Monitoring Applications for End-user Distributed Analysis

Gerhild Maier

September 23rd 2009

Outline

- Monitoring distributed analysis jobs

- QAOES:
  1. detecting problematic Grid components
  2. solving problems with an Expert System

- Outlook

Monitoring Grid Jobs

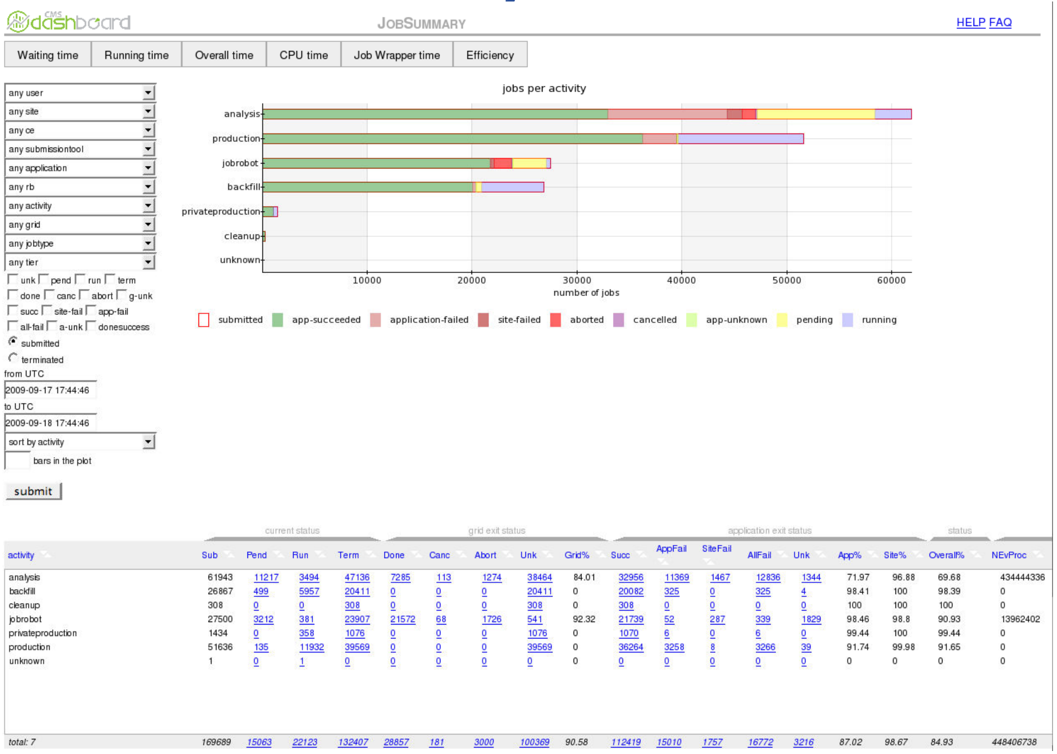
- Reliable Grid infrastructure needed for LHC experiments' success
- Detailed and trustworthy monitoring helps improve the Grid
- Production jobs on the Grid:
  - coordinated activity by experts
  - load on sites and storage is predictable
- Analysis jobs on the Grid:
  - chaotic activity
  - behaviour not predictable

Monitoring Analysis Jobs

- challenging task
- example from CMS:
  - 100-200 users running analysis jobs every day
  - since 01/2008, 1000 distinct users submitted analysis jobs, number expected to increase with LHC start-up
- users not necessarily experienced → simple, user-friendly monitoring needed
- strong monitoring for user support groups

Dashboard Applications (1/3)

- interactive job summary
  - shows what is going on now
  - distribution of jobs by site, CE, user, submission tool, application version, dataset, etc…, exit status
  - CPU and wall clock time of jobs



any user
any site
any ce
any submissiontool
any application
any rb
any activity
any grid
any jobtype
any tier

□ unk □ pend □ run □ term
□ done □ canc □ abort □ g-unk
□ succ □ site-fail □ app-fail
□ all-fail □ a-unk □ donesuccess

