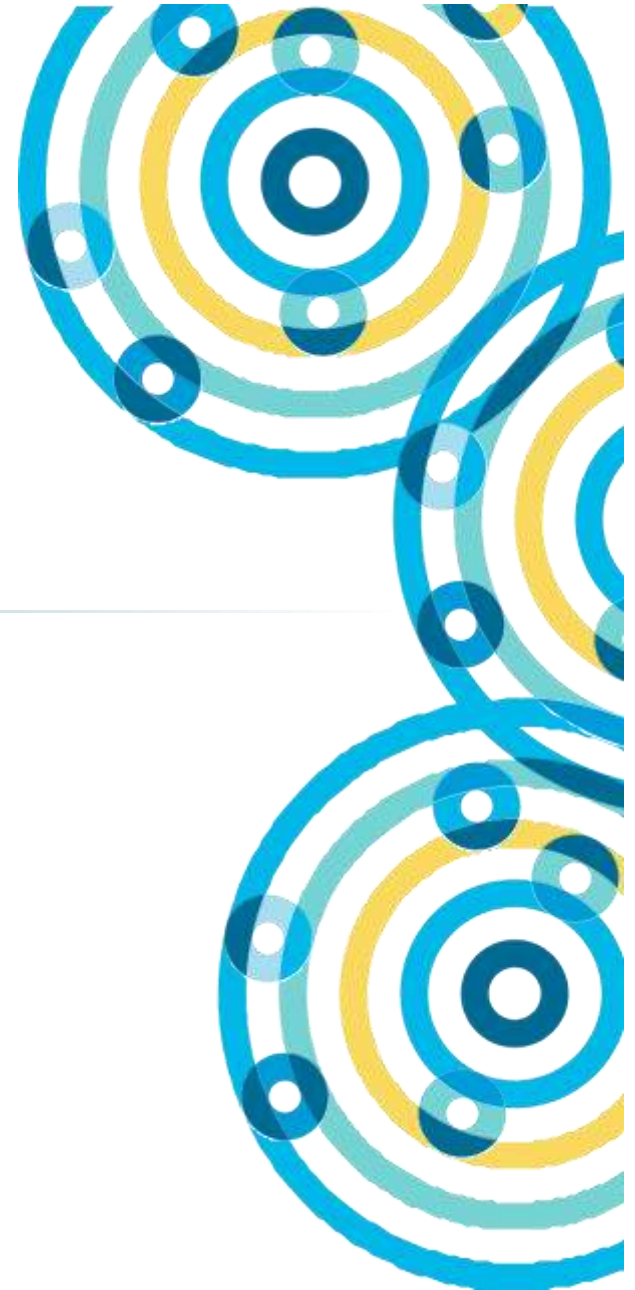




Capturing Data with Apache Flume



Capturing Data with Apache Flume

In this chapter you will learn

What are the main architectural components of Flume

How these components are configured

How to launch a Flume agent

How to configure a standard Java application to log data using Flume

Chapter Topics

Capturing Data with Apache Flume

Introduction to Flume

What is Apache Flume?

Basic Flume Architecture

Flume Sources

Flume Sinks

Flume Channels

Flume Configuration

Conclusion

What Is Apache Flume?

Apache Flume is a high-performance system for data collection

- Name derives from original use case of near-real time log data ingestion
- Now widely used for collection of any streaming event data
- Supports aggregating data from many sources into HDFS

Originally developed by Cloudera

- Donated to Apache Software Foundation in 2011
- Became a top-level Apache project in 2012
- Flume OG gave way to Flume NG (Next Generation)

Benefits of Flume

- Horizontally-scalable
- Extensible
- Reliable



Flume's Design Goals: Reliability

Channels provide Flume's reliability

Memory Channel

- Data will be lost if power is lost

Disk-based Channel

- Disk-based queue guarantees durability of data in face of a power loss

Data transfer between Agents and Channels is transactional

- A failed data transfer to a downstream agent rolls back and retries

Can configure multiple Agents with the same task

- For example, 2 Agents doing the job of 1 'collector' – if one agent fails then upstream agents would fail over

Flume's Design Goals: Scalability

Scalability

- The ability to increase system performance linearly or by adding more resources to the system
- Flume scales horizontally
- As load increases, more machines can be added to the configuration

