FlumeProposal

Flume - A Distributed Log Collection System

Abstract

Flume is a distributed, reliable, and available system for efficiently collecting, aggregating, and moving large amounts of log data to scalable data storage systems such as Apache Hadoop's HDFS.

Proposal

Flume is a distributed, reliable, and available system for efficiently collecting, aggregating, and moving large amounts of log data from many different sources to a centralized data store. Its main goal is to deliver data from applications to Hadoop’s HDFS. It has a simple and flexible architecture for transporting streaming event data via flume nodes to the data store. It is robust and fault-tolerant with tunable reliability mechanisms that rely upon many failover and recovery mechanisms. The system is centrally configured and allows for intelligent dynamic management. It uses a simple extensible data model that allows for lightweight online analytic applications. It provides a pluggable mechanism by which new sources, destinations, and analytic functions which can be integrated within a Flume pipeline.

Background

Flume was initially developed by Cloudera to enable reliable and simplified collection of log information from many distributed sources. It was later open-sourced by Cloudera on GitHub as an Apache 2.0 licensed project in June 2010. During this time Flume has been formally released five times as versions 0.9.0 (June 2010), 0.9.1 (Aug 2010), 0.9.1u1 (Oct 2010), 0.9.2 (Nov 2010), and 0.9.3 (Feb 2011). These releases are also distributed by Cloudera as source and binaries along with enhancements as part of Cloudera Distribution including Apache Hadoop (CDH).

Rationale

Collecting log information in a data center in a timely, reliable, and efficient manner is a difficult challenge but important because when aggregated and analyzed, log information can yield valuable business insights. We believe that users and operators need a manageable systematic approach for log collection that simplifies the creation, the monitoring, and the administration of reliable log data pipelines. Oftentimes today, this collection is attempted by periodically shipping data in batches and by using potentially unreliable and inefficient ad-hoc methods.

Log data is typically generated in various systems running within a data center that can range from a few machines to hundreds of machines. In aggregate, the data acts like a large-volume continuous stream with contents that can have highly-varied format and highly-varied content. The volume and variety of raw log data makes Apache Hadoop's HDFS file system an ideal storage location before the eventual analysis. Unfortunately, HDFS has limitations with regards to durability as well as scaling limitations when handling a large number of low-bandwidth connections or small files. Similar technical challenges are also suffered when attempting to write data to other data storage services.

Flume addresses these challenges by providing a reliable, scalable, manageable, and extensible solution. It uses a streaming design for capturing and aggregating log information from varied sources in a distributed environment and has centralized management features for minimal configuration and management overhead.

Initial Goals

Flume is currently in its first major release with a considerable number of enhancement requests, tasks, and issues recorded towards its future development. The initial goal of this project will be to continue to build community in the spirit of the "Apache Way", and to address the highly requested features and bug-fixes towards the next dot release.

Some goals include:

- To stand up a sustaining Apache-based community around the Flume codebase.
- Implementing core functionality of a usable highly-available Flume master.
- Performance, usability, and robustness improvements.
- Improving the ability to monitor and diagnose problems as data is transported.
- Providing a centralized place for contributed connectors and related projects.

Current Status

Meritocracy

Flume was initially developed by Jonathan Hsieh in July 2009 along with development team at Cloudera. Developers external to Cloudera provided feedback, suggested features and fixes and implemented extensions of Flume. Cloudera engineering team has since maintained the project with Jonathan Hsieh, Henry Robinson, and Patrick Hunt dedicated towards its improvement. Contributors to Flume and its connectors include developers from different companies and different parts of the world.

Community

Flume is currently used by a number of organizations all over the world. Flume has an active and growing user and developer community with active participation in user and developer mailing lists. The users and developers also communicate via IRC on #flume at irc.freenode.net.

Since open sourcing the project, there have been over 15 different people from diverse organizations who have contributed code. During this period, the project team has hosted open, in-person, quarterly meetups to discuss new features, new designs, and new use-case stories.

Core Developers

The core developers for Flume project are:

- Andrew Bayer: Andrew has a lot of expertise with build tools, specifically Jenkins continuous integration and Maven.
- Jonathan Hsieh: Jonathan designed and implemented much of the original code.
- Patrick Hunt: Patrick has improved the web interfaces of Flume components and contributed several build quality improvements.
- Bruce Mitchener: Bruce has improved the internal logging infrastructure as well as edited significant portions of the Flume manual.
- Henry Robinson: Henry has implemented much of the ZooKeeper integration, plugin mechanisms, as well as several Flume features and bug fixes.
- Eric Sammer: Eric has implemented the Maven build, as well as several Flume features and bug fixes.

All core developers of the Flume project have contributed towards Hadoop or related Apache projects and are very familiar with Apache principals and philosophy for community driven software development.

Alignment

Flume complements Hadoop Map-Reduce, Pig, Hive, HBase by providing a robust mechanism to allow log data integration from external systems for effective analysis. Its design enable efficient integration of newly ingested data to Hive's data warehouse.

Flume's architecture is open and easily extensible. This has encouraged many users to contribute integrate plugins to other projects. For example, several users have contributed connectors to message queuing and bus services, to several open source data stores, to incremental search indexes, and to a stream analysis engines.

