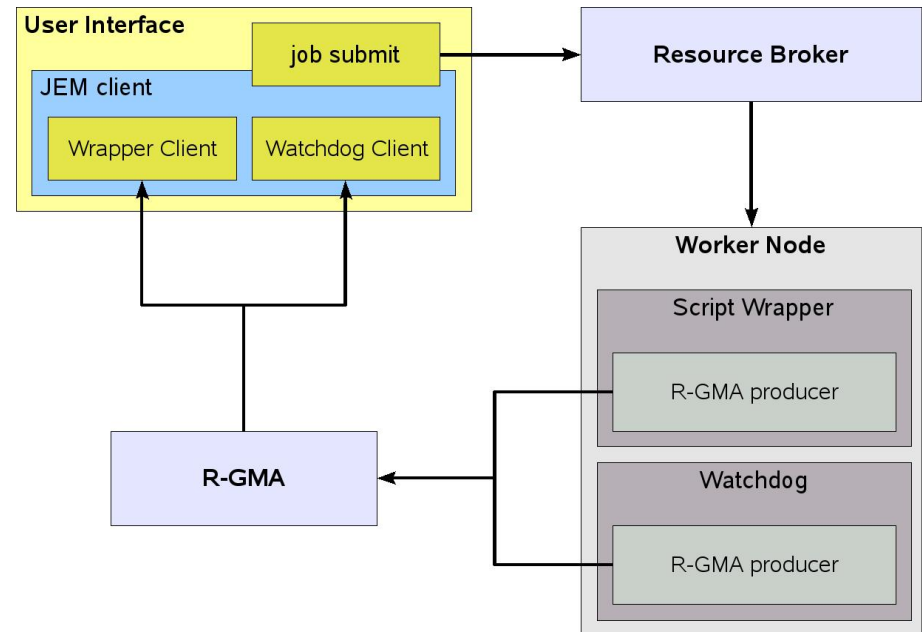
Job Monitoring (3)

Architecture of JEM

- 2 components (UI / WN)
- Information exchange via R-GMA
- Other exchange mechanisms will be studied

UI-component

- Add WN-component to a job and submit
- Manage multiple jobs
- Receive the monitoring data
- Display hardware data using rrdtool, e.g. load average





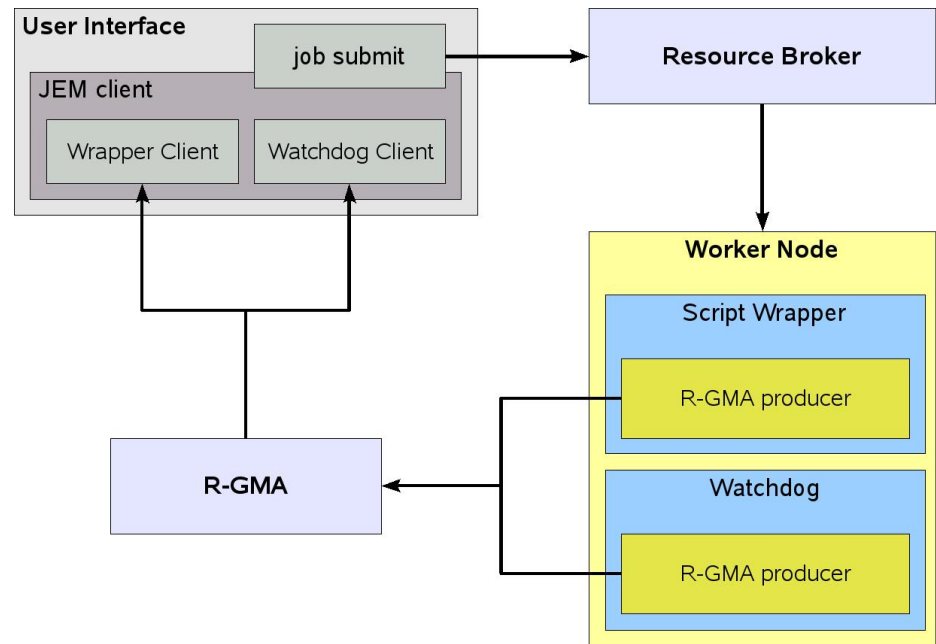
Job Monitoring (4)

WN-component

- Bash and Python tracing (script wrapper)
- Hardware monitor (watchdog)
- Publish data continuously
- On a JEM error, stop monitoring (stability)

Outlook

- Integration into ATLAS SW framework
- Classification of failures (expert system)
- Integration into GGUS



- Preexecution tests (reliability)
- Display script tracing
- ...

