Review of job efficiencies at CERN
status report

James Casey, Daniel Rodrigues, Ulrich Schwickerath

Conclusions from March presentation

- there is not one simple reason for bad job efficiencies
- an efficient and complete monitoring is necessary to identify avoidable inefficiencies
- the use of pilot jobs often makes site administrators blind for what is going on

=> we need the help of the experiments to track these problems down

March requests to the experiments:

- experiment frameworks need to be instrumented to record timings
- all jobs need to be monitored, not only the successful jobs

Implementation ideas

... how can we do that ?

Mandate: define an interface/monitoring system which

- Catches all jobs
- Can get the full picture over what a job is doing
- Provide a common interface
- Is easy to use/ easy to apply from framework developers

Idea: use MSG, see also:

Proposal for improving Job Reliability Monitoring by James Casey

see GDB April 2008

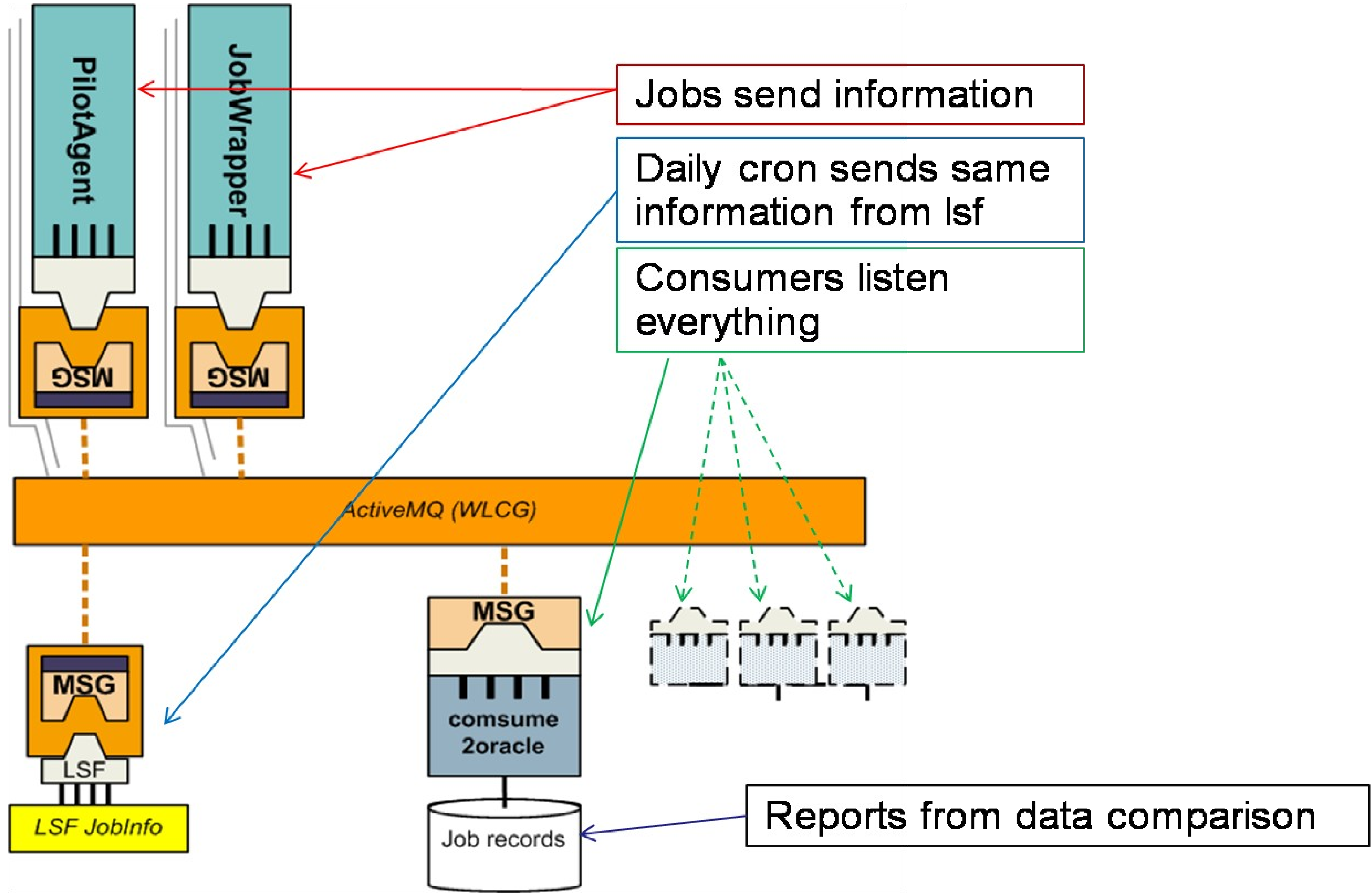
MSG Architecture: Messaging System for the Grid:

