



Job Execution Monitoring in Ganga

Martin Rau

D-Grid – Monitoring Workshop
26.05.2008 / 27.05.2008
Dresden

- Motivation
- JEM (Job Execution Monitor)
 - Leistungen
 - Vorteile
 - Funktionsweise
- Ganga
 - Integration von JEM in Ganga
 - Modifikation Input-/OutputSandbox
- Ausblick

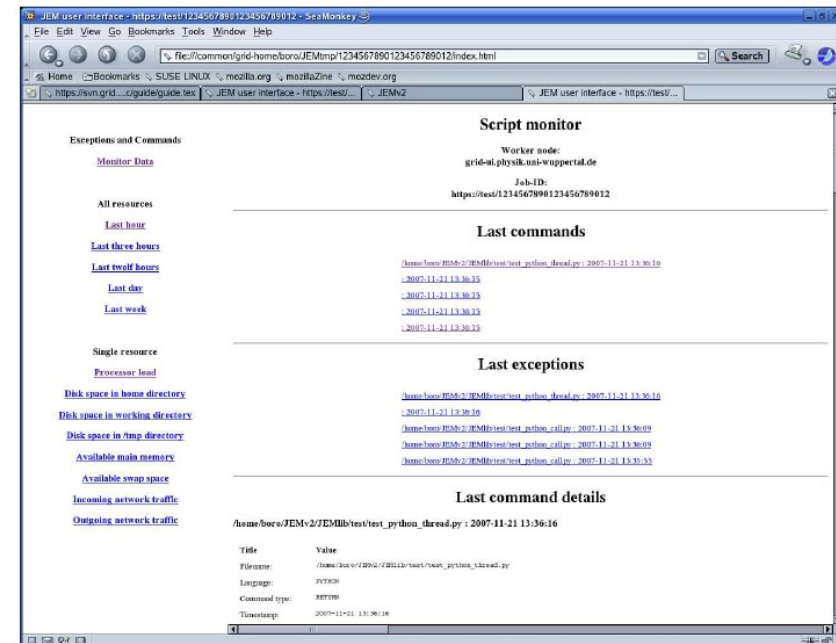
- Problem
 - ~20 % aller Jobs werden abgebrochen
 - ineffiziente Ressourcennutzung
 - unzureichende Fehlerausgaben an den User
 - keine Möglichkeit den Job zu überwachen

- Problemlösung
 - JEM (Job Execution Monitor)
 - Monitoring direkt am Job

- Überwachung von Skriptprogrammen
 - schrittweise Befehlsausführung
 - protokolliert alle Ein-/Ausgaben
 - Emergency Mode für interne Fehler
- modularer Systemaufbau
 - Überwachung von Python- und Bashskripten
 - einfache Einbindung von Modulen anderer Sprachen

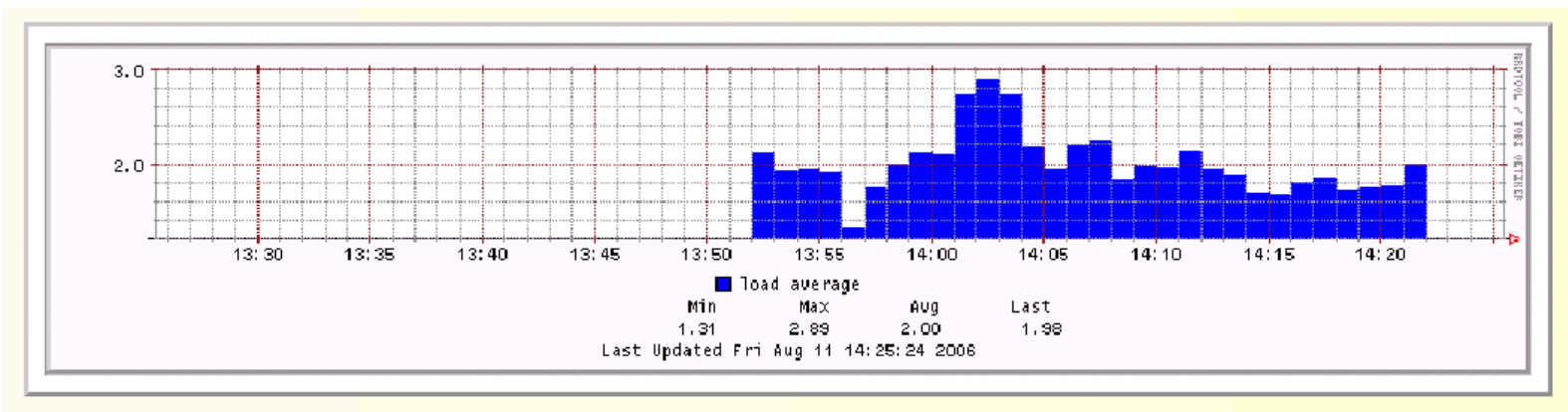
- Überwachen der ausgeführten Jobs
- Protokoll der ausgeführte Befehle und Ergebnisse
- keine Job-Modifikation nötig
- Ressourcenüberwachung

