Analysis of Chukwa 0.4.0 and Chukwa 0.6.0

Relax Recovery method.

1. **A summary of the most noticeable differences between the two versions according to the recovered architectural views.**

In version 0.4.0 there are 3 major clusters named src, build and contrib but in version 0.6.0 build is merged with src and there are 2 clusters named src and contrib.

Build cluster in 0.4.0 contains sub clusters named HICC / chukwa and collector which are IO objects which intern contains WEB_INF sub-clusters which contains topics like DumpChunks, DumpRecord, Filter, PidFile, Deamon watcher, cluster-configuration etc where as in 0.6.0 all these topics are merged with src cluster.

In 0.6.0 separate cluster named Data trigger which is a networking object is added, that is not present separately in 0.4.0 this cluster contains methods for trigger event, trigger action and HTTP trigger action.

For displaying data, Bean which is a Graphic object and Rest which is IO object are added to HICC in 0.6.0 where as in 0.4.0 they are not present under HICC.

In 0.4.0 networking object under Rest cluster had 16 components where as in 0.6.0 it is reduced to 1.

2. **Why does this recovery method show these changes and not others?**

RELAX recovery method show these changes because it is Semantic based and it does classification by taking code entity of java file and determines affinity to pre-defined classes using classifier which is trained with word list of certain topics like security, input, networking etc, then determine based on the threshold to which cluster they belong and it gives Hierarchical view, dependencies whereas other recovery doesn't give clusters based on classifiers.

3. **Do you think the differences accurately reflect the changes in the system that have taken place between the two versions?**

Yes, but in some cases like Long term trending visualization might not work for large scale data.

4. **Based on the results of the recovery, what conclusions would you draw about the future of the system?**

Based on the result of recovery, collector clusters which are deprecated and merged with src cluster makes the system read and write more efficiently all these changes shows improved performance with low latency.

ACDC Recovery method.

1. **A summary of the most noticeable differences between the two versions according to the recovered architectural views.**

In 0.4.0 it contains writerException clusters under writer data collection/ localfs sub system where as in 0.6.0 it is moved to datacollection/connector sub system.

In 0.6.0 there is sub system "SyslogAdaptor$FacilityType",”PipelineableWriter”,” Annotation” added as part of data collection adaptor which is not present in 0.4.0.

In 0.6.0 there is "PipelineConnector" sub system added to connector which is not present in 0.4.0

security utils are added to 0.6.0 and there is dependency on security topics such as "security.UserGroupInformation", "security.SecurityUtil" etc where as in 0.4.0 there is no dependencies on security utils.

In 0.6.0 HBase writers has been added which are not present in 0.4.0 like "HBaseWriter$StatReportingTask",” HBaseWriter”, “HBaseWriter$1”, “HConstants” etc.

2. **Why does this recovery method show these changes and not others?**

ACDC recovery method generates output based on structure and it is also using common programming patterns. It creates clusters based on patterns that commonly occur in manual decomposition of software system, identifies subsystems using a collection of criteria that are based on connections between entities, then names the sub systems and creates clusters. It generates clusters and dependencies files and human readable files.

3. **Do you think the differences accurately reflect the changes in the system that have taken place between the two versions?**

Yes, because removal of collectors and adding HBase write directly by chukwa agents reduces latency time and increases the performance.

4. **Based on the results of the recovery, what conclusions would you draw about the future of the system?**

Based on the result of recovery in 0.6.0 security utils are added hence security vulnerability is audited which is very much essential for the system and it helps the system to be more secure in further releases.

ARC Recovery Method

1. **A summary of the most noticeable differences between the two versions according to the recovered architectural views.**

In 0.6.0 “ChukwaDailyRollingFileAppender”(IO object) is depended on “RecordConstants” and “org.apache.hadoop.chukwa.” but in 0.4.0 “ChukwaDailyRollingFileAppender”(networking object) is depended on “chukwa.inputtools” and “org.apache.hadoop.chukwa”

Component "MetricDataLoaderPool" which is a IO object in 0.4.0 is depended on "org.apache.hadoop.chukwa" networking object. this "MetricDataLoaderPool" is removed from 0.6.0 version.

In Chukwa 0.4.0 there are 4 primary components (Agent, collector, Map reduce job, HICC) where as in 0.6.0 there are 5 primary components (Adaptor, Agent, ETL processes, Data analytics scripts, HICC)

In addition to this, 0.6.0 version contains Demux processes like “demux.processor.mapper.DatanodeProcessor$1”,” JobTrackerProcessor” which are not present in 0.4.0 version.

2. **Why does this recovery method show these changes and not others?**

ARC recovery method applies topic modeling to the source code. For each source file it then determines which topic it is most aligned with and groups the source files into clusters for that topic. ARC method gives dependencies between different components.

3. **Do you think the differences accurately reflect the changes in the system that have taken place between the two versions?**

Yes, some IO objects are removed like collectors and networking objects are added which improves the performance of the system.

4. **Based on the results of the recovery, what conclusions would you draw about the future of the system?**

In 0.4.0 Data is sent across the network to collector which intern writes to HDFS. Map reducing jobs (jobs Archiving and Demux) are used for organizing and processing incoming data. Where as in 0.6.0 collectors are deprecated and Chukwa agents write directly to HBase and data nodes which is better for continuous monitoring of data stream and periodically produce reports. Data analytics scripts are used for aggregating the data stored in HBase and provides visualization and interpretation. All these changes will help the system to be improve the performance with latency time in further releases.