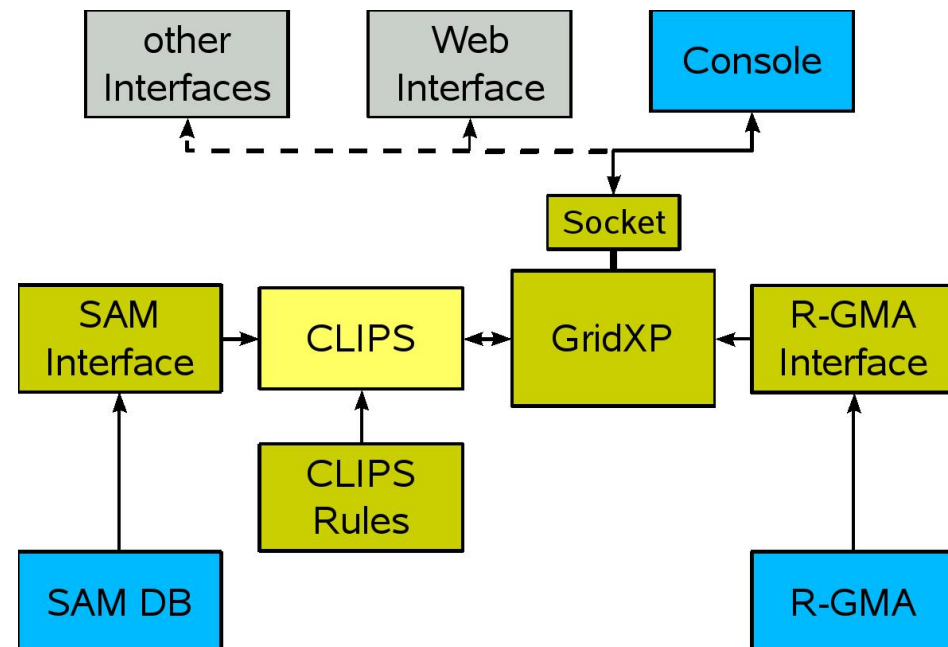


Grid Expert System (1)

Helps fixing job failures and error conditions

Architecture:

- Client-server architecture
- CLIPS expert system shell as backend
- Client-server connection via socket
- Many client interfaces possible
 - Command line
 - Web interface
 - ...





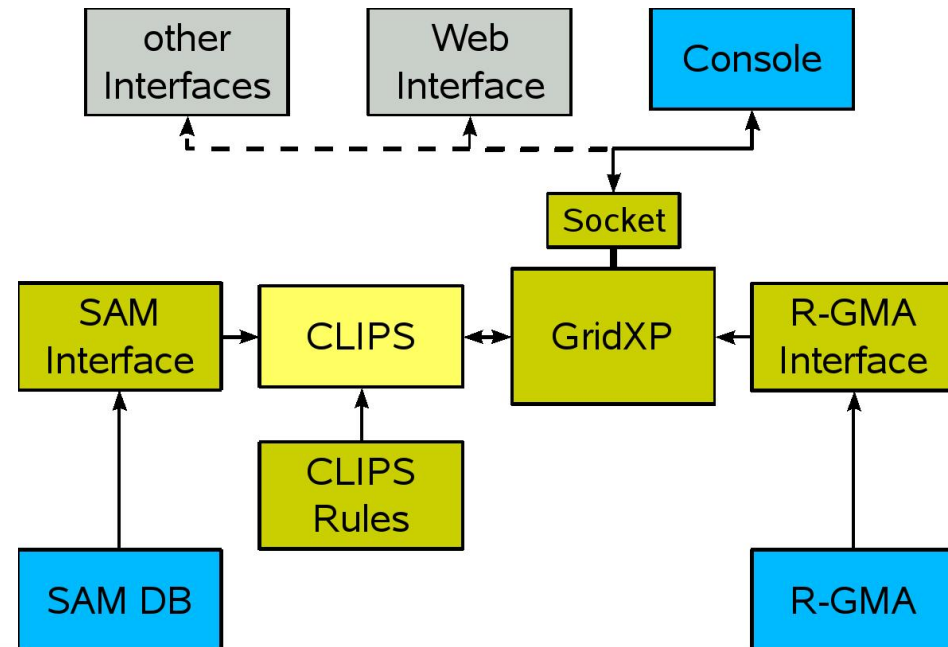
Grid Expert System (2)

