Introduction to Apache NiFi

RequiTest
Define Your Success!

What is Apache NiFi?

An Open Source Data Distribution and Processing System

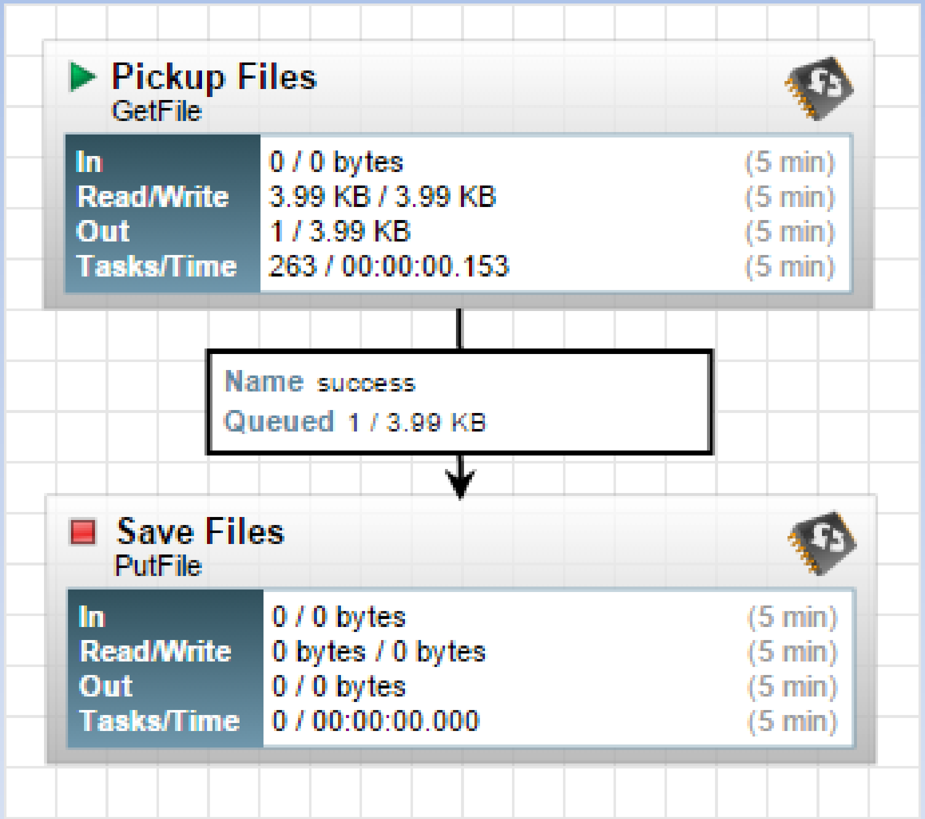
What does that mean?

Apache NiFi provides a way to move data from one place to another, making routing decisions and transformations as necessary along the way

Why Use Apache NiFi?

- Easy to use
- Powerful
- Reliable
- Secure
- Scalable

Can handle Basic Flows…



▶ Pickup Files
GetFile

In	0 / 0 bytes	(5 min)
Read/Write	3.99 KB / 3.99 KB	(5 min)
Out	1 / 3.99 KB	(5 min)
Tasks/Time	263 / 00:00:00.153	(5 min)