Flume's Design Goals: Extensibility

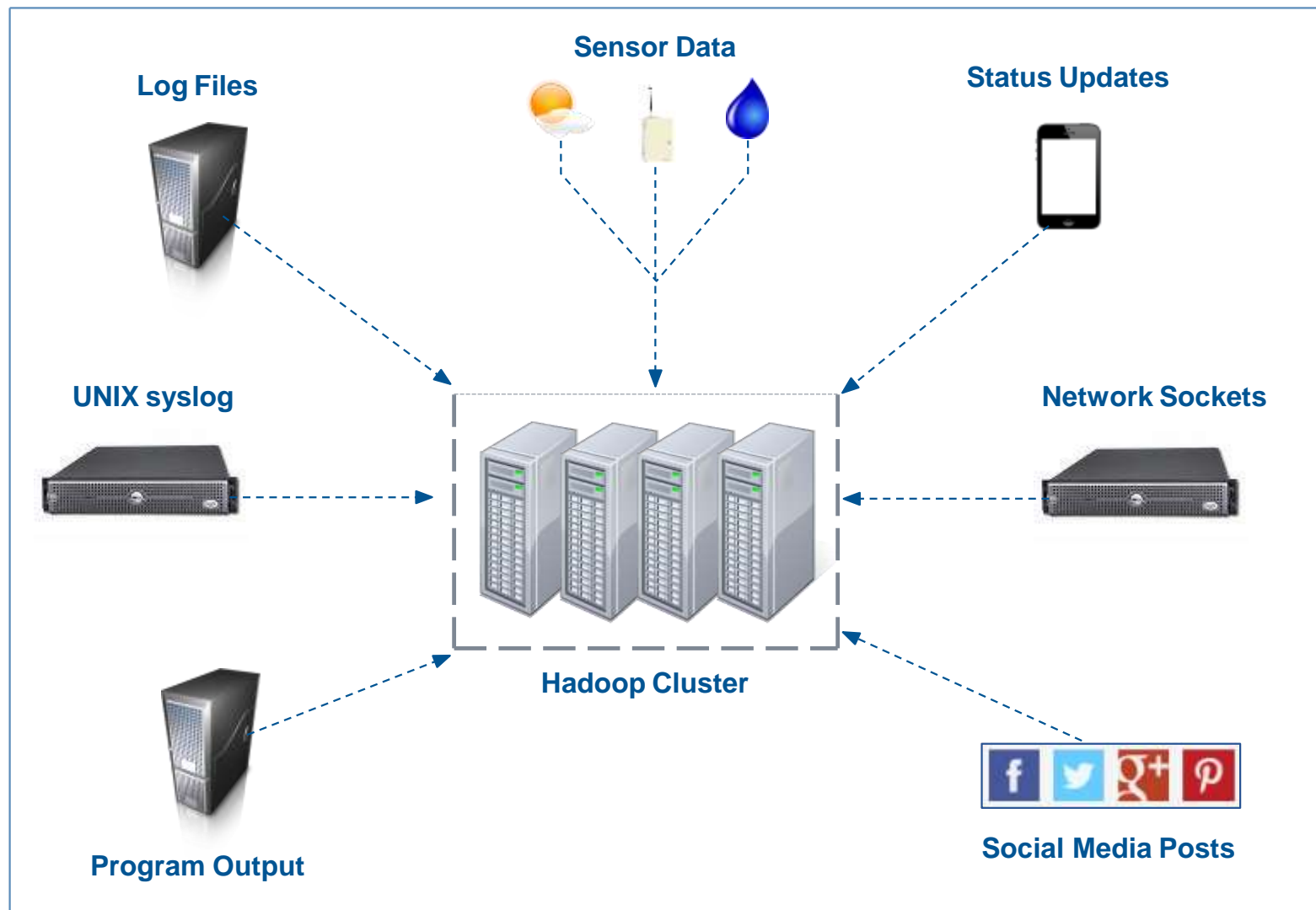
Extensibility

- The ability to add new functionality to a system

Flume can be extended by adding Sources and Sinks to existing storage layers or data platforms

- General Sources include data from files, syslog, and standard output from any Linux process
- General Sinks include files on the local filesystem or HDFS
- Developers can write their own Sources or Sinks