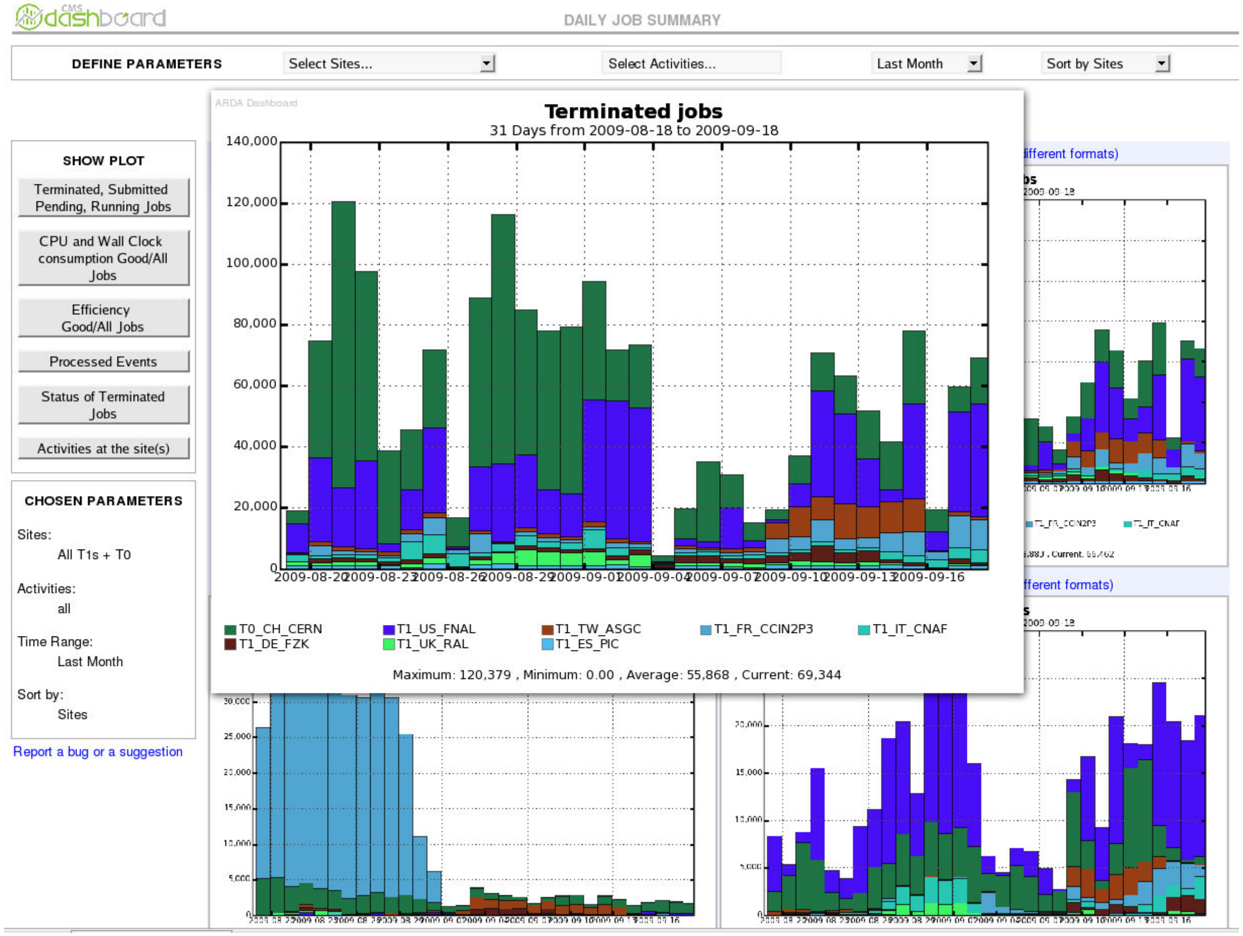
● submitted
○ terminated

from UTC
2009-09-17 17:44:46
to UTC
2009-09-18 17:44:46
sort by activity
bars in the plot