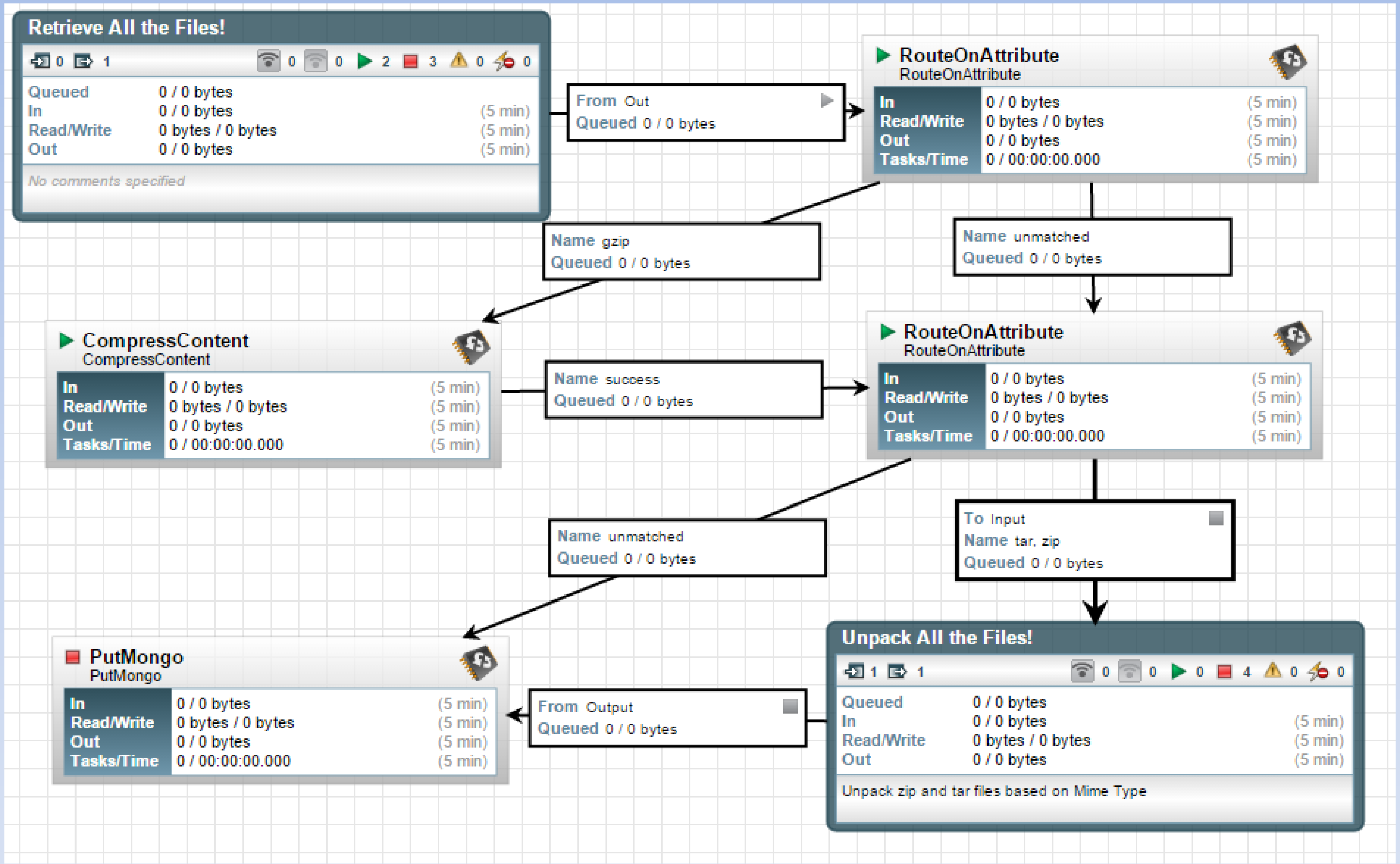
Name success
Queued 1 / 3.99 KB

■ Save Files
PutFile

In	0 / 0 bytes	(5 min)
Read/Write	0 bytes / 0 bytes	(5 min)
Out	0 / 0 bytes	(5 min)
Tasks/Time	0 / 00:00:00.000	(5 min)

REQUITEST
Define Your Success!

… To More Advanced Flows



Retrieve All the Files!
0 1 0 0 2 3 0 0

Queued	0 / 0 bytes	
In	0 / 0 bytes	(5 min)
Read/Write	0 bytes / 0 bytes	(5 min)
Out	0 / 0 bytes	(5 min)

No comments specified

From Out
Queued 0 / 0 bytes

RouteOnAttribute
RouteOnAttribute

In	0 / 0 bytes	(5 min)
Read/Write	0 bytes / 0 bytes	(5 min)
Out	0 / 0 bytes	(5 min)
Tasks/Time	0 / 00:00:00.000	(5 min)

Name gzip
Queued 0 / 0 bytes

Name unmatched
Queued 0 / 0 bytes

CompressContent
CompressContent

In	0 / 0 bytes	(5 min)
Read/Write	0 bytes / 0 bytes	(5 min)
Out	0 / 0 bytes	(5 min)
Tasks/Time	0 / 00:00:00.000	(5 min)

Name success
Queued 0 / 0 bytes

RouteOnAttribute
RouteOnAttribute

In	0 / 0 bytes	(5 min)
Read/Write	0 bytes / 0 bytes	(5 min)
Out	0 / 0 bytes	(5 min)
Tasks/Time	0 / 00:00:00.000	(5 min)

Name unmatched
Queued 0 / 0 bytes

To Input
Name tar, zip
Queued 0 / 0 bytes

Unpack All the Files!
1 1 0 0 0 4 0 0

Queued	0 / 0 bytes	
In	0 / 0 bytes	(5 min)
Read/Write	0 bytes / 0 bytes	(5 min)
Out	0 / 0 bytes	(5 min)

Unpack zip and tar files based on Mime Type

From Output
Queued 0 / 0 bytes

PutMongo
PutMongo

In	0 / 0 bytes	(5 min)
Read/Write	0 bytes / 0 bytes	(5 min)
Out	0 / 0 bytes	(5 min)
Tasks/Time	0 / 00:00:00.000	(5 min)

REQUITEST
Define Your Success!