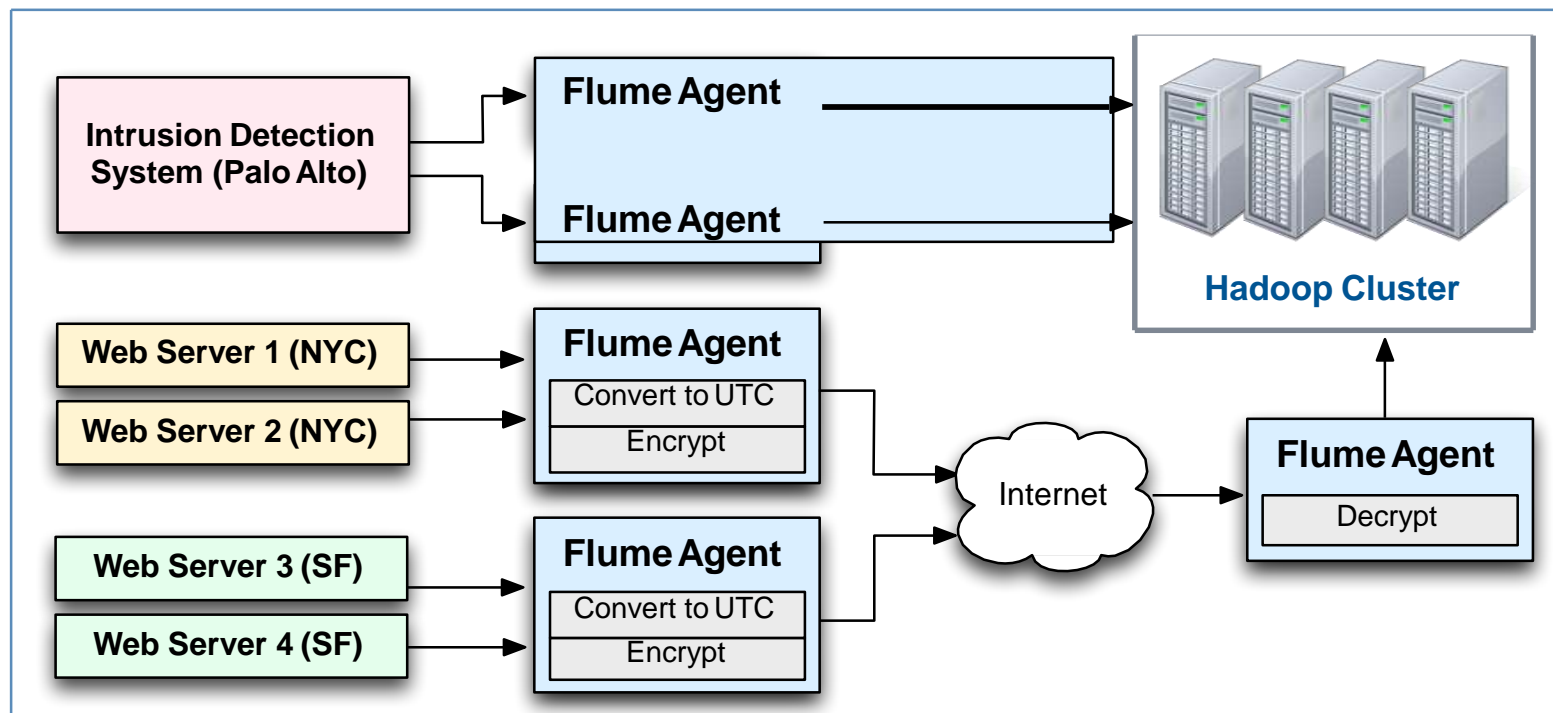
Common Flume Data Sources



Large-Scale Deployment Example

Flume collects data using configurable “agents”

- Agents can receive data from many sources, including other agents
- Large-scale deployments use multiple tiers for scalability and reliability
- Flume supports inspection and modification of in-flight data



Chapter Topics

Capturing Data with Apache Flume

Introduction to Flume

What is Apache Flume?