	current status				grid exit status					application exit status							status	
activity	Sub	Pend	Run	Term	Done	Canc	Abort	Unk	Grid%	Succ	AppFail	SiteFail	AllFail	Unk	App%	Site%	Overall%	NEvProc
analysis	61943	11217	3494	47136	7285	113	1274	38464	84.01	32956	11369	1467	12836	1344	71.97	96.88	69.68	434444336
backfill	26867	499	5957	20411	0	0	0	20411	0	20082	325	0	325	4	98.41	100	98.39	0
cleanup	308	0	0	308	0	0	0	308	0	308	0	0	0	0	100	100	100	0
jobrobot	27500	3212	381	23907	21572	68	1726	541	92.32	21739	52	287	339	1829	98.46	98.8	90.93	13962402
privateproduction	1434	0	358	1076	0	0	0	1076	0	1070	6	0	6	0	99.44	100	99.44	0
production	51636	135	11932	39569	0	0	0	39569	0	36264	3258	8	3266	39	91.74	99.98	91.65	0
unknown	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
total: 7	*169689*	*15063*	*22123*	*132407*	*28857*	*181*	*3000*	*100369*	*90.58*	*112419*	*15010*	*1757*	*16772*	*3216*	*87.02*	*98.67*	*84.93*	*448406738*

Dashboard Applications (2/3)

- historical interface
  - job statistics distributed over time



DAILY JOB SUMMARY

DEFINE PARAMETERS: Select Sites... | Select Activities... | Last Month | Sort by Sites

SHOW PLOT
- Terminated, Submitted Pending, Running Jobs
- CPU and Wall Clock consumption Good/All Jobs
- Efficiency Good/All Jobs
- Processed Events
- Status of Terminated Jobs
- Activities at the site(s)

CHOSEN PARAMETERS

Sites:
All T1s + T0

Activities:
all

Time Range:
Last Month

Sort by:
Sites

Report a bug or a suggestion

ARDA Dashboard

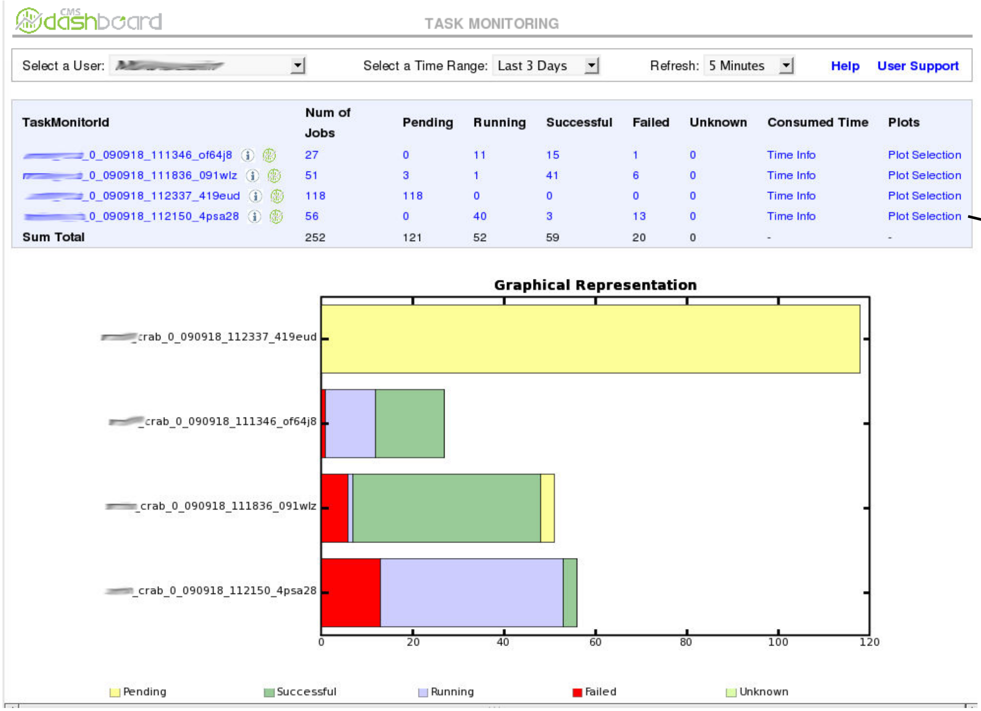
Terminated jobs

31 Days from 2009-08-18 to 2009-09-18

Maximum: 120,379 , Minimum: 0.00 , Average: 55,868 , Current: 69,344

Dashboard Applications (3/3)

