Monitoring Solution for Workflow – based Satellite Image Processing Applications Running in a Grid Environment

Author: 
Dragoș Dumitru Sbirlea

Coordinators: 
Prof. Dr. Ing. Valentin Cristea
As. Dr. Ing. Florin Pop
Summary

• Satellite Image Processing
• Grids
• gProcess Workflows
• Monitoring Solution Goals
• Improving the gProcess Architecture
• Sensors
• Visualization Capabilities
• Runtime Estimates
• Conclusion
Satellite Image Processing

• Domains
  – Identification of fertile land
  – Monitoring disasters (floods, eruptions)
  – Discovery of geological characteristics
  – Identification of ore resources
  – Environment characteristics
    • Draft
    • Vegetation
    • Water
• The project adds monitoring to all stages of such a processing.
gProcess

• Engine for workflow-based satellite image processing

• Example:

Enhanced Vegetation Index

\[ EVI = \frac{2.5 \times (NIR - RED)}{NIR + 6 \times RED + 7.5 \times BLUE + 1} \]
Monitoring Solution Goals

• Data Accounting and Visualization
  – System and application parameters

• Error Management
  – Discover
  – Log
  – Notify

• Scheduling improvement through statistics

• Profiling & system tuning help
Improving the gProcess Architecture

1. Client Application
2. gProcess webservices & Scheduler & Workflow job management
3. MonaLisa Repository
4. Database Server
5. Feedback System: Statistics help improve scheduling

Automated Error Detection System & Statistics computing unit

gLite Cluster

Client Application
Monitoring System Components

- Colored blocks -> my implemented components
- Gray blocks -> libraries & frameworks used
Sensors

• Implemented by me:
  – System sensors
    • CPU (user, system)
    • Memory (free, total, percentual, swap)
  – Job related sensors
    • LifeCycle steps
  – Extensible sensors engine

• Hundreds more accessible through MonALISA
Visualization

• Using MonALISA Web Client
  – Histogram and real time view for parameter values

• Using customized MonALISA Repository
  – History or Near Realtime
  – Custom graphs
  – Combined graphs
  – Statistical analysis
Runtime Estimates

• Scheduling improvement through runtime estimates
  
  – Fixed Percentile Estimate (FPE)
    \[ \text{Cost}'_{\text{wf}}(N+1) = \text{Cost}'_{\text{wf}}(N) \times (1-p) + \text{Cost}_{\text{wf}}(N) \times p \]

  – Decreasing Percentile Estimate (DPE)
    \[ \text{Cost}'_{\text{wf}}(N+1) = \text{Cost}'_{\text{wf}}(N) \times (1-p/N) + \text{Cost}_{\text{wf}}(N) \times p/N \]
Scenarios & Testing

- Testing validates the theoretical analysis
Conclusions

• Real time access to monitoring data  
  $\Rightarrow$ new possibilities for scheduling and error detection

• Made possible by a powerful monitoring framework

• Contributions:
  – Workflow runtime estimates
  – Monitoring solution implementation
  – gProcess operators implementation
  – MonALISA module for PBS updated