Basic Flume Architecture

Flume Sources

Flume Sinks

Flume Channels

Flume Configuration

Conclusion

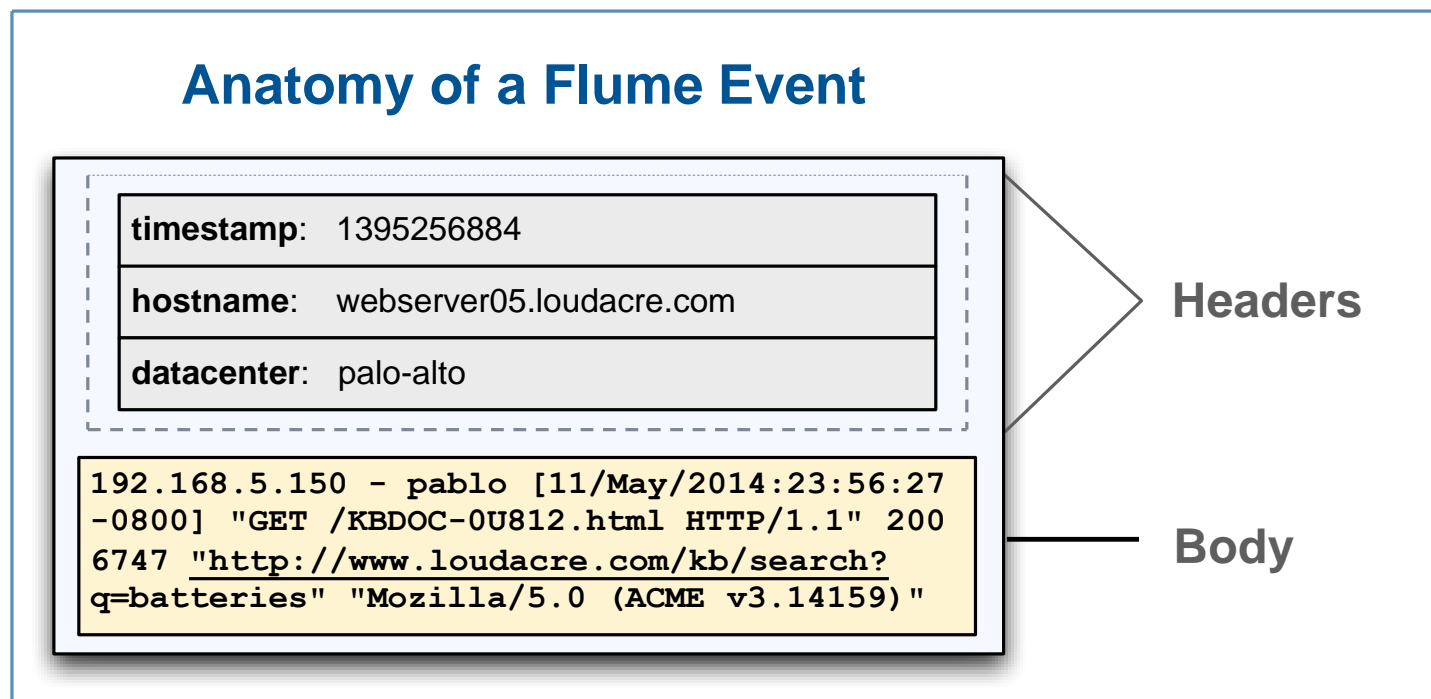
Flume Events

An *event* is the fundamental unit of data in Flume

- Consists of a body (payload) and a collection of headers (metadata)

Headers consist of name-value pairs

- Headers are mainly used for directing output



Components in Flume's Architecture

Source

- Receives events from the external actor that generates them

Sink

- Sends an event to its destination

Channel

- Buffers events from the source until they are drained by the sink

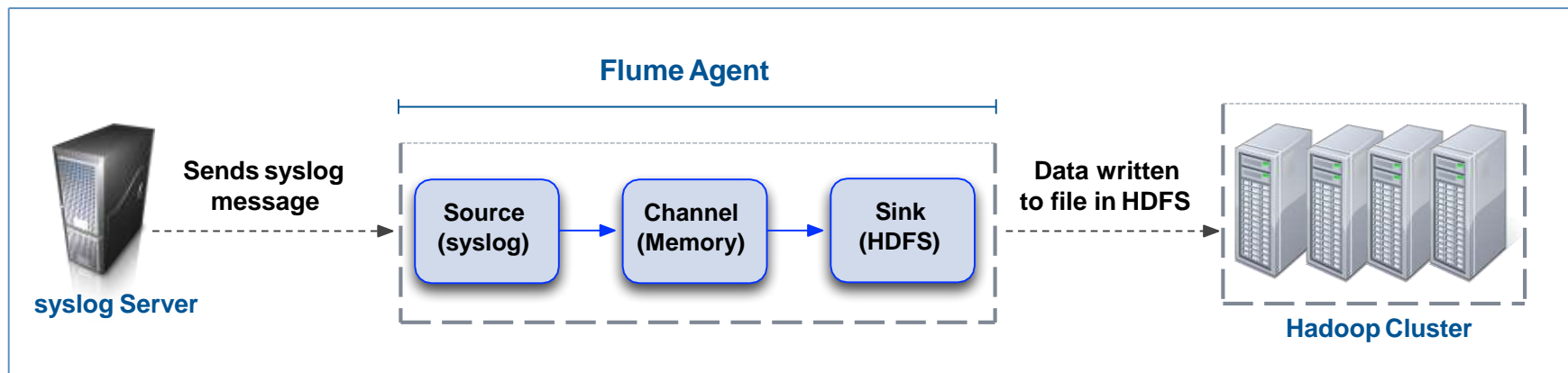
Agent

- Java process that configures and hosts the source, channel, and sink

Flume Data Flow

This diagram illustrates how syslog data might be captured to HDFS

1. Message is logged on a server running a syslog daemon
2. Flume agent configured with syslog source receives event
3. Source pushes event to the channel, where it is buffered in memory
4. Sink pulls data from the channel and writes it to HDFS