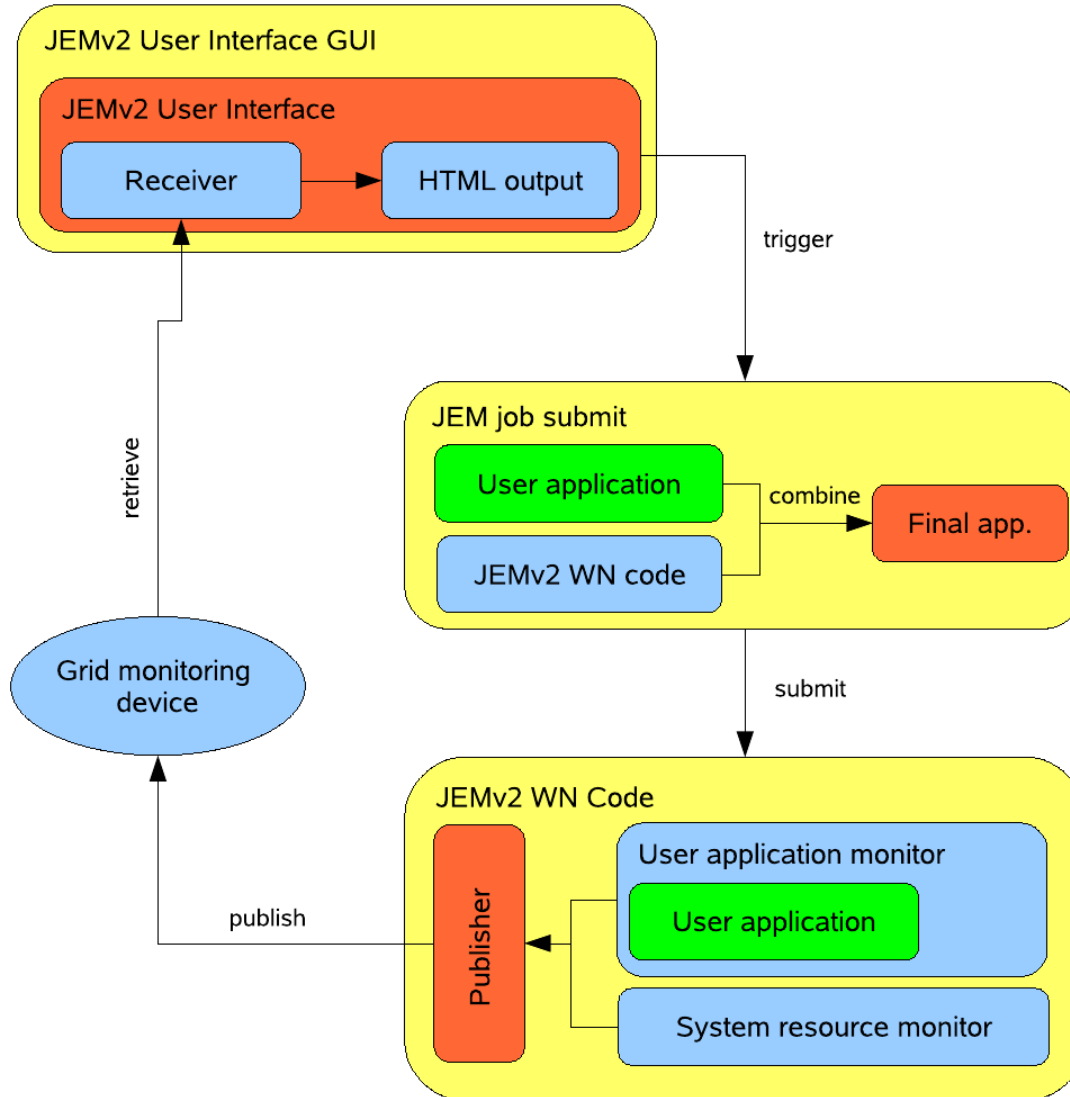
- kaum Overhead
 - bei Bashskripten ca. 10%
 - bei Pythonskripten ca. 100% (Setup-Phase)
 - Binärdatei kein Overhead
- mehrere Output-Formen
 - HTML-Seiten
 - Log-Files