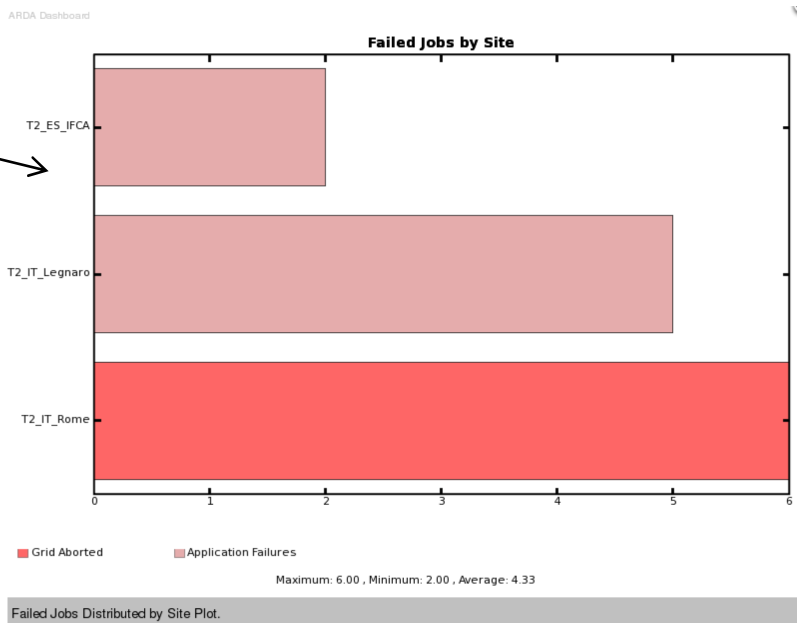
- task monitoring (deployed for CMS)
  - provides complete information about analysis tasks
  - job status of individual jobs in the task
  - assists in detecting problems
  - provides debugging information



CMS dashboard TASK MONITORING

Select a User: [redacted] Select a Time Range: Last 3 Days Refresh: 5 Minutes Help User Support

TaskMonitorId	Num of Jobs	Pending	Running	Successful	Failed	Unknown	Consumed Time	Plots
[redacted]_0_090918_111346_of64j8	27	0	11	15	1	0	Time Info	Plot Selection
[redacted]_0_090918_111836_091wlz	51	3	1	41	6	0	Time Info	Plot Selection
[redacted]_0_090918_112337_419eud	118	118	0	0	0	0	Time Info	Plot Selection
[redacted]_0_090918_112150_4psa28	56	0	40	3	13	0	Time Info	Plot Selection
Sum Total	252	121	52	59	20	0	-	-



Maximum: 6.00 , Minimum: 2.00 , Average: 4.33

Failed Jobs Distributed by Site Plot.

QAOES - overview

- QAOES = Quick Analysis Of Error Sources

- Goal:
  - detect problematic Grid components
  - find error source, not only impact
  - facilitate the problem solving