Features

- Web-based Interface
  - Flow construction, control, and monitoring all from a single easy to use interface
- Data Provenance
  - Track data throughout the entire flow
  - Information about FlowFiles as they traverse the flow are automatically indexed
  - Critical for supporting troubleshooting and flow optimization.

Features

- Data Recovery
  - Ages off content as space is needed
  - Allows for fine grained download, recovery, and replay of individual files.
- Secure
  - Provides content encryption, communication over secure protocols (SSL, SSH, HTTPS), etc.
  - Provides a pluggable role-based authentication/authorization mechanism for both data transfer and user management

Features

- Highly configurable
  - Fine grained Quality of Service control
  - Dataflow modifiable at runtime
  - Loss tolerant vs guaranteed delivery
  - Low latency vs high throughput
  - Back pressure

Features

- Extensible
  - Build your own processors, controller services, and more
  - Enables rapid development and effective testing
  - Allows for development of simple single function components that can be reused and combined to make more complex flows
  - Provides classloader isolation for easier management of dependencies

Definitions

FlowFile

- The data that moves through the flow
- Can be cloned, merged, split, modified, transferred, and deleted
- Consists of:
  - Map of key/value pair attribute strings
  - Content of zero or more bytes

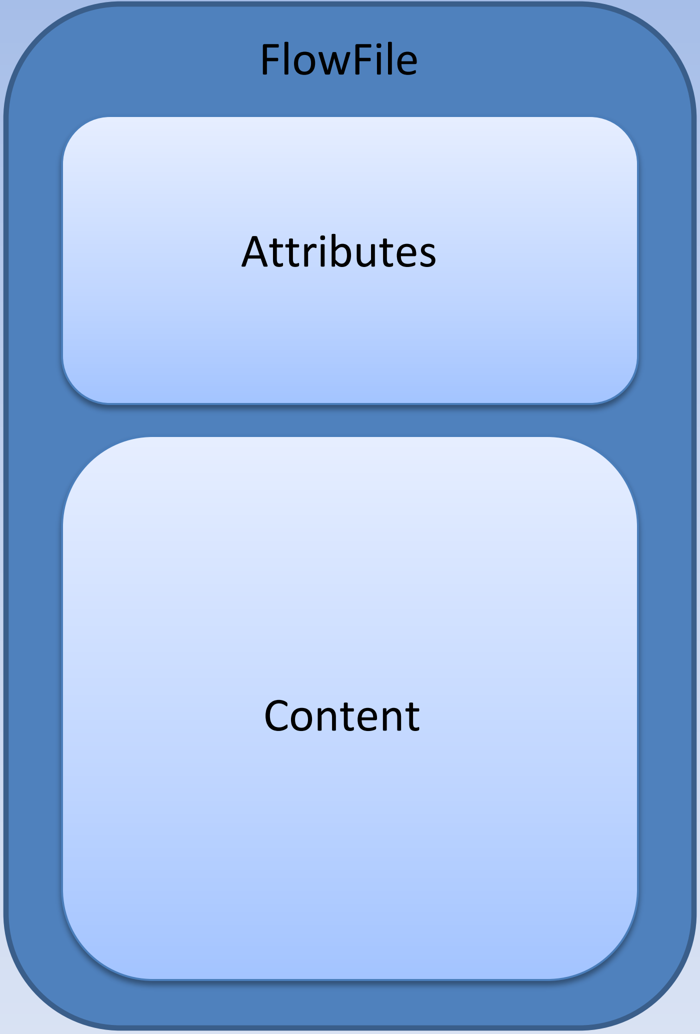
FlowFile Breakdown

Attributes

- Map of Key/Value pairs
- Heavily used to make routing decisions
- Values accessed using NiFi's Expression Language

Content

- The actual data that is being routed through the dataflow
- May be manipulated multiple times throughout the course of a dataflow