Sources of data

- R-GMA
 - Data retrieval take time
 - Data continuously fed in
 - Often not accessible
- SAM database
 - Data collected when needed
 - Access restricted to known IP addresses
- CLIPS rules combine information from different sources

- Work in progress

- Definition of rules
- Classification of job failures
- Looking for additional sources of data (MonALISA?)



Online Steering of HEP Applications

University of Siegen

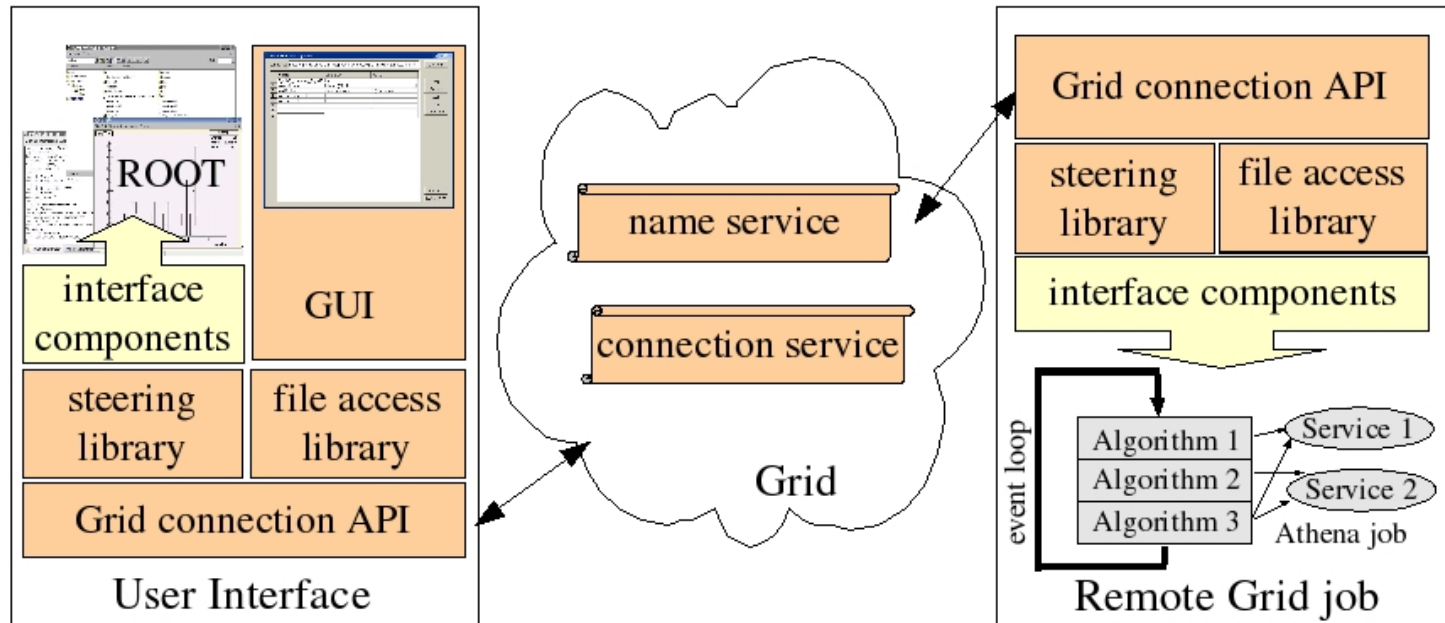


What is RMOST?

