

# Chukwa

撰寫人：黃俊堯 ([allen501pc@gmail.com](mailto:allen501pc@gmail.com))

發布日期：2011 年 5 月 21 日

版本號：3

## 簡介

MapReduce 支援 Log processing。但不幸的是，由於 Hadoop 是一個叢集式系統，跨多台機器的大量 Log processing 讓原本 Hadoop 的效率變慢，而且 HDFS 也不能支援一致性儲存這些 Log，只能在各自的子系統中自行維護 Log。

Chukwa 專案發起的主要目標是對於分散式資料蒐集以及快速的資料處理問題，能夠提供一個彈性且強力的平台。

為了維護的彈性需求，Chukwa 採用計算機組織提到的[管線設計](#)與處理 stage，示意圖如圖 1。

Chukwa 有四個主要物件，第 2 頁會各自詳述：

1. **Agents**：在各自機器上作資料的發送。
2. **Collectors**：從 agent 上接收資料並存入儲存體如 HDFS、HBase 等。
3. **MapReduce jobs**：利用 MapReduce 傳遞並將資料作歸檔([archive](#))。
4. **HICC**：Hadoop Infrastructure Care Center，利用 web 介面展示資料。

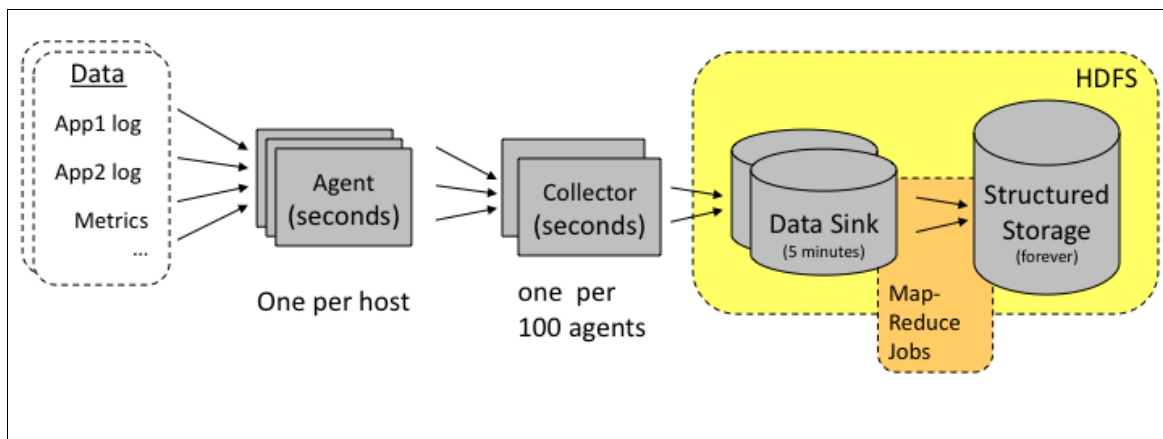


圖 1: Chukwa Pipeline 架構

## Agents and Adaptors

Chukwa agent 支援動態啟動 Adaptors。這裡的 Adaptor 是一個在 Agent 程序內的負責資料回應的小型動態控制模組。不同的資料來源(如 file 或 Unix 命令列工具等)分別對應至不同的 Adaptors。

因為這些特殊的資料如 Log 會隨著時間流逝、機器之間的資料傳遞而變化，因此這裡特別強調資料的動態控制。例如，Hadoop 在啟動與停止 task 時，必須監視 tasks 產生的各個紀錄檔，我們可能要增加資料蒐集速度以即早發現異常狀況。

## Chukwa 的資料模型

Chukwa 透過 **chunk** 發送資料。**chunk** 是一個位元組序列(byte sequence)，包含了一些 metadata。而這些 metadata 欄位通常會由 Agent 或 Adaptor 自動幫您設定好。只是使用者還必須設定兩個欄位：

- cluster : cluster name 的參數設定在 conf/chukwa-env.sh 裡，它是 Agents 所需要的共用參數設定。
- datatype : 用來描述由 Adaptor 蒐集的資料格式。