Based on Messaging Oriented Middleware, provides:

- Flexible architecture: Multicast/ Point to point modes;
- Reliable delivery of messages;
- Highly Scalable through Network of Brokers;
- Currently based on Apache ActiveMQ message broker
- Prototype using python:
  - Msg-publish : no dependencies, easy to deploy;
  - Msg-consume2oracle : reading records into an oracle database;

Easy deployment in 3 steps:

- Agree on the semantics of records;
- Install and configure msg-publish;
- Send messages calling msg-publish;


PilotAgent
JobWrapper
MSG
MSG
Jobs send information
Daily cron sends same information from lsf
Consumers listen everything
ActiveMQ (WLCG)
MSG
LSF
LSF JobInfo
MSG
comsume 2oracle
Job records
Reports from data comparison

Proof of concept: a first prototype implementation

Idea: Instrumentation of the job starter script which is used at CERN

- feature provided by LSF, job wrapper
- run with user privileges
- for local jobs: creates local job directories and does the cleanup later
- logs additional information (job start, job end) on the worker node
- ideal candidate to log job start and job end records into a central ddb

Prototype implementation: status report

Use two independent sources of data:

- Batch system accounting files (from LSF):
  - Available only after the job has ended
  - Contains information for all jobs, including those which never started
  - Contains the full picture of the job, including all wrappers
- Instrumented job starter
  - Gives the information on what the job was doing in each phase

Need to be able to match these two sources

Use a unique primary key which is made of

- The local job ID
- The local user
- The execution host name

Prototype/ proof of concept status at CERN

Accounting records:

- Created once per day at midnight
- Uploaded in a single bulk operation for one full day

Status: in place for the CERN production system

- Contains all jobs, including non-grid activity
- May need to add a filter for non-LHC related jobs

Job start/Job end records from instrumented “job_starter”

- A job start record is created in the job starter script before the user job is executed
- A Job end record is created in the job starter script when the payload returns

Status: in place in the LSF test system right now (accessible via ce110)

How does it work ?

Batch system accounting records:

Create a file with records for each job like that:

```
localjobid:666585
ownerdn:uschwick
wnhostname:lxb8365
state:JOBEND
context:LSF
cpuUsage:0.760883
memoryUsage:15
cpufactor:1.900000
exitcode:0
walltime:29
submittime:1212755375
finishtime:1212755406
EOT
```

Primary key (localjobid, ownerdn, wnhostname)

Job state and context (state, context)

Record end (EOT)

Context: for now, can be one of "JOBWRAPPER" or LSF (later PBS etc)

There is a script which allows to upload the resulting file in one bulk operation

Conclusions

We have a working prototype

There are still some things which can be improved

The CERN jobs accounting information is already available in the system

- Aim: get some experience with this
- Restriction: The information records must be given in a file. Piping is not working yet
- next step: how to use this data ?

The required software is already available on the worker nodes at CERN.

Experiments can start to use it to record job transition information.