- RMOST is a **Result Monitoring and Online Steering Tool**
 - Supports ATLAS Grid jobs
 - Integrated into the Athena framework and the ROOT visualization toolkit
 - Online steering is an established mechanism to accelerate computational research

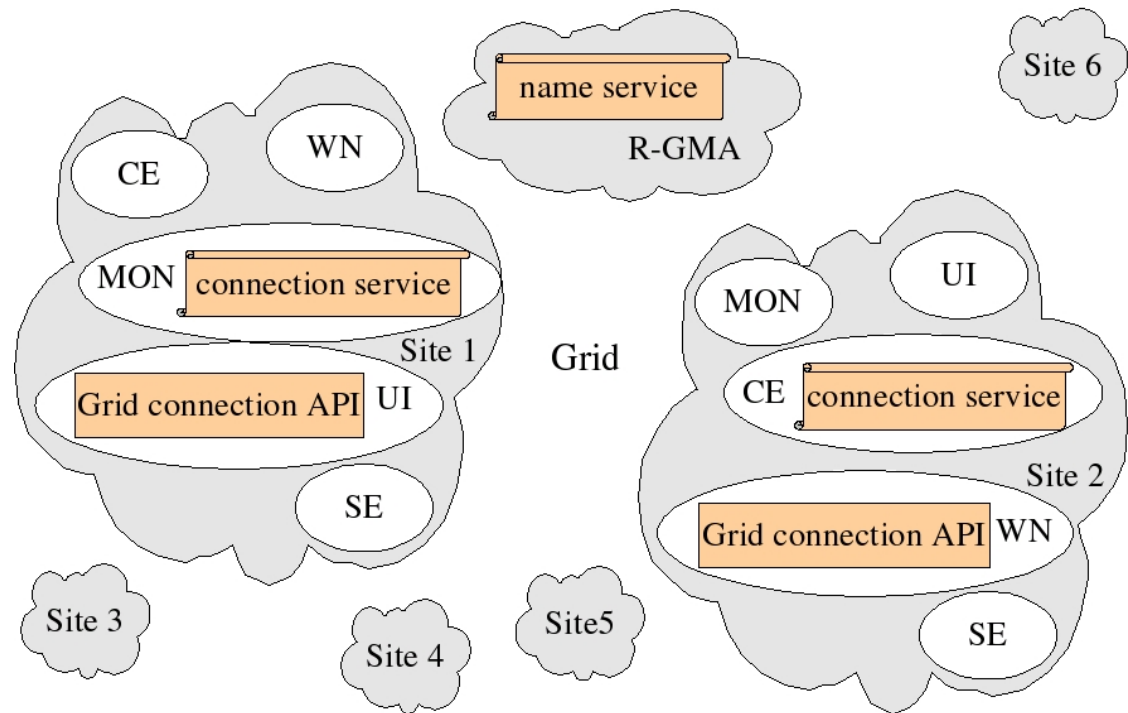
- Features of RMOST:
 - **Connect** (securely) to an **executing Grid job** at runtime
 - Monitor and **visualize intermediate results**
 - **Modify the job parameters** without resubmitting the job

RMOST Components



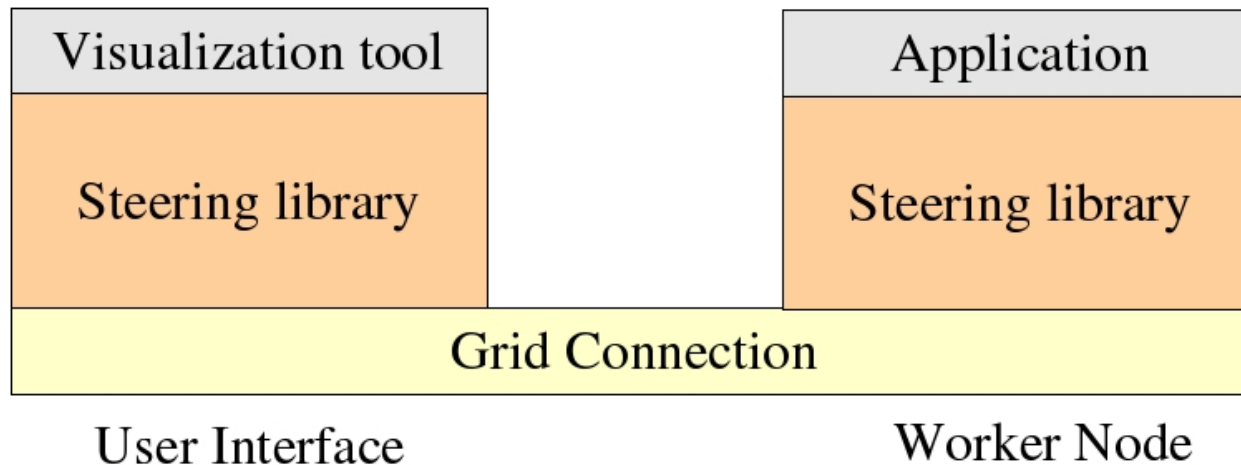
- Main libraries are **application independent**
- **Thin interface layer** to ATLAS software
- **Reusability** of most components supports **sustainability**