Chapter Topics

Capturing Data with Apache Flume

Introduction to Flume

What is Apache Flume?

Basic Flume Architecture

Flume Sources

Flume Sinks

Flume Channels

Flume Configuration

Conclusion

Notable Built-in Flume Sources

Syslog

- Captures messages from UNIX syslog daemon over the network

Netcat

- Captures any data written to a socket on an arbitrary TCP port

Exec

- Executes a UNIX program and reads events from standard output *

Spooldir

- Extracts events from files appearing in a specified (local) directory

HTTP Source

- Receives events from HTTP requests

* Asynchronous sources do not guarantee that events will be delivered

Chapter Topics

Capturing Data with Apache Flume

Introduction to Flume

What is Apache Flume?

Basic Flume Architecture

Flume Sources

Flume Sinks

Flume Channels

Flume Configuration

Conclusion

Interesting Built-in Flume Sinks

Null

- Discards all events (Flume equivalent of `/dev/null`)

Logger

- Logs event to INFO level using SLF4J

IRC

- Sends event to a specified Internet Relay Chat channel

HDFS

- Writes event to a file in the specified directory in HDFS

HBaseSink

- Stores event in HBase

SLF4J: Simple Logging Façade for Java

Chapter Topics

Capturing Data with Apache Flume

Introduction to Flume

What is Apache Flume?

Basic Flume Architecture

Flume Sources

Flume Sinks

Flume Channels

Flume Configuration

Conclusion

Built-In Flume Channels

Memory

- Stores events in the machine's RAM
- Extremely fast, but not reliable (memory is volatile)

File

- Stores events on the machine's local disk
- Slower than RAM, but more reliable (data is written to disk)

JDBC

- Stores events in a database table using JDBC
- Slower than file channel

Chapter Topics

Capturing Data with Apache Flume

Introduction to Flume

What is Apache Flume?

Basic Flume Architecture

Flume Sources

Flume Sinks

Flume Channels

Flume Configuration

Conclusion

Flume Agent Configuration File

Flume agent is configured through a Java properties file

- Multiple agents can be configured in a single file

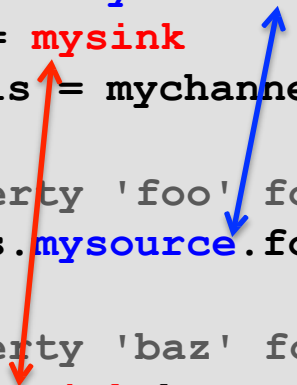
The configuration file uses hierarchical references