***Contacts:* James Casey, Daniel Rodrigues and myself**

Example: (preliminary) LSF CPU overhead measurement

A1 | LOCALJOBID

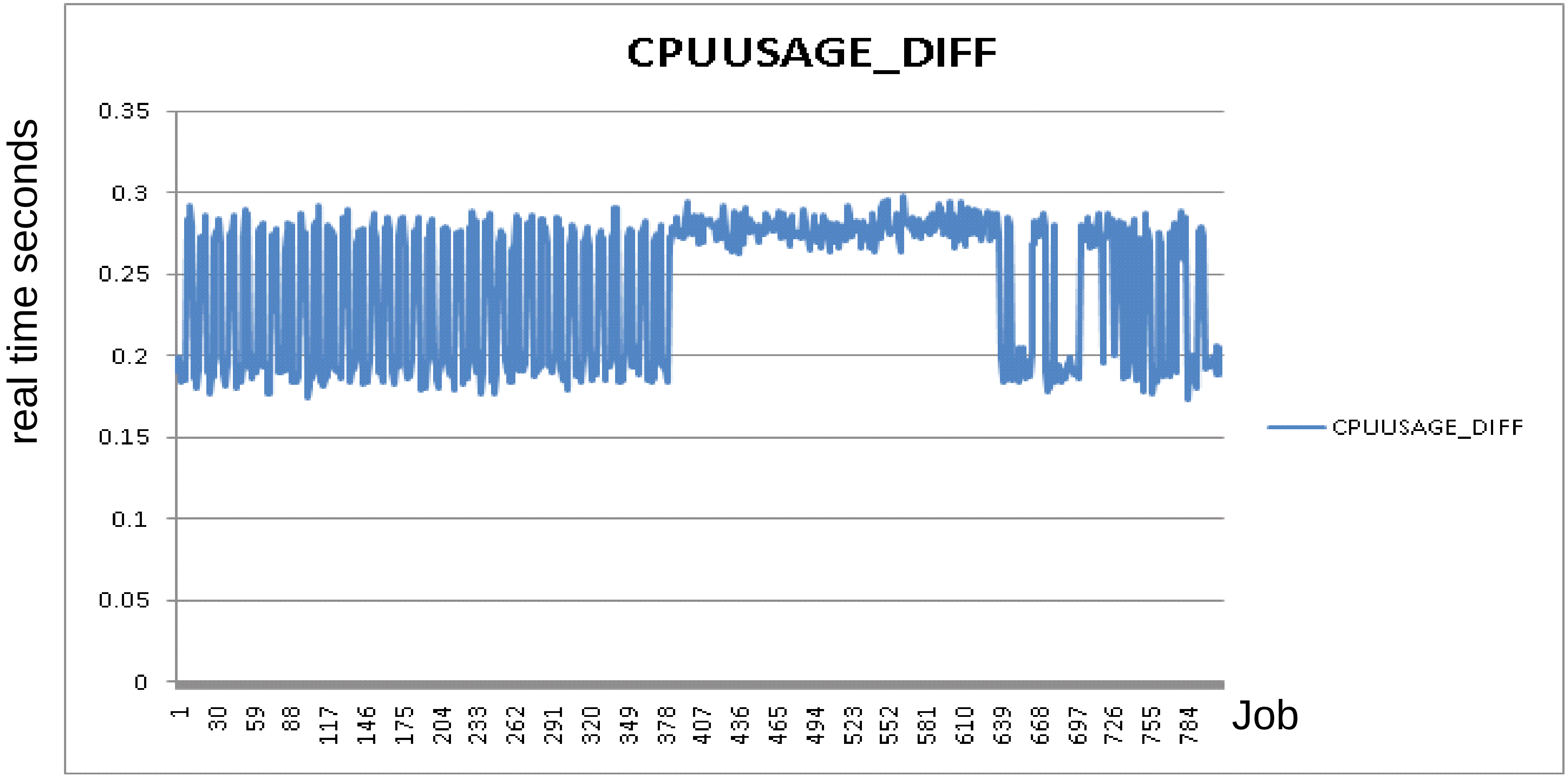
	A	B	C	D	E	F	G	H	I	J
1	LOCALJOBID	WNHOSTNAME	OWNERDN	LSF_CPUUSAGE	WRAPPER_CPUUSAGE	CPUUSAGE_DIFF	LSF_WALLTIME	WRAPPER_WALLTIME	WALLTIME_DIFF	LSF_CPUFA
2	664664	lxb8394	dteam066	2.53	2.34	0.19	80	79	1	
3	664666	lxb8420	dteam066	2.51	2.31	0.2	75	74	1	
4	664667	lxb8393	dteam066	2.53	2.33	0.2	107	105	2	
5	664668	lxb8394	dteam066	2.53	2.35	0.18	99	98	1	
6	664669	lxb8419	dteam066	2.52	2.33	0.19	99	98	1	
7	664670	lxb8446	dteam066	2.52	2.33	0.19	99	98	1	
8	664673	lxb8365	dteam066	2.52	2.33	0.19	101	100	1	
9	664676	lxb8445	dteam066	2.52	2.33	0.19	107	106	1	
10	664678	lxb6832	dteam066	3.86	3.58	0.28	99	98	1	
11	664679	lxb6321	dteam066	3.9	3.63	0.27	100	99	1	
12	664681	lxb6322	dteam066	3.9	3.61	0.29	107	105	2	
13	664682	lxb6323	dteam066	4.03	3.74	0.29	101	100	1	
14	664683	lxb6310	dteam066	3.87	3.59	0.28	99	98	1	
15	664687	lxb8394	dteam066	2.51	2.32	0.19	128	122	6	
16	664689	lxb8419	dteam066	2.61	2.42	0.19	128	122	6	
17	664694	lxb8446	dteam066	2.62	2.43	0.19	120	118	2	
18	664696	lxb8365	dteam066	2.52	2.34	0.18	128	122	6	
19	664699	lxb8445	dteam066	2.53	2.34	0.19	126	124	2	
20	664701	lxb8393	dteam066	2.53	2.34	0.19	124	122	2	
21	664702	lxb6832	dteam066	3.88	3.61	0.27	125	122	3	
22	664708	lxb6321	dteam066	3.9	3.63	0.27	128	121	7	
23	664709	lxb6323	dteam066	3.88	3.61	0.27	124	121	3	
24	664711	lxb6310	dteam066	4.05	3.78	0.27	128	121	7	
25	664713	lxb6322	dteam066	4.04	3.75	0.29	128	121	7	
26	664719	lxb8394	dteam066	2.51	2.32	0.19	94	93	1	
27	664720	lxb8419	dteam066	2.52	2.32	0.2	90	89	1	
28	664722	lxb8446	dteam066	2.52	2.34	0.18	90	89	1	
29	664728	lxb8365	dteam066	2.64	2.45	0.19	91	90	1	
30	664729	lxb8445	dteam066	2.61	2.42	0.19	91	90	1	
31	664732	lxb8393	dteam066	2.53	2.34	0.19	93	89	4	
32	664736	lxb6832	dteam066	3.88	3.61	0.27	90	89	1	
33	664737	lxb6310	dteam066	3.91	3.63	0.28	89	88	1	

