Common logging format

Aleš Křenek, Mirek Ruda, Zdeněk Šustr

Overview

- Intentionally subjective view
- Comments to SCG document
  - good starting point
  - what may turn to be a problem with current L&B
  - what is not covered by the draft
- "L&B over syslog"
  - another utilization of good syslog information
  - critical issues for machine reading – good and bad practice

L&B over syslog

New job types in L&B

- recently L&B was extended to trace also PBS and Condor jobs
- no code instrumentation – parsing existing logs only
- no direct impact on the rest of gLite
- **valuable experience with parsing various log files**

Foreseen role in gLite

- promissing approach
- enrich current L&B functionality in gLite (get more info)
- additional "gLite job types" (file transfers, advance reservations, . . .)

Self-contained messages

Global identifiers

- jobid (L&B, Condor, PBS), file-transfer id, ...
- any message related to a globally-identified entity should carry the identifier
- PBS almost optimal, Condor rather bad
- missing in the "jss-serv" example in SCG doc :-(

The little context the better

- one occurence of eg. session-jobid relationship is sufficient in theory
- message processing is more complicated (context-sensitive)

Message interrelation

A resubmitted job

- PBS server log

```
17:19:44;Server@erebor;Job;302.erebor;Job Run on hosts odin
17:19:44;Server@erebor;Job;302.erebor;Job Modified
17:20:14;Server@erebor;Job;302.erebor;Obit received
17:20:14;Server@erebor;Job;302.erebor;Job Modified
17:20:14;Server@erebor;Job;302.erebor;Job Run on hosts odin
```

- PBS mom log (at host odin)

```
17:19:44;pbs_mom;Job;302;Started, pid = 32261
17:19:53;pbs_mom;Job;302;kill_job
17:20:14;pbs_mom;Job;302;Obit sent
17:20:15;pbs_mom;Job;302;Started, pid = 32372
17:20:30;pbs_mom;Job;302;task 00000001 terminated
17:20:30;pbs_mom;Job;302;Terminated
```

Message interrelation (2)

The issue

- clocks are not be synchronized
- message delivery (both syslog and L&B) is delayed
- matching the two attempts is still possible but it requires non-trivial semantical knowledge
- neither session or request identifier helps
  - it's still the same request at PBS server

Suggestion

- unique identifier of the transaction process SHOULD be generated
- both invoking and invoked services SHOULD log the transaction identifier
- required for both call-return and passing over the control

Multi-line events

Condor's "log tranactions"

```
105
103 5.0 ClaimId "<147.251.3.18:60837>#1171370638#1"
103 5.0 LastJobLeaseRenewal 1171370942
103 5.0 RemoteHost "erebor.ics.muni.cz"
103 5.0 RemoteVirtualMachineID 1
103 5.0 ShadowBday 1171370942
103 5.0 JobStartDate 1171370942
103 5.0 JobCurrentStartDate 1171370942
103 5.0 JobRunCount 1
106
```

- 105/106 are explicit begin/end transaction markers

Multi-line events (2)

Pro's and contra's

- \+ better human readability while still machine readable
- \+ overcome 1024 bytes limit, even for a single "name=value"
- – requires reliable and ordered message delivery
- – more complicated machine parsing
  - – begin/end transaction markers are
  - – transaction identifier (if more can be opened by single process)

Suggestion

- multi-line events should be supported

What to avoid

Indistinguishable messages

- `01/23/2007 17:20:14;Server@erebor;Job;302.erebor;Obit received`
- indicates intra-deamon communication
- the communicion carries useful info (eg. job terminated or failed)
- the log line is rather useless

Missing information

- job resubmit in PBS, important point in job life
- PBS server does not log any job activity
- resubmit is guessed from another match by PBS scheduler

For-human-only logs

- most Condor logs
- job-related messages without explicit jobid binding

Error propagation and recovery

Error handling

- failure in lower software layer causes failure in higher layers
- any layer can recover
  - whole request does not fail
  - it is still worth to log the error details

How to log

- gather details in calls and log at topmost level only
  - format of the chained error should be specified
- log details immediately, return only eg. errno code
  - explicit interrelation of individual messages must be present

Suggestion

- both approaches should be supported

Logging verbosity vs. performance

L&B megajob

- 1 M jobs/day $\approx$ 11 jobs/sec
- one job $\approx$ 10 events
  - each is an autonomous request for L&B
  - changing internal state
  - hence several syslog lines
- alltogether several hundreds messages per second

Suggestion

- number of messages in full service load should be considered
- configurability is needed
  - filtering on severity is not sufficient
  - another dimension: configure also **what** is logged

Miscelaneous comments

Message format

- more escaped characters
  - usual `\n`, `\t`, `\ddd` (octal digits)
- non-ascii charsets can occur eg. in certificate DN
- long messages
  - eg. JDL can span several kB, it may be still worth logging
  - 1024 bytes limit

Log info sources

- even "normal" user commands (unprivileged, non-service)
- not only DAEMON facility

Miscelaneous comments (2)

MUST and SHOULD

- for L&B any incoming event "modify internal state"
- routine operation – should not be logged higher than DEBUG

Message aggregation

- silly but illustrative example:

  ```
  2007-03-09T11:12:13 user/password submitted job XX
  ```

- "highest severity" requirement makes sense
- insensitive part with its normal severity should be logged too
  - can be routed to another log file where it is expected to appear

Conclusions

Summary

- SCG document rewieved thoroughly
  - well written, only minor comments
  - good starting point for whole JRA1
- L&B and JP specific view
- additional usage of syslog data foreseen
  - L&B may start reading your logs tomorrow
  - be kind to us and we shall pay back to you

Further steps

- similar subjective review should come from others too
- comments to be reflected in the draft – who will do it?
- project-wide common library – is it realistic?