- direkte Überwachung
 - nahezu Echtzeitüberwachung
 - Datenversand mit R-GMA, Socket
 - Datenversand mit MonALISA (in Entwicklung)
- indirekte Überwachung
 - Datenversand über OutputSandbox
 - ◆ JEM-Statuslog
 - ◆ JEM-Monitorlog

- zuletzt ausgeführte Kommandos / Exceptions
- Variablenwerte bei Job-Fehler
- Skriptname, Funktionsname, Zeilennummer
- Systemressourcen über gesamten Zeitraum
- Zugriff auf stdout & stderr
- Exit-Codes



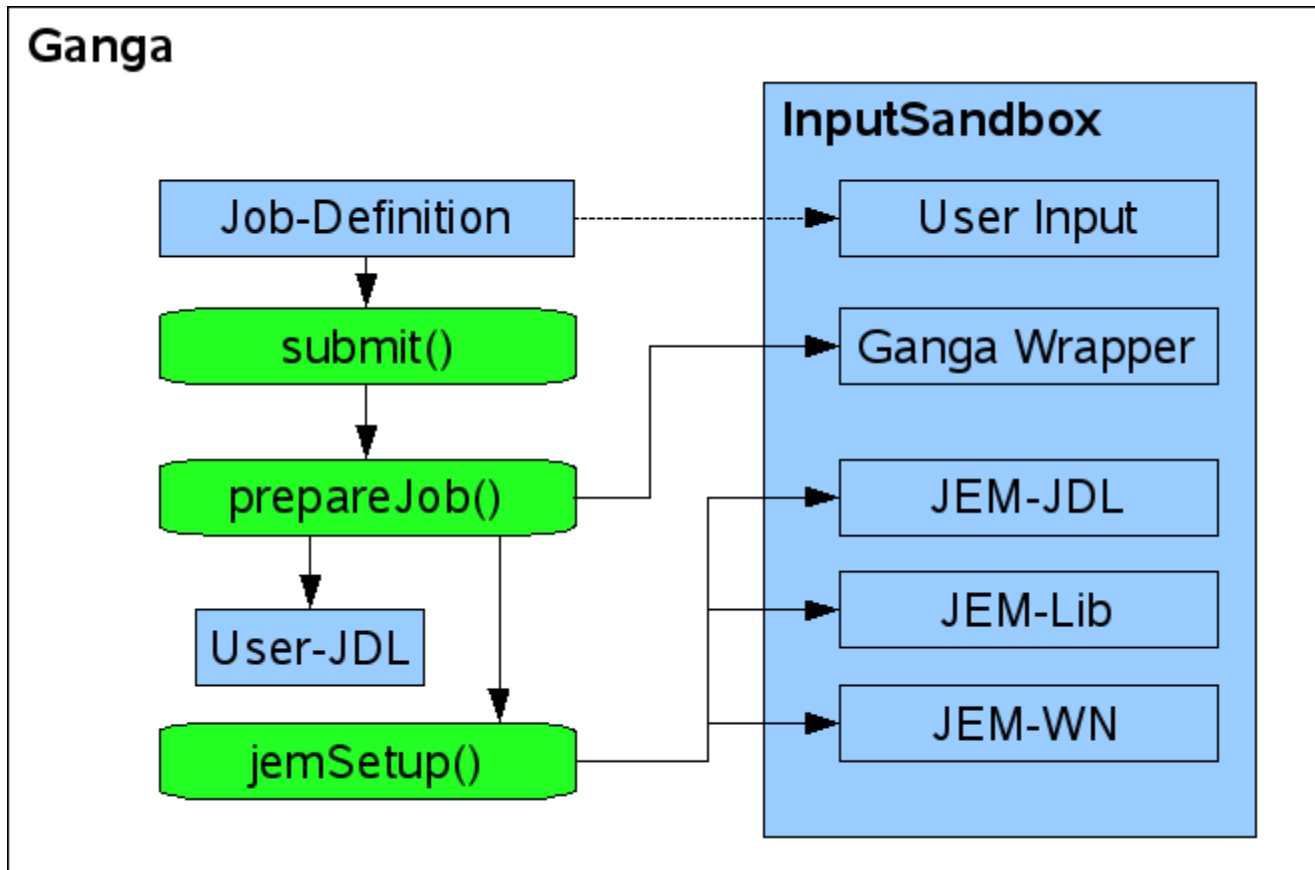


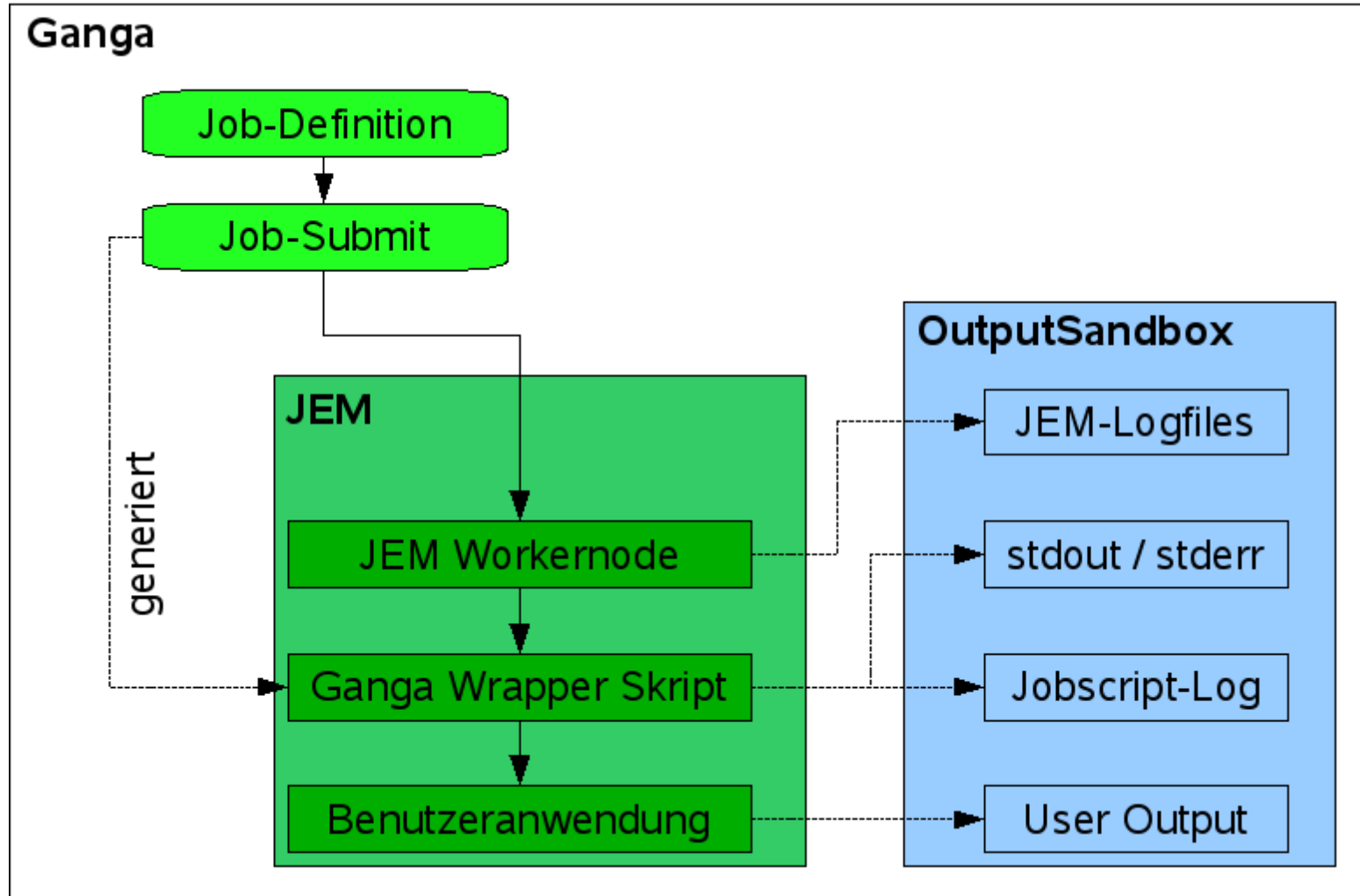
- Was ist Ganga?
 - Front-End Client für Jobdefinition & Jobmanagement
 - entwickelt für Auswertungen in ATLAS und LHCb
 - wichtigstes Tool für verteilte Analyse in ATLAS

■ Integration

- JEM wird als Plugin in Ganga eingebunden
- Monitoring wird über ein Flag im LCG-Backend aktiviert / deaktiviert
- Vorgang für User transparent

```
j=Job()  
j.application=Executable()  
j.application.exe='/bin/sh'  
j.application.args="test.sh"  
j.outputsandbox=["user.data"]  
j.inputsandbox=["test.sh"]  
j.backend=LCG()  
j.backend.monitoring=True
```





- aktueller Status der Integration
 - Monitoring Output in Form von Logfiles
 - rudimentäre Tests abgeschlossen

- Ausblick
 - reale Testjobs (u.a. in Athena)
 - (Echtzeitmonitoring in Ganga)
 - JEM wird stetig weiterentwickelt

Vielen Dank für Ihre Aufmerksamkeit!

<http://www.grid.uni-wuppertal.de/grid/>