Common Attributes

filename – A filename that can used when storing data locally or on a remote system

path – the directory that can be used when storing data

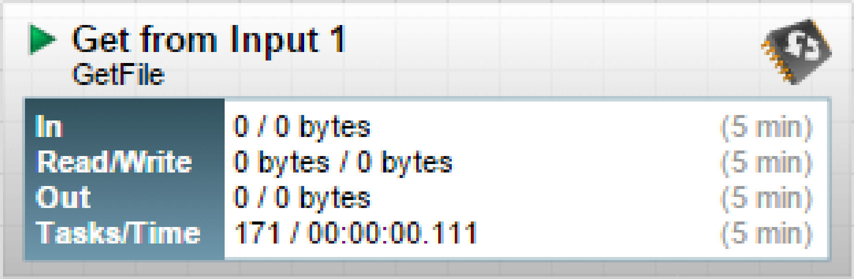
uuid – A Universally Unique Identifier that distinguishes FlowFile from other FlowFiles

entryDate – the date and time at which the FlowFile entered the system

lineageStartDate – The date and time at which the oldest ancestor of the FlowFile entered the system.

fileSize – Represents the number of bytes taken up by the FlowFile’s Content

Definitions



Get from Input 1

GetFile

In	0 / 0 bytes	(5 min)
Read/Write	0 bytes / 0 bytes	(5 min)
Out	0 / 0 bytes	(5 min)
Tasks/Time	171 / 00:00:00.111	(5 min)

FlowFile Processor

- Single step in the flow
- Performs the work on the FlowFile
  - Routing
  - Data Transformation
  - Mediation between systems
- Has access to FlowFile content and attributes
- Can operate on zero or more FlowFiles in a single unit of work

REQUITEST

Define Your Success!

FlowFile Processor Examples

Ingestion

- **GetFile** – Pull content from the local disk and delete the original file
- **GetSFTP** – Pull content from a remote system then delete the original file

Routing

- **RouteOnAttribute** – Route FlowFiles based on the values of specific FlowFile attributes

Data Transformation

- **CompressContent** – Compress or decompress content
- **ReplaceText** – Use Regular Expressions to modify textual content

FlowFile Processor Examples

Data Egress

- **PutFile** – Writes the FlowFile contents to a directory on the local disk
- **PutSFTP** – Copies the contents of the FlowFile to a remote server

Attribute Extraction

- **UpdateAttribute** – Adds or updates attributes using statically defined values or dynamically derived values using NiFi's Expression Language
- **ExtractText** – Creates attributes based on User defined Regular Expressions

Splitting and Aggregation

- **UnpackContent** – Unpacks archive formats such as TAR and ZIP and sends each file within the archive as a separate FlowFile through the dataflow

Definitions

Connection

- Provides linkage between processors
- Queues that allow rate control
- Dynamically prioritizable
- Enable back pressure via configurable upper bounds

Definitions

Controller Service

- A single service that can be shared between multiple FlowFile processors
- Performs a specific task or maintains a common set of information
- Example: **StandardSSLContextService** provides a single configuration for a keystore and/or truststore that can be used throughout a dataflow

Definitions

Flow Controller

- Scheduler
- Maintains processor and connection configuration
- Handles scheduling of threads which processes use

