Use of common session ids for tracking end user requests in SRM/dCache

Prepared by A. Baranovski for CEDPS

dCache

- The dCache stores, retrieves and managing petabytes of data
- The data is distributed among a large number of heterogeneous server nodes.
- dCache supports a variety of management and access protocols
  - gridftp,srm, dccp, xrootd,
- All protocols observe common virtual filesystem tree.
- The dCache functionality is to combine separate disk storage systems of several hundred tera bytes into a uniformly accessible filesystem tree.
  - To make this process manageable, dCache does load balancing, detects failing hardware and replicated files.

Troubleshooting

- The dCache is a distributed system
  - Consists of pools , doors, admin , name and routing domain processes scattered over several machines
- In such a setup, hardware or configuration problems are difficult to track
  - Specifically if they are highly correlated to the context of a particular user request.
- Problem: How to trace end user request to dcache in all log files written by the system (dCache + SRM)
  - Log files are scattered
  - Log records only contain atomic context relevant to the code that produced the output

Common session ids

- Instrument dCache /SRM message infrastructures with common session identifiers.
- Session id is generated at the time user requests something from SRM or dCache
- All dCache/SRM servers do processing preserving session id
- Use log4j to print out session id in each log message.
- Use session id can reconstruct and troubleshoot distributed dCache processes in the context of a particular user request
  - i.e. track previously disjoint inter component message communication scenarios

Log4j (concepts)

- Loggers
  - dCache classes have loggers which inherit from common ROOT
    - ROOT logger policies propagate to other loggers unless overridden
- Appenders
  - an output destination for each logger
- Pattern Layout
  - Customize output format for each appender
- http://logging.apache.org/log4j/1.2/manual.html

Log4j configuration in dCache

- Log4j is available in dCache 1.9.x
  - etc/logConfig.xml, srm-webapp/WEB-INF/logConfig.xml
  - <**appender** name="detailed" class="org.apache.log4j.ConsoleAppender">
  - <**layout** class="org.apache.log4j.PatternLayout">
  - <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss.SSS} (%X{cells.cell}) [%x] %l %p - %m%n"/>
  - </layout>
  - </appender
  - <category name="logger.org.dcache">
  - <priority value="ALL"/> <appender-ref ref="detailed"/>
  - </category>
- logger.org.dcache common name for all dCache specific class loggers
- org.apache.log4j.ConsoleAppender – output to console, commonly found in /var/log/<processname>.log
- ALL – debug level. Other possible values are : INFO, DEBUG, WARN, OFF, ERROR

Patterns

- log message pattern is configured using layout object
  - <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss.SSS} %X{cells.session} %X{srm.session} [%x] %l %p - %m%n"/>
- %X{cells.session} %X{srm.session} – session id printout pattern
  - %X instructs pattern layout manager to use log4j MDC context . This context is managed by session id propagation code
  - cells.session –use this string to find dCache session id value in the MDC context
  - srm.session - use this string to find SRM session id value in the MDC context
  - cells.session is built from srm.session using data specific to individual file requests
- %x – additional, nested debug information which may exist at a time log message is printed. This optional information is also session id specific.

Examples

- Simulate error condition: copy a file that exists
- SRM logs: query pnsfDomain to check if file exists
  - 2009-03-03 12:17:21.093 (**v2:srmCopy:115651740:-2147438648:-2147438647**) dmg.cells.nucleus.CellNucleus.log(CellNucleus.java:884) ERROR - org.dcache.srm.scheduler.NonFatalJobFailure: org.dcache.srm.SRMException: local copy **failed with code =1** details: copy failed:CacheException(rc=10008;msg=File exists)
- dCache logs: pnsfDomain is queried – file found
  - 03 Mar 2009 12:17:21 (PnfsManager) (**v2:srmCopy:115651740:-2147438648:-2147438647**)) getFileMetaData of 000100000000000000002028 -> [-rw-r--r--;500;500][c=02.05-12:21:49;m=02.05-12:21:49;a=02.05-12:21:49]
- You can grep all log files based on:
  - "-2147438648" – srm group request id
  - v2:srmCopy – operation used
  - "-2147438647" – individual file request

Conclusions

- New logging gives more control over format and granularity of the dCache log messages
- You can use cells.session,srm.session,cells.cell, cells.domain to add user specific information to each log message
- cells.session has nested structure for srm requests
- One grep command can be used to extract entire user workflow from all dCache log files

References

- http://www.dcache.org
- http://www.dcache.org/manuals/Book/
- http://logging.apache.org/log4j/1.2/manual.html
- /opt/d-cache/etc/logConfig.xml
- /opt/d-cache/srm-webapp/WEB-INF/logConfig.xml
- /var/log/<dCacheProcess>.log
- /opt/d-cache/libexec/apache-tomcat-5.5.20/logs/