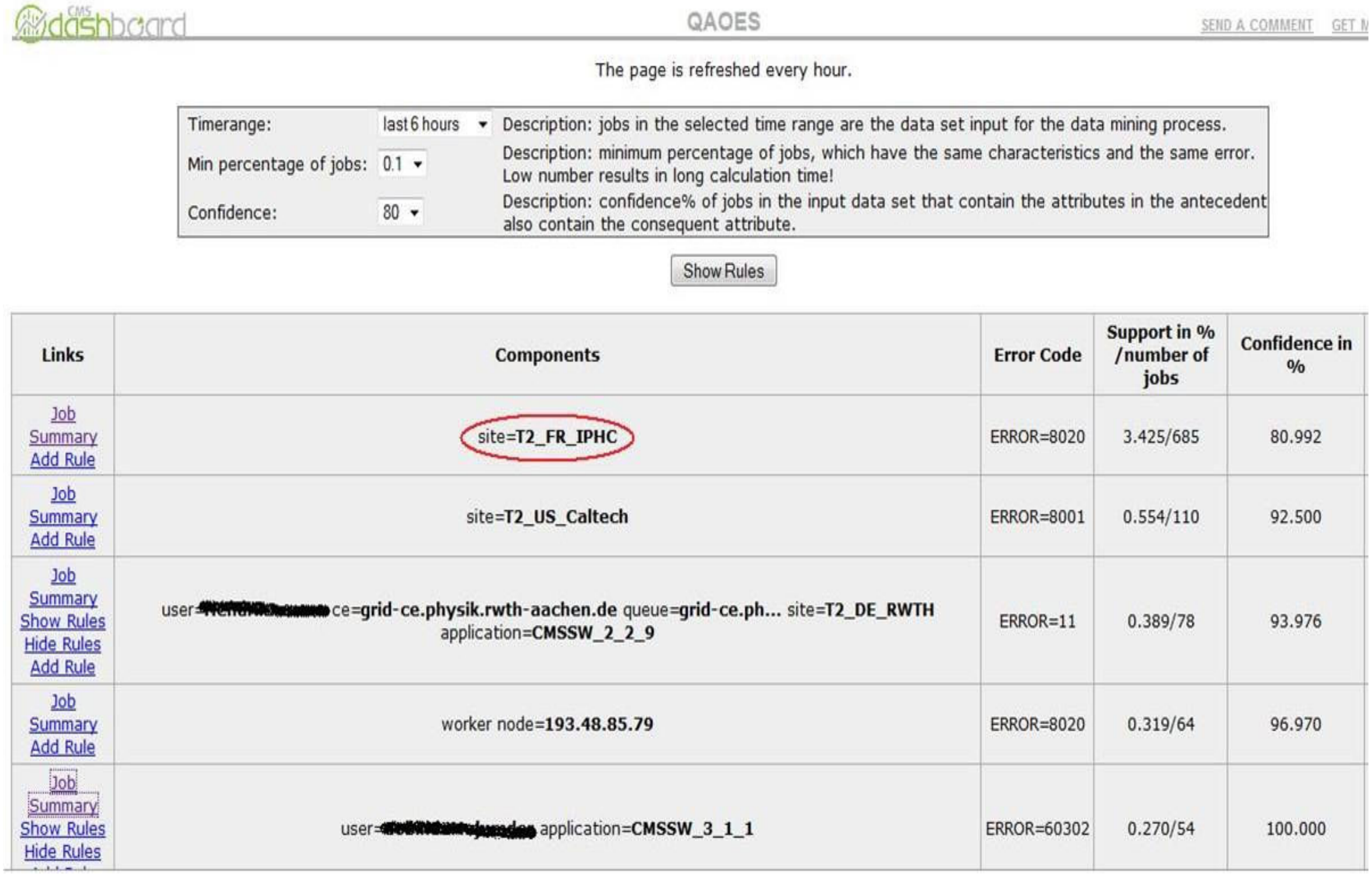
- Approach:
  1. Association Rule Mining (ARM) on job monitoring data
  2. Expert System to collect and reuse human expertise

Association Rule Mining (1/2)

- Input:
  - job monitoring data with a number N of attributes
  - attribute: job characteristics like user, site, dataset, se,...
- Goal: find dependencies between items
- Association rule:
  - A →C, where A, C sets of items
  - A = antecedent, C = consequent
  - item: attribute-value pair
- Quality of an association rule:
  - support: s% of the data include all items
  - confidence: c% of the data including the antecedent also include the consequent

Association Rule Mining (2/2)

- Output: list of potential problems



The page is refreshed every hour.

Timerange: last 6 hours — Description: jobs in the selected time range are the data set input for the data mining process.

Min percentage of jobs: 0.1 — Description: minimum percentage of jobs, which have the same characteristics and the same error. Low number results in long calculation time!

Confidence: 80 — Description: confidence% of jobs in the input data set that contain the attributes in the antecedent also contain the consequent attribute.