Sheet1

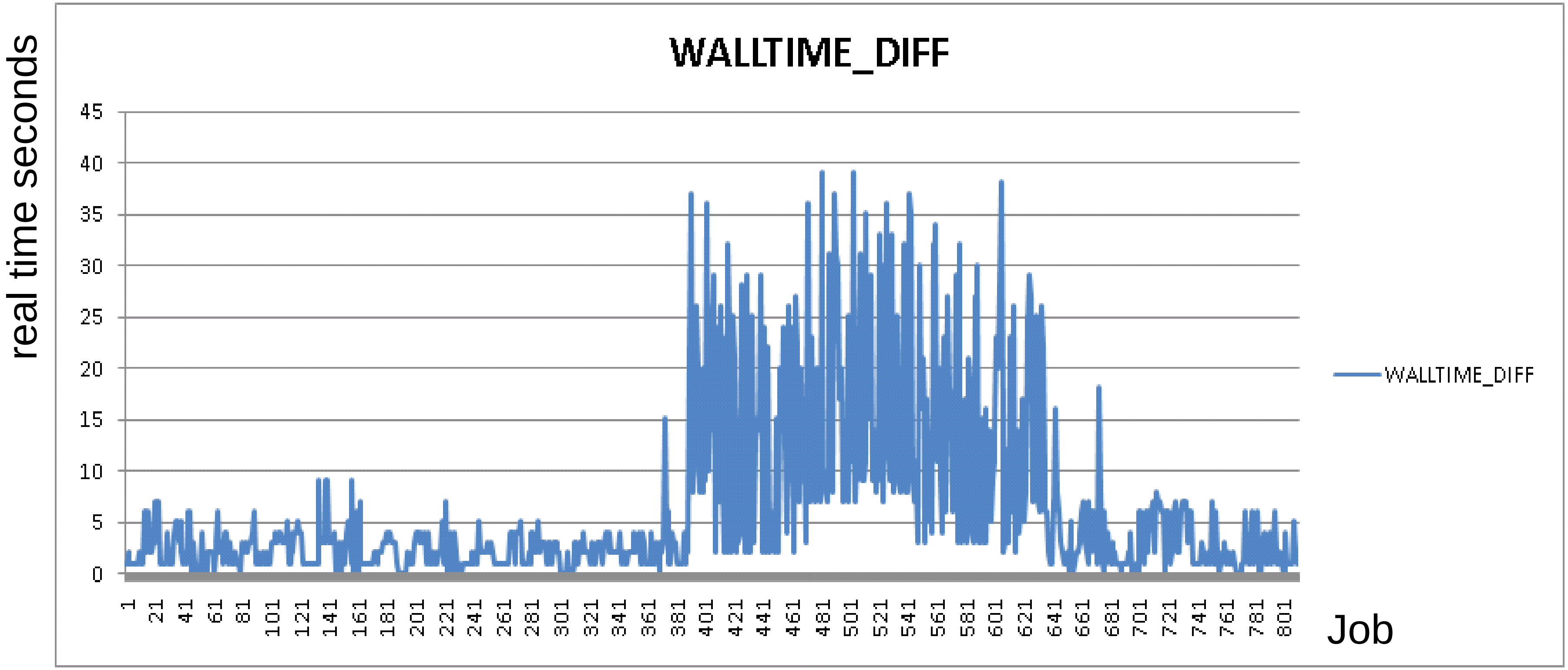
Sheet 1 / 1 | Default | 100% | STD | * | Sum=0

Preliminary!

Example: (preliminary) LSF CPU overhead measurement



Example: (preliminary) LSF WALL overhead measurement



Preliminary !