表 1 為 Chukwa chunk 的 metadata 格式。

表 1: chunk meata 參數表

欄位名稱	敘述	設定方式
Source	標記 Chunk 產生的來源位置	自動設定
Cluster	Cluster 的描述	使用者設定
Datatype	輸出格式	使用者設定
Sequence ID	Chunk 的 Offset ID，同時也表示每個 Adaptor 已經發送了多少 bytes。 <sup>註 1</sup>	由 adaptor 自動設定，起始編號為 0。
Name	來源資料的名稱	由 Adaptor 自動設定

註 1：若今天有 Adaptor 發送了 100 bytes 的 Chunk，那麼它的 Sequence Id 就是 100。之後再次發送 100 bytes 的 Chunk，則 Sequence Id 為 200，以此類推。此機制與 TCP 協定相差無幾。因此，Sequence Id 可用來防止 crash 後重複派發資料以進行復原重建的情況。

## Collectors

資料蒐集器負責蒐集資料並存進 HDFS。每個蒐集器可能會透過 HTTP 協定從上百個 host 端蒐集資料，並寫入一個名為 **sink file** 的單一檔案，此檔案是已序列化過後的 Chunks 序列檔案。資料蒐集器會週期性地將 **sink file** 關閉、重新命名並標記為有效檔，然後再開啟另一個 **sink file** 檔案繼續寫入。

因此，透過資料蒐集器，可以大大地減少 Chukwa 產生的 HDFS 檔案。另外，資料蒐集器隱藏 HDFS 的使用細節，甚至是 Hadoop 版本，以簡化設定。這使得 Chukwa 能夠監督不同 Hadoop 版本開發的叢集式系統環境，甚至是非 Hadoop 環境也可以。

## MapReduce processing

雖然說資料蒐集器可以幫您方便作資料蒐集與存放的工作，但是不能幫您作資料分析與搜尋特定項目。因此，Chukwa 提供了一套基於 MapReduce Job 的工具箱，讓您可以方便開發。

這些 MapReduce Job 分為兩類：

1. Archiving Job：接收 Chunk 資料後，將資料作排序、序列化、群組歸檔後，輸出為 Chunk 序列。此動作不會更改資料。
2. Demux Job：接收 Chunk 資料後，輸出一個名為 ChukwaRecords 的 key-value 配對集合。

## HICC(Hadoop Infrastructure Care Center)

HICC 透過 Web 介面展示資料。而資料(由 MapReduce 的 Demux 程序產生)的獲取須透過 MySQL 資料庫來擷取。Chukwa 整體架構如圖 2。

