Apache Chukwa

Apache Chukwa moved into the Attic in 2020-05. Apache Chukwa mission was Open source data collection system for monitoring large distributed systems.

The website, downloads and issue tracker all remain open, though the issue tracker is read-only. See the website at **http://chukwa.apache.org** for more information on Chukwa.

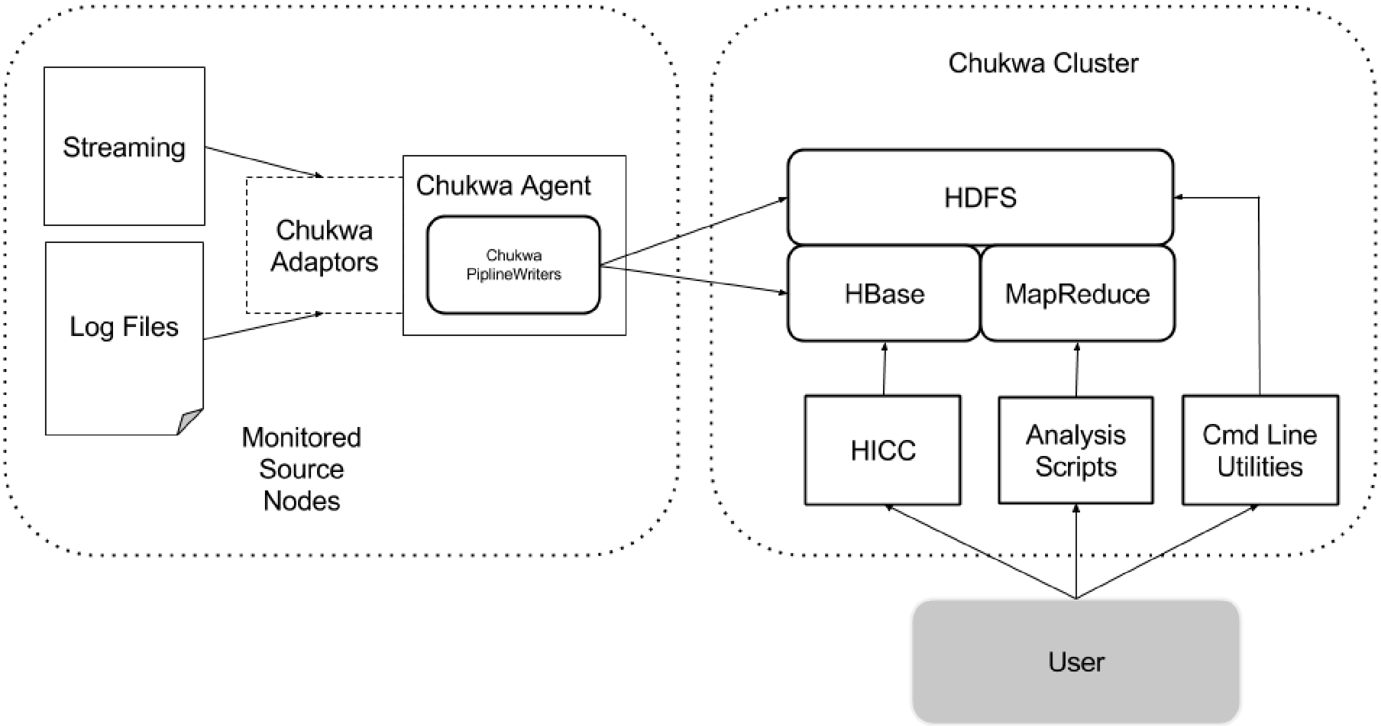
Introduction

Apache Chukwa aims to provide a flexible and powerful platform for distributed data collection and rapid data processing. Our goal is to produce a system that's usable today, but that can be modified to take advantage of newer storage technologies (HDFS appends, HBase, etc) as they mature. In order to maintain this flexibility, Apache Chukwa is structured as a pipeline of collection and processing stages, with clean and narrow interfaces between stages. This will facilitate future innovation without breaking existing code.

Apache Chukwa has five primary components:

- **Adaptors** that collect data from various data source.
- **Agents** that run on each machine and emit data.
- **ETL Processes** for parsing and archiving the data.
- **Data Analytics Scripts** for aggregate Hadoop cluster health.
- **HICC**, the Hadoop Infrastructure Care Center; a web-portal style interface for displaying data.

Below is a figure showing Apache Chukwa data pipeline, annotated with data dwell times at each stage. A more detailed figure is available at the end of this document.



Agents and Adaptors

Apache Chukwa agents do not collect some particular fixed set of data. Rather, they support dynamically starting and stopping *Adaptors*, which small dynamically-controllable modules that run inside the Agent process and are responsible for the actual collection of data.

These dynamically controllable data sources are called adaptors, since they generally are wrapping some other data source, such as a file or a Unix command-line tool. Apache Chukwa agent guide includes an up-to-date list of available Adaptors.

Data sources need to be dynamically controllable because the particular data being collected from a machine changes over time, and varies from machine to machine. For example, as Hadoop tasks start and stop, different log files must be monitored. We might want to increase our collection rate if we detect anomalies. And of course, it makes no sense to collect Hadoop metrics on an NFS server.

ETL Processes

Apache Chukwa Agents can write data directly to HBase or sequence files. This is convenient for rapidly getting data committed to stable storage.

HBase provides index by primary key, and manage data compaction. It is better for continous monitoring of data stream, and periodically produce reports.

HDFS provides better throughput for working with large volume of data. It is more suitable for one time research analysis job . But it's less convenient for finding particular data items. As a result, Apache Chukwa has a toolbox of MapReduce jobs for organizing and processing incoming data.

These jobs come in two kinds: *Archiving* and *Demux*. The archiving jobs simply take Chunks from their input, and output new sequence files of Chunks, ordered and grouped. They do no parsing or modification of the contents. (There are several different archiving jobs, that differ in precisely how they group the data.)

Demux, in contrast, take Chunks as input and parse them to produce ChukwaRecords, which are sets of key-value pairs. Demux can run as a MapReduce job or as part of HBaseWriter.

For details on controlling this part of the pipeline, see the Pipeline guide. For details about the file formats, and how to use the collected data, see the Programming guide.

Data Analytics Scripts

Data stored in HBase are aggregated by data analytic scripts to provide visualization and interpretation of health of Hadoop cluster. Data analytics scripts are written in PigLatin, the high level language provides easy to understand programming examples for data analyst to create additional scripts to visualize data on HICC.

HICC

HICC, the Hadoop Infrastructure Care Center is a web-portal style interface for displaying data. Data is fetched from HBase, which in turn is populated by collector or data analytic scripts that runs on the collected data, after Demux. The Administration guide has details on setting up HICC.

Apache HBase Integration

Apache Chukwa has adopted to use HBase to ensure data arrival in milli-seconds and also make data available to down steam application at the same time. This will enable monitoring application to have near realtime view as soon as data are arriving in the system. The file rolling, archiving are replaced by HBase Region Server minor and major compactions.