RMOST: Interactive Grid Connection



- Fully operable
- Needs R-GMA
 - Very slow: planned to substitute it
- Security depends on GSSAPI
 - Works with Globus TK4 and gLite (Globus TK2)
- Goal: distribute it with middleware (gLite or/and D-Grid)

RMOST: Steering Library



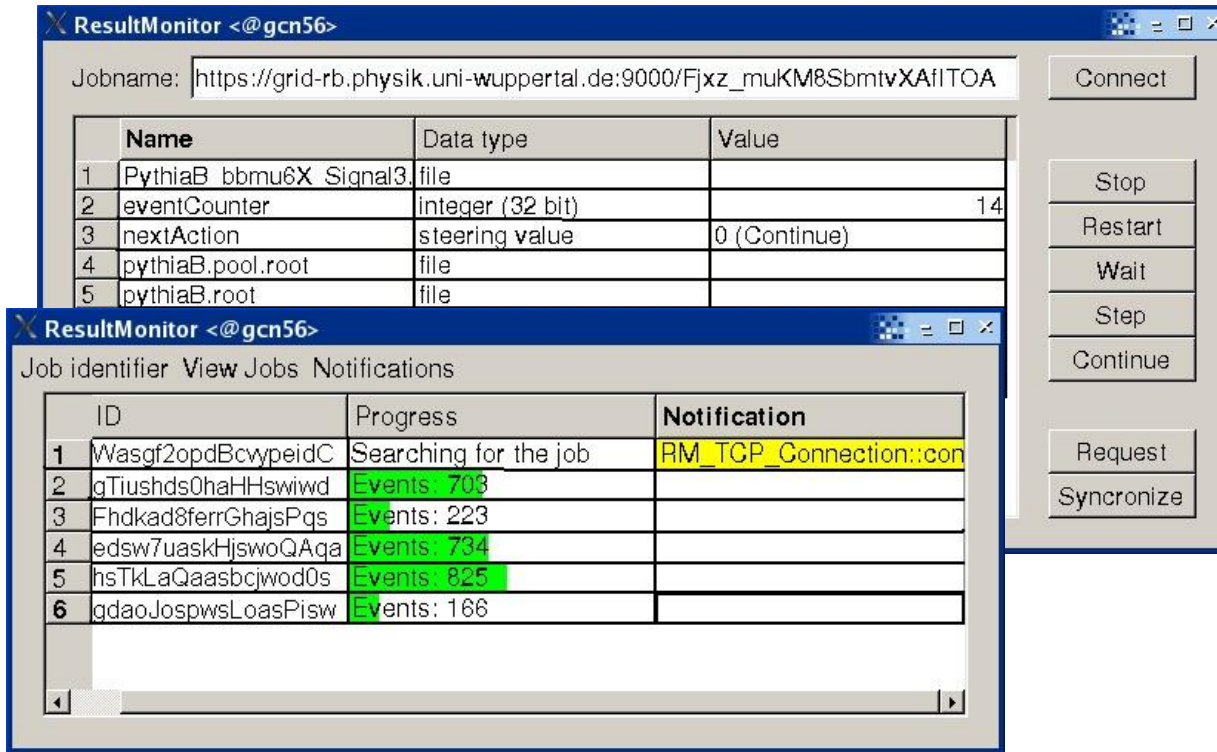
- **Application independent** steering library
- **Operable** (test phase)
- Needs the Grid connection library
- **Goal:** Integration into D-Grid and/or ATLAS