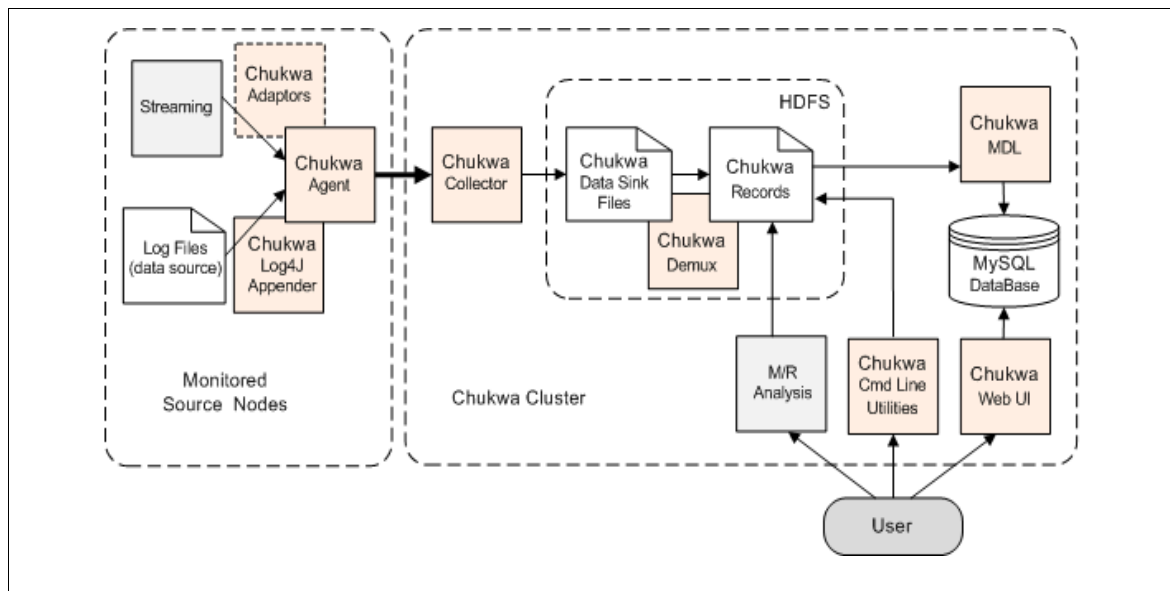


圖 2: Chukwa 整體架構

根據上圖 2，我們可知一個最小的 Chukwa 由三個組件組成：

- 運行 Chukwa 的 Hadoop 叢集系統將儲存資料
- 資料蒐集器負責將資料寫入 HDFS
- 一個或多個 Agent 同時運行，負責發送並監督資料給資料蒐集器。而運行 Agent 的運算節點則被 Chukwa 歸類為監督中的資料來源節點。

另外，您或許會想要自行設計 Chukwa Demux Job，用來剖析資料；或者是 HICC。

## 系統需求

- Chukwa (目前穩定版為 0.4)
- Hadoop 0.20.1+
- MySQL 5.1+
- 系統狀態監看工具 [sysstat](#)

## 安裝 Chukwa

Chukwa 至少包含了三個主要元件：

- Hadoop 叢集系統，用來儲存 Chukwa 所需資料。
- Collector 程序，將資料寫入 HDFS。
- 一個或多個的 Agent 程序，負責送監聽的資料給 Collector。

## 安裝步驟

從 [Chukwa 下載頁面](#)，抓取最新且穩定的 Chukwa。

如果下載下來的檔案為 chukwa-0.4.0.tar.gz，則輸入以下指令解壓縮：

```
cd /opt/  
tar xzf chukwa-0.4.0.tar.gz
```

把 chukwa 資料夾複製給其他要被監督的 node。

註：此後寫的 **CHUKWA\_HOME** 代表包含 Chukwa 程式的資料夾。

預先處理的步驟

我們需要先安裝 sysstat，備份 hadoop 設定檔，並將相關 jar 檔案複製到 **hadoop/lib** 底下

```
$ sudo apt-get install sysstat  
$ cd /opt/hadoop  
$ cp /opt/chukwa/conf/hadoop-metrics.properties.template conf/  
$ cp conf/hadoop-metrics.properties conf/hadoop-metrics  
$ cp /opt/chukwa/chukwa-hadoop-0.4.0-client.jar ./lib/  
$ cp /opt/chukwa/tools-0.4.0.jar ./lib/
```

## Chukwa 一般設定

進入 **chukwa/conf** 底下，並做以下設定：

編輯 alert 檔案

```
allen501pc@gmail.com
```

編輯 chukwa-collector-conf.xml

```
<property>  
  <name>writer.hdfs.filesystem</name>  
  <value>hdfs://cloud1:9000/</value>  
  <description>HDFS to dump to</description>  
</property>  
<property>  
  <name>chukwaCollector.outputDir</name>  
  <value>/chukwa/logs/</value>  
  <description>Chukwa data sink directory</description>  
</property>  
<property>  
  <name>chukwaCollector.http.port</name>  
  <value>8080</value>  
  <description>The HTTP port number the collector will listen on</description>  
</property>
```

編輯 jdbc.conf (這裡的 cluster name 為 cloud)

```
cloud=jdbc:mysql://cloud1:3306/cloud?user=root
```

編輯 nagios.properties

```
log4j.appen          =dout$1      o          H          .
```

編輯 chukwa-env.sh

```
export JAVA_HOME=/usr/lib/jvm/java-6-sun
export HADOOP_HOME="/opt/hadoop"
export HADOOP_CONF_DIR="/opt/hadoop/conf"
```