- Each component is assigned a user-defined ID
- That ID is used in the names of additional properties

```
# Define sources, sinks, and channel for agent named 'agent1'
agent1.sources = mysource
agent1.sinks = mysink
agent1.channels = mychannel

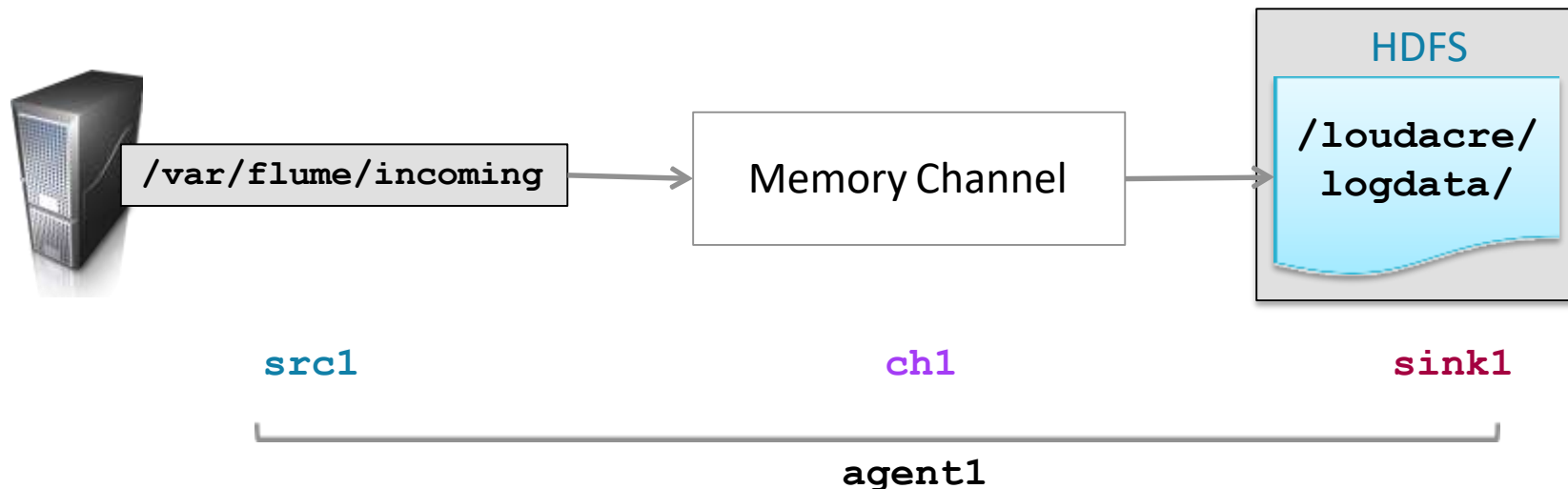
# Sets a property 'foo' for the source associated with agent1
agent1.sources.mysource.foo = bar

# Sets a property 'baz' for the sink associated with agent1
agent1.sinks.mysink.baz = bat
```



Example: Configuring Flume Components (1)

Example: Configure a Flume Agent to collect data from remote spool directories and save to HDFS



Example: Configuring Flume Components (2)

```
agent1.sources = src1
agent1.sinks = sink1
agent1.channels = ch1

agent1.channels.ch1.type = memory

agent1.sources.src1.type = spooldir
agent1.sources.src1.spoolDir = /var/flume/incoming
agent1.sources.src1.channels = ch1

agent1.sinks.sink1.type = hdfs
agent1.sinks.sink1.hdfs.path = /loudacre/logdata
agent1.sinks.sink1.channel = ch1
```

Connects **source**
and channel

Connects **sink**
and channel

Properties vary by component type (source, channel, and sink)

- Properties also vary by subtype (e.g., netcat source vs. syslog source)
- See the Flume user guide for full details on Configuration

Aside: HDFS Sink Configuration

Path may contain patterns based on event headers, such as timestamp

The HDFS sink writes uncompressed SequenceFiles by default

- Specifying a codec will enable compression

```
agent1.sinks.sink1.type = hdfs
agent1.sinks.sink1.hdfs.path = /loudacre/logdata/%y-%m-%d
agent1.sinks.sink1.hdfs.codec = snappy
agent1.sinks.sink1.channel = ch1
```

Setting fileType parameter to DataStream writes *raw* data

- Can also specify a file extension, if desired

```
agent1.sinks.sink1.type = hdfs
agent1.sinks.sink1.hdfs.path = /loudacre/logdata/%y-%m-%d
agent1.sinks.sink1.hdfs.fileType = DataStream
agent1.sinks.sink1.hdfs.fileSuffix = .txt
agent1.sinks.sink1.channel = ch1
```


Starting a Flume Agent

Typical command line invocation

- The **--name** argument must match the agent's name in the Configuration file
- Setting root logger as shown will display log messages in the terminal

```
$ flume-ng agent \  
  --conf /etc/flume-ng/conf \  
  --conf-file /path/to/flume.conf \  
  --name agent1 \  
  -Dflume.root.logger=INFO,console
```

* ng = Next Generation (prior version now referred to as og)

Chapter Topics

Capturing Data with Apache Flume

Introduction to Flume

What is Apache Flume?

Basic Flume Architecture

Flume Sources

Flume Sinks

Flume Channels

Flume Configuration

Conclusion

EssenFal Points

Apache Flume is a high-performance system for data collection

- Scalable, extensible, and reliable

A Flume agent manages the source, channels, and sink

- Source receives event data from its origin
- Sink sends the event to its destination
- Channel buffers events between the source and sink

The Flume agent is configured using a properties file

- Each component is given a user-defined ID
- This ID is used to define properties of that component

Bibliography

The following offer more information on topics discussed in this chapter

Flume User Guide

–<http://flume.apache.org/FlumeUserGuide.html>