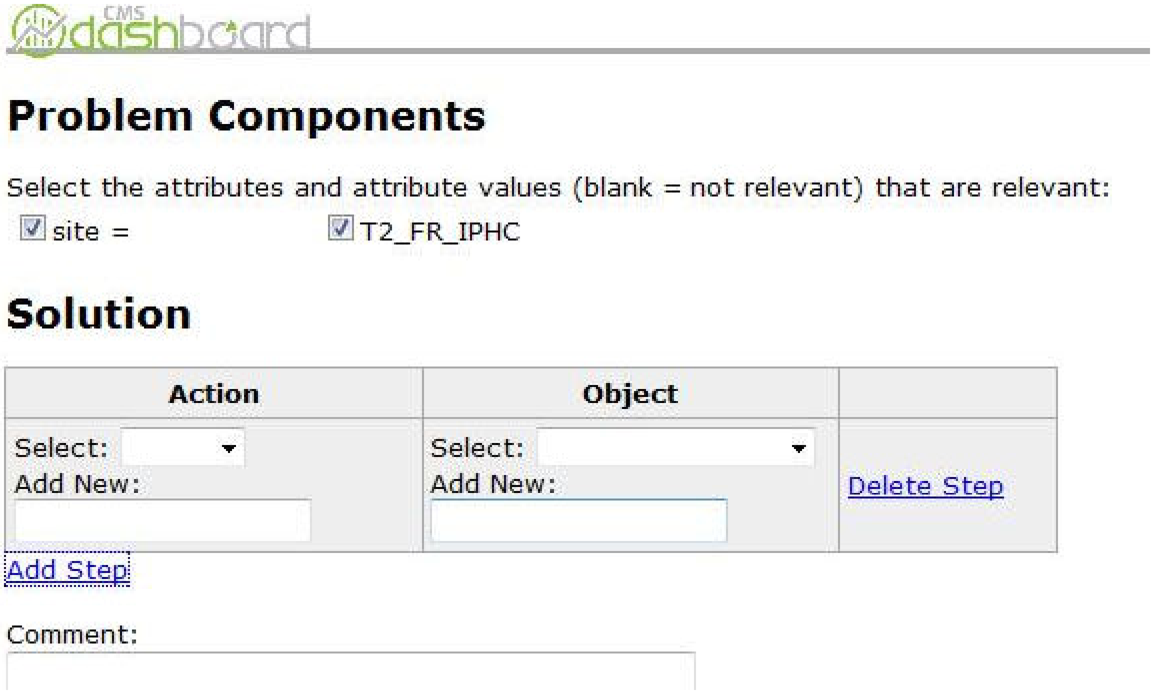
Links	Components	Error Code	Support in % /number of jobs	Confidence in %
Job Summary Add Rule	site=**T2_FR_IPHC**	ERROR=8020	3.425/685	80.992
Job Summary Add Rule	site=**T2_US_Caltech**	ERROR=8001	0.554/110	92.500
Job Summary Show Rules Hide Rules Add Rule	user=[redacted] ce=**grid-ce.physik.rwth-aachen.de** queue=**grid-ce.ph...** site=**T2_DE_RWTH** application=**CMSSW_2_2_9**	ERROR=11	0.389/78	93.976
Job Summary Add Rule	worker node=**193.48.85.79**	ERROR=8020	0.319/64	96.970
Job Summary Show Rules Hide Rules	user=[redacted] application=**CMSSW_3_1_1**	ERROR=60302	0.270/54	100.000

Expert System (1/2)

- Use human expertise to improve problem solving on the grid
- 2 main components:
  1. knowledge base: collection of human expert knowledge in the format of rules
  2. inference engine: matches potential problems to rules in the knowledge base
- Rule = ProblemDefinition + SolutionOption
  - ProbelmDefinition: attributes, and values (optional)
  - SolutionOption: number of steps; step = action applied to object

Expert System (2/2)

- Add a rule:



Problem Components

Select the attributes and attribute values (blank = not relevant) that are relevant:

☑ site = ☑ T2_FR_IPHC

Solution

Action	Object	
Select: ▾ Add New:	Select: ▾ Add New:	Delete Step

Add Step

Comment:

- Rule ranking:
  - by user
  - automatically: ranking decreased over time

Outlook and Summary

- Outlook:
  - improve the inference engine

- Summary:
  - importance of monitoring for analysis users
  - failure detection using job monitoring tools
  - QAOES approach of data mining combined with expert system on top, currently applied to CMS analysis job monitoring data

Links

- Experiment Dashboard:
  http://dashboard.cern.ch
- QAOES:
  http://dashb-cms-mining-devel.cern.ch/dashboard/request.py/qaoes
- Twiki:
  https://twiki.cern.ch/twiki//bin/view/ArdaGrid/AutomaticFaultDetection