編輯 agents

```
cloud1
cloud2
```

將 chukwa-agent-conf.xml.template 重新命名為 chukwa-agent-conf.xml ，編輯 chukwa-agent-conf.xml

```
<property>
  <name>chukwaAgent.tags</name>
  <value>cluster="cloud"</value>
  <description>The cluster's name for this agent</description>
</property>
<property>
  <name>chukwaAgent.hostname</name>
  <value>localhost</value>
  <description>The hostname of the agent on this node. Usually localhost, this is used by
the chukwa instrumentation agent-control interface library</description>
</property>
```

編輯 collectors

```
cloud1
cloud2
```

將 initial\_adaptors.template 改名為 initial\_adaptors

複製 CHUKWA\_HOME/conf/hadoop-log4j.properties 到  
HADOOP\_HOME/conf/log4j.properties

複製 CHUKWA\_HOME/conf/hadoop-metrics.properties 到  
HADOOP\_HOME/conf/hadoop-metrics.properties ，並將@CHUKWA\_LOG@改為你  
在 chukwa\_env.sh 設定的 log 資料夾

**Hbase 相關設定部分**

根據 <http://sites.google.com/site/waue0920/Home/hbase/hbase-cong-ji-an-zhuang> 這篇，  
設定 HBase 與 Zookeeper

到 <http://svn.apache.org/viewvc?view=revision&revision=1062530> 抓取  
hbase.schema。

執行, HBASE\_HOME/bin/hbase shell 後，將 hbase.schema 的 create table 部分一一建立。

目前執行結果

執行順序(重要，第 1,2 步不能顛倒，否則系統會報錯。錯誤訊息如後所述)

1. 啟動 chukwa，CHUKWA\_HOME/bin/start-all.sh
2. 啟動 hadoop，HADOOP\_HOME/bin/start-all.sh
3. 啟動 HBase，HBASE\_HOME/bin/start-hbase.sh
4. 啟動 chukwa hicc，CHUKWA\_HOME/bin/chukwa hicc

已知 Bug

1. Chukwa 0.4 版並非完全支援 MySQL，縱然抓取 0.3 版的 dbAdmin.sh，安裝 MySQL Master-Slave，使其強制支援 MySQL，仍然無法成功啟動。
2. CHUKWA\_HOME/bin/stop-all.sh 無法完全關閉 agent 與 collector。必須一一至 CHUKWA\_PID\_DIR 抓取相關 pid 值，執行 kill command 方可完全清除。(問題仍存在)
3. HICC 仍無法顯示抓取到的資料。(問題仍存在)

按照威宇學長所說的，以下嘗試使用單機安裝模式：

將 Hadoop 運作模式改為單機模式。

Chukwa 單機安裝步驟與上述 distributed mode 安裝方式大致相同，唯獨注意以下。

chukwa-collector-conf.xml

```
<property>
  <name>writer.hdfs.filesystem</name>
  <value>hdfs://localhost:9000/</value>
  <description>HDFS to dump to</description>
</property>
<property>
  <name>chukwaCollector.outputDir</name>
  <value>/chukwa/logs/</value>
  <description>Chukwa data sink directory</description>
</property>
<property>
  <name>chukwaCollector.http.port</name>
  <value>8080</value>
  <description>The HTTP port number the collector will listen on</description>
</property>
```

編輯 jdbc.conf (這裡的 cluster name 為 cloud)

```
cloud=jdbc:mysql://localhost:3306/cloud?user=root
```

編輯 nagios.properties

```
log4j.appender.NAGIOS.Host=localhost
```

## 編輯 agents

```
localhost
```

## 編輯 collectors

```
localhost
```

方法 1 :