RMOST: File Access Library



- **Transparent redirection** to remote files
 - Without changes to a program's source code
 - Based on dynamic linker: preload of the file access library
- Allows to open remote result files in visualization tools, which are designed for the use with local files
- **90% ready** (scheduled for August)
- Needs steering library
- **Goal:** distribute it together with steering library

RMOST: GUI



The screenshot shows two windows from the RMOST GUI. The top window, titled 'ResultMonitor <@gcn56>', displays a table of job parameters and a set of control buttons. The bottom window, titled 'Job identifier View Jobs Notifications', displays a table of job progress and notifications.

ResultMonitor <@gcn56> Jobname: `https://grid-rb.physik.uni-wuppertal.de:9000/Fjxz_muKM8SbmtvXAfITOA`

	Name	Data type	Value
1	PythiaB bbmu6X_Signal3	file	
2	eventCounter	integer (32 bit)	14
3	nextAction	steering value	0 (Continue)
4	pythiaB.pool.root	file	
5	pythiaB.root	file	

Control Buttons: Connect, Stop, Restart, Wait, Step, Continue, Request, Synchronize

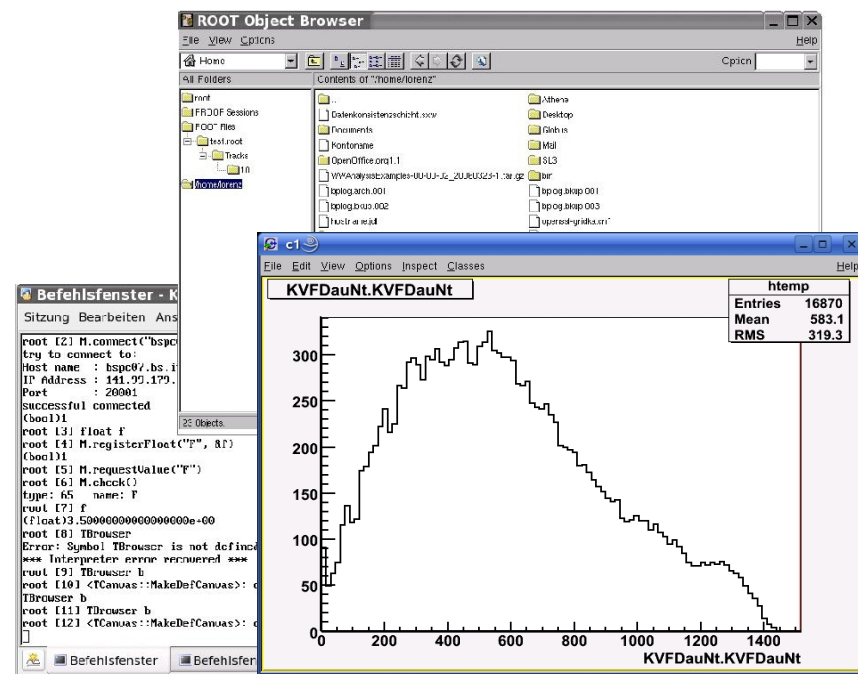
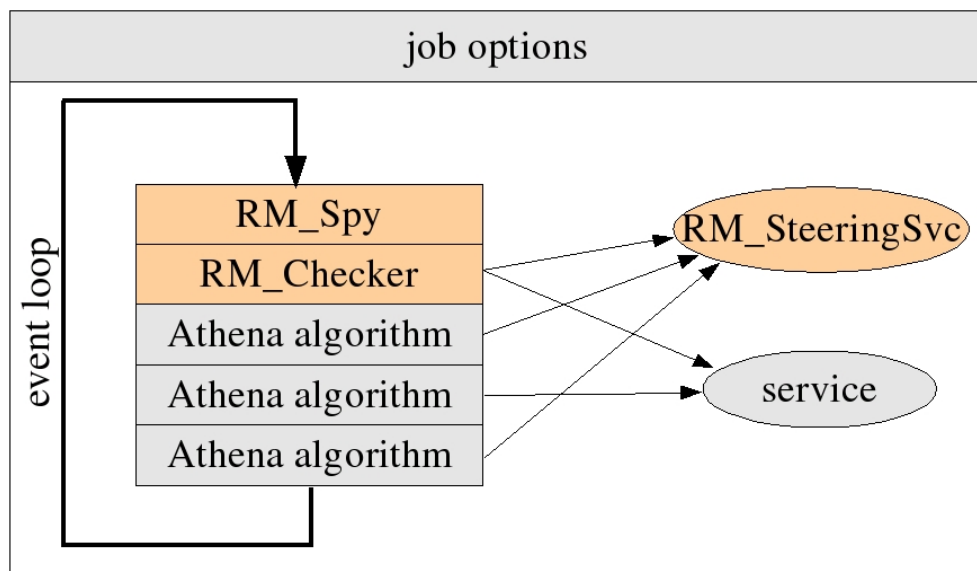
Job identifier View Jobs Notifications