Known Risks

Orphaned Products

Flume is already deployed in production at multiple companies and they are actively participating in feature requests and user led discussions. Flume is getting traction with developers and thus the risks of it being orphaned are minimal.

Inexperience with Open Source

All code developed for Flume has is open sourced by Cloudera under Apache 2.0 license. All committers of Flume project are intimately familiar with the Apache model for open-source development and are experienced with working with new contributors.

Homogeneous Developers

The initial set of committers is from a reduced set of organizations. However, we expect that once approved for incubation, the project will attract new contributors from diverse organizations and will thus grow organically. The participation of developers from several different organizations in the mailing list is a strong indication for this assertion.

Reliance on Salaried Developers

It is expected that Flume will be developed on salaried and volunteer time, although all of the initial developers will work on it mainly on salaried time.

Relationships with Other Apache Products

Flume depends upon other Apache Projects: Apache Hadoop, Apache Log4J, Apache ZooKeeper, Apache Thrift, Apache Avro, multiple Apache Commons components. Its build depends upon Apache Ant and Apache Maven.

Flume users have created connectors that interact with several other Apache projects including Apache HBase and Apache Cassandra.

Flume's functionality has some indirect or direct overlap with the functionality of Apache Chukwa but has several significant architectural diffferences. Both systems can be used to collect log data to write to hdfs. However, Chukwa's primary goals are the analytic and monitoring aspects of a Hadoop cluster. Instead of focusing on analytics, Flume focuses primarily upon data transport and integration with a wide set of data sources and data destinations. Architecturally, Chukwa components are individually and statically configured. It also depends upon Hadoop MapReduce for its core functionality. In contrast, Flume's components are dynamically and centrally configured and does not depend directly upon Hadoop MapReduce. Furthermore, Flume provides a more general model for handling data and enables integration with projects such as Apache Hive, data stores such as Apache HBase, Apache Cassandra and Voldemort, and several Apache Lucene-related projects.

An Excessive Fascination with the Apache Brand

We would like Flume to become an Apache project to further foster a healthy community of contributors and consumers around the project. Since Flume directly interacts with many Apache Hadoop-related projects by solves an important problem of many Hadoop users, residing in the Apache Software Foundation will increase interaction with the larger community.

Documentation

- All Flume documentation (User Guide, Developer Guide, Cookbook, and Windows Guide) is maintained within Flume sources and can be built directly.
- Cloudera provides documentation specific to its distribution of Flume at: http://archive.cloudera.com/cdh/3/flume/
- Flume wiki at GitHub: https://github.com/cloudera/flume/wiki
- Flume jira at Cloudera: https://issues.cloudera.org/browse/flume

Initial Source

- https://github.com/cloudera/flume/tree/

Source and Intellectual Property Submission Plan

- The initial source is already licensed under the Apache License, Version 2.0. https://github.com/cloudera/flume/blob/master/LICENSE

External Dependencies

The required external dependencies are all Apache License or compatible licenses. Following components with non-Apache licenses are enumerated:

- org.arabidopsis.ahocorasick : BSD-style

Non-Apache build tools that are used by Flume are as follows:

- AsciiDoc: GNU GPLv2
- FindBugs: GNU LGPL
- Cobertura: GNU GPLv2
- PMD : BSD-style

Cryptography

Flume uses standard APIs and tools for SSH and SSL communication where necessary.

Required Resources

Mailing lists

- flume-private (with moderated subscriptions)
- flume-dev
- flume-commits
- flume-user

Subversion Directory

https://svn.apache.org/repos/asf/incubator/flume

Issue Tracking

JIRA Flume (FLUME)

Other Resources

The existing code already has unit and integration tests so we would like a Jenkins instance to run them whenever a new patch is submitted. This can be added after project creation.

Initial Committers

- Andrew Bayer (abayer at cloudera dot com)
- Derek Deeter (ddeeterctrb at gmail dot com)
- Jonathan Hsieh (jon at cloudera dot com)
- Patrick Hunt (phunt at cloudera dot com)
- Aaron Kimball (akimball83 at gmail dot com)
- Bruce Mitchener (bruce.mitchener at gmail dot com)
- Arvind Prabhakar (arvind at cloudera dot com)
- Ahmed Radwan (ahmed at cloudera dot com)
- Henry Robinson (henry at cloudera dot com)
- Eric Sammer (esammer at cloudera dot com)
- Nick Verbeck (nerdynick at gmail dot com)

Affiliations

- Andrew Bayer, Cloudera
- Derek Deeter, Intuit
- Jonathan Hsieh, Cloudera
- Patrick Hunt, Cloudera
- Aaron Kimball, Odiago
- Bruce Mitchener, Independent
- Arvind Prabhakar, Cloudera
- Ahmed Radwan, Cloudera
- Henry Robinson, Cloudera
- Eric Sammer, Cloudera
- Nick Verbeck, Lijit

Sponsors

Champion

- Nigel Daley

Nominated Mentors

- Tom White
- Nigel Daley
- Ralph Goers
- Patrick Hunt

Sponsoring Entity

- Apache Incubator PMC