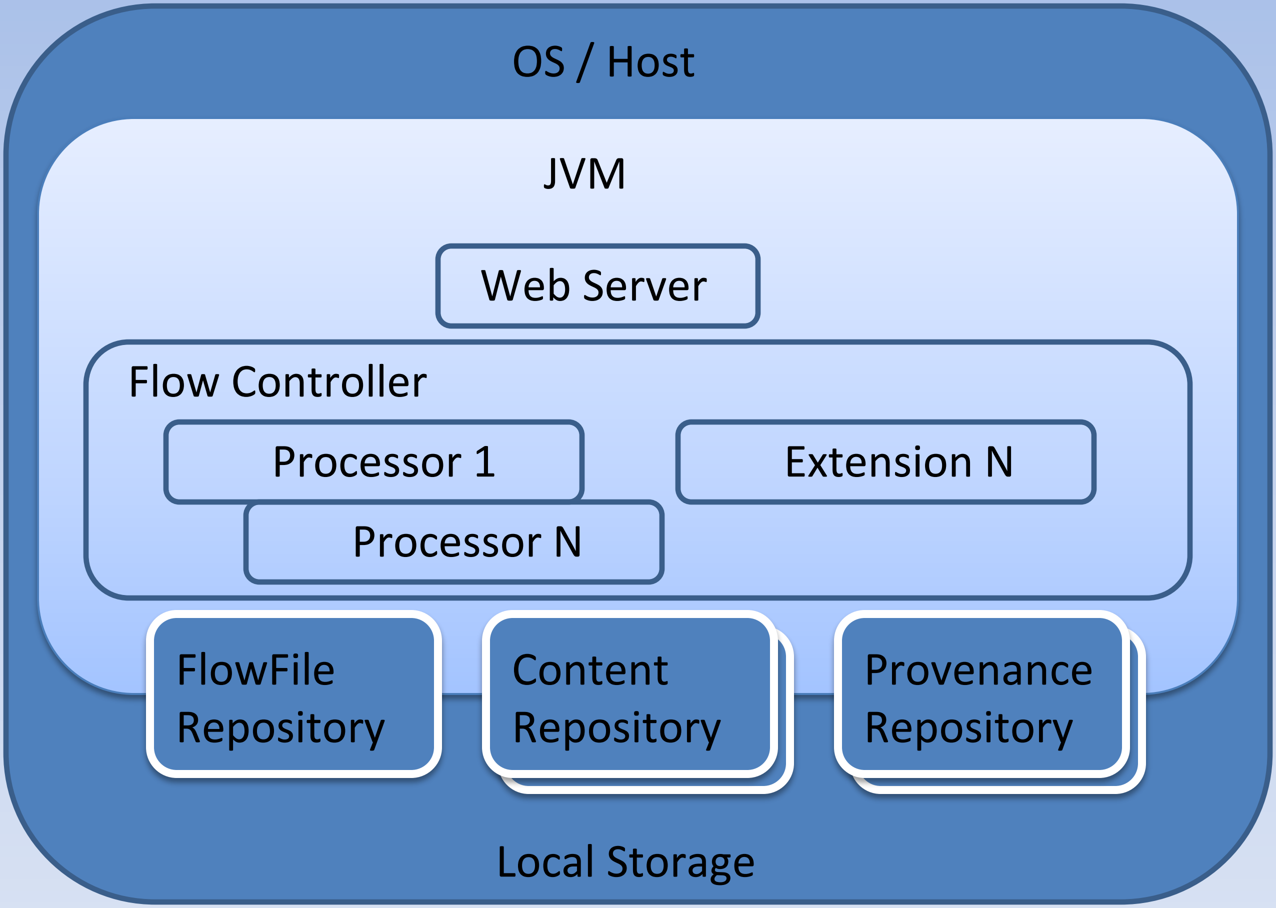
Process Group

- Set of processors and their connections
- Receives data via input port(s) and sends data via output port(s)

NiFi Architecture

- NiFi is a Java based system that executes within a JVM.
- Primary components are:
  - Web Server
    - Hosts NiFi’s HTTP-based control API
  - Flow Controller
    - Provides and schedules threads for execution
  - Extensions
    - FlowFile Processors, Controller Services, etc.
  - Repositories
    - FlowFile
    - Content
    - Provenance

NiFi Architecture



REQUITEST
Define Your Success!

Repositories

FlowFile Repository

- Holds information pertaining to the FlowFile and its attributes

Content Repository

- Holds all of the FlowFile content

Provenance Repository

- Holds all information pertaining to the life of the FlowFile as it traverses the dataflow

How Do I Get It?

http://nifi.apache.org/download.html

Two versions available:

- A “tarball” tailored for Linux
- A zip file tailored for Windows

Download the appropriate version and extract to the location from which you want to run NiFi.

Mac OSX Users may also use the tarball or can install via Homebrew by running:

```
brew install nifi
```

Running NiFi (Linux/Mac OSX)

Using a Terminal window, navigate to the directory where NiFi was installed.

To run NiFi in the foreground, run

```
bin/nifi.sh run
```

Use Ctrl-C to stop the application.

To run NiFi in the background, run

```
bin/nifi.sh start
```

To stop the application, use

```
bin/nifi.sh stop
```

Running NiFi (Windows)

Nagivate to the folder where NiFi was installed.
Double-click the *bin/run-nifi.bat* file.

To stop the application, select the window that was launched and press Ctrl-C.

Congratulations!

To start using NiFi, open a web browser and navigate to http://localhost:8080/nifi

Port 8080 is the default port and can be changed by editing the `nifi.properties` file in the NiFi `conf` directory.

Further Resources

- RequiTest Website:
  http://requitest.com/

- Apache NiFi Website:
  http://nifi.apache.org/

- Apache NiFi Users Mailing List:
  http://mail-archives.apache.org/mod_mbox/nifi-users/

- Apache NiFi Developers Mailing List:
  http://mail-archives.apache.org/mod_mbox/nifi-dev/

REQUITEST
Define Your Success!