ID	Progress	Notification
1	Wasgf2opdBcvpeidC	Searching for the job
2	gTiushds0haHHswiwd	Events: 703
3	Fhdkad8ferrGhajsPqs	Events: 223
4	edsw7uaskHjswoQAqa	Events: 734
5	hsTkLaQaasbcjwod0s	Events: 825
6	gdaoJospwsLoasPisw	Events: 166

- Operable (test phase)
- Needs QT3
- Goal: distribute it together with steering library
- Extendable with arbitrary data types

RMOST: Interface Components

- Extensions for ROOT and Athena
- Python interface for GANGA plugin
- Application specific



User-Centric Monitoring of Jobs and their Resource Usage

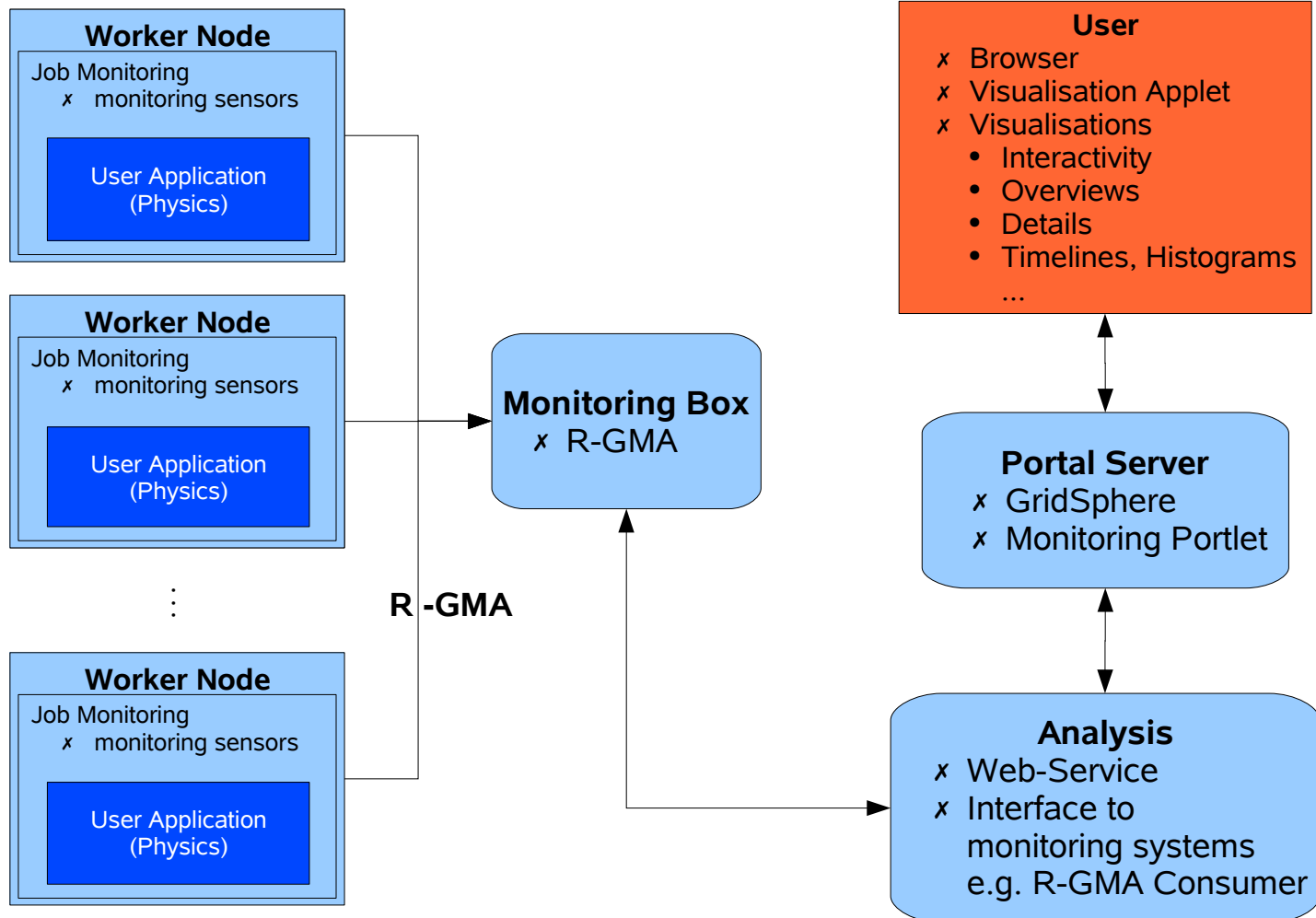
Technische Universität Dresden



Goals

- Usual scenario in HEP
 - Hundreds or thousands of jobs
- User needs overview what is going on with his/her jobs
 - Status, runtime information (resource usage)
- Two types of users
 - Physicist with his/her analyses or simulation production
 - Resource providers who want to know what is happening on their machines
or Grid management that wants to plan infrastructure
- Provide sufficient information in a helpful way
 - Graphical visualisation with interactivity
 - Collect and prepare useful information

Architecture



Major Improvements Since Last Meeting (1)

- Based on LCG worker node monitoring
- Script lcg-mon-wn is started on worker node if environment variable WN_MONITOR is set in jdl
- Extended to collect more information (configurable for user)
 - General: job ID; user name; the names of the resource broker, the computing element and the worker node (WN); job ID on the WN
 - CPU: WallClockTime; UsedCPUTime; load averages
 - Memory: real, virtual, total, and free memory; free and total swap space
 - Storage: free space on home, temporary and work directory; summary of file system properties
 - File I/O: I/O rates for every I/O access by the application
 - Network: received and transmitted network