啟動方式如之前所述。(05/21 再度重測所有方法) , 訊息記錄如下 :

```
allen501pc@localhost:~$ /opt/chukwa/bin/start-all.sh ## 啟動 chukwa
localhost: starting collector, logging to /opt/chukwa/logs/chukwa-chukwa-collector-localhost.out
localhost: starting agent, logging to /opt/chukwa/logs/chukwa-chukwa-agent-localhost.out
starting archive, logging to /opt/chukwa/logs/chukwa-chukwa-archive-localhost.out
starting demux, logging to /opt/chukwa/logs/chukwa-chukwa-demux-localhost.out
starting dp, logging to /opt/chukwa/logs/chukwa-chukwa-dp-localhost.out
#### 目前為止 , 沒有問題 #####
allen501pc@localhost:~$ /opt/hadoop/bin/start-all.sh ## 啟動 hadoop
starting namenode, logging to /opt/hadoop/logs/hadoop-allen501pc-namenode-localhost.out
localhost: starting datanode, logging to /opt/hadoop/logs/hadoop-allen501pc-datanode-localhost.out
localhost: log4j:ERROR cleanUpRegex == null || !cleanUpRegex.contains("$fileName")
localhost: starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-allen501pc-secondarynamenode-localhost.out
localhost: log4j:ERROR cleanUpRegex == null || !cleanUpRegex.contains("$fileName")
starting jobtracker, logging to /opt/hadoop/logs/hadoop-allen501pc-jobtracker-localhost.out
log4j:ERROR cleanUpRegex == null || !cleanUpRegex.contains("$fileName")
localhost: starting tasktracker, logging to /opt/hadoop/logs/hadoop-allen501pc-tasktracker-localhost.out
localhost: log4j:ERROR cleanUpRegex == null || !cleanUpRegex.contains("$fileName")
```

方法 2 :

若是先啟動/opt/chukwa/bin/chukwa collector , 則出現以下訊息(尚未啟動 hicc) :

```
/oallen501pc@localhost:~$ /opt/chukwa/bin/chukwa collector
allen501pc@localhost:~$ /opt 2011-05-21 14:21:39.055::INFO: Logging to STDERR via
org.mortbay.log.StdErrLog
2011-05-21 14:21:39.116::INFO: jetty-6.1.11
```

見解 : 但是根據[5]的說明 , jetty 是由 hicc 喚起。但是似乎很有問題。因為只有啟動 collector , 未啟動 hicc , 何來 jetty 錯誤 ?

方法 3 :

若是先啟動 hadoop , 則訊息記錄如下 :

```
starting namenode, logging to /opt/hadoop/logs/hadoop-allen501pc-namenode-localhost.out
```



```
localhost: starting datanode, logging to /opt/hadoop/logs/hadoop-allen501pc-datanode-localhost.out
localhost: Error initializing ChukwaClient with list of currently registered adaptors, clearing our local list of
adaptors
localhost: starting secondarynamenode, logging to /opt/hadoop/logs/hadoop-allen501pc-
secondarynamenode-localhost.out
localhost: Error initializing ChukwaClient with list of currently registered adaptors, clearing our local list of
adaptors
starting jobtracker, logging to /opt/hadoop/logs/hadoop-allen501pc-jobtracker-localhost.out
Error initializing ChukwaClient with list of currently registered adaptors, clearing our local list of adaptors
localhost: starting tasktracker, logging to /opt/hadoop/logs/hadoop-allen501pc-tasktracker-localhost.out
localhost: Error initializing ChukwaClient with list of currently registered adaptors, clearing our local list of
adaptors
```

見解：綜合[5][6]的結果，我想應該是 chukwa 本身附的 hadoop-metrics.properties 本身有問題，但是官方網站沒解釋 hadoop-metrics.properties 要怎麼修正。

方法 4：

不管錯誤訊息，按照之前所述啟動 chukwa/bin/start-all.sh、hadoop/bin/start-all.sh，chukwa/bin/chukwa hicc。但是顯示如下：

```
allen501pc@localhost:~$ /opt/chukwa/bin/chukwa hicc
allen501pc@localhost:~$ 2011-05-21 14:49:02.593::INFO: Logging to STDERR via
org.mortbay.log.StdErrLog
2011-05-21 14:49:02.756::INFO: jetty-6.1.11
```

## 參考文獻

1. [http://hadoop.apache.org/common/docs/r0.20.2/hadoop\\_arc](http://hadoop.apache.org/common/docs/r0.20.2/hadoop_arc)
2. <http://incubator.apache.org/chukwa/docs/r0.4.0/admin.html#Installing+Chukwa>
3. <http://incubator.apache.org/chukwa/docs/r0.4.0/agent.html>
4. <http://trac.nchc.org.tw/cloud/wiki/waue/2>
5. <http://www.open-c / g o>
6. <http://www.mail-arc -a w k u user@hadoop.apache.org/msg00386.html>