 - Job status: information from Logging&Bookkeeping

Major Improvements Since Last Meeting (2)

- I/O Monitoring
 - Collects data for every I/O access
 - e.g.
- Authorization
 - Contacts VOMS
 - 3 authorization levels
 - user: access to her/his data only
 - admin: access to site data
 - voadmin: access to all VO data
- Visualization of the new data

Ongoing Work

- R-GMA temporary storage only
 - Persistent storage of monitoring data
- Pre-analysis algorithms for monitoring data to give users hints for possible problems
- Gathering of Resource Broker data to compare requests and results
- Improvement of user interface
 - Visualization
 - Interactivity

Integration into gLite

- Extended LCG WN monitoring uses the same interface as the old one
 - Integration = just an updated rpm for the worker node
- Other components (Web Service, Gridsphere, visualization etc.)
 - Independent of gLite / Scientific Linux
 - Only the Web Service needs some Java-Libraries to read out R-GMA
 - Can be installed on a Scientific Linux node
 - We have it running on SuSE installations

Integration into DGI or Other Projects

- System design generic
- Web Service for data gathering + visualization + Gridsphere integration independent of monitoring system and HEP
 - Provide convenient interface for users
- Only LCG Worker Node monitoring, R-GMA and VOMS are HEP specific
- Allows to plugin other monitoring and VO management systems
 - Globus / MDS
 - Unicore
 - Ganglia
 - ...

Summary and Outlook

- Integration of AP 2
 - Generic frameworks
 - Many Components are independent of application specific software
 - Cooperations with GANGA
 - Goal: Integrate software into different distributions

- Monitoring Workshop next week in Wuppertal
 - All D-Grid projects
 - Cooperation among D-Grid monitoring groups
 - Possibility to present already existing tools