

ACCUM2_BENCHMARKING

Accum2_Benchmarking is a set of bash scripts for installing and running Accumulo 2.0 tests for benchmarking purposes. Refer to these documents for more information on Accumulo benchmark testing.

[Accumulo Testing](#)

[Accumulo Benchmarking Document](#)

Hadoop and Accumulo provide numerous configuration options to optimize performance for a given platform, and type of workload. It is therefore not the intention of these scripts to optimize performance. Their purpose is to allow quick setup of a Hadoop/Accumulo environment in which the user can then run performance benchmarks and optimize their installation to meet their specific needs.

These scripts were not written to create a secure production Accumulo environment. They will create an insecure environment meant only for testing.

This instance is set up for Maven 3.6.3, Zookeeper 3.6.2, hadoop 3.1.2, accumulo 2.0.1, and accumulo-testing 2.0, but could be modified for other versions. These scripts should be relatively Linux hardware platform independent.

Obviously there could be many derivations of these setup scripts. These could include; different sw versions, different OS, specific hardware tweaks, performance optimizations for any of the preceding, etc, etc. If you wish to contribute a derivation setup, please do so on a branch, and thoroughly document your intent.

Assumptions/Requirements

1. These scripts were tested with openjdk 1.8.
2. These scripts were tested on AWS.
3. These scripts require gcc version 5.1 or greater for accumulo native library compile.
4. These scripts require pdsh.
5. These scripts require patch.
6. These scripts require that nodes have access to both a shared file system and at least one local drive.
7. These scripts require that the user have read, write, and execute permission on local drives.
8. These scripts require that the user can passwordless scp between nodes.
9. These scripts assume CentOS 7.

Usage

If this repository was cloned from github, the user will also need to download the following files from the appropriate locations on the internet prior to running the install script:

- 2.0.zip - Accumulo 2.0 tests
- accumulo-2.0.1-bin.tar.gz - Accumulo 2.0.1
- apache-zookeeper-3.6.2-bin.tar.gz - Zookeeper 3.6.2
- apache-maven-3.6.3-bin.tar.gz - Maven 3.6.3
- hadoop-3.1.2.tar.gz - Hadoop 3.1.2
- Optional: main.zip - accumulo-proxy

Set Configuration

The file `acc_test.conf` contains configuration information for software installation as well as execution. It is divided into the following sections:

- System Related Variables
- Hadoop Configuration
- Zookeeper Configuration
- Accumulo Configuration
- Maven Configuration
- Accumulo Tests Configuration
- Accumulo Proxy Configuration
- Pi and Terasort mapreduce test variables
- Ingest test variables
- Bulk Ingest test variables
- Walk test variables
- Moru table test variables
- Verify table test variables

Run Installation Script

```
./acc_test_sw_install
```

This script verifies some requirement and setup aspects and then gives the user the option to proceed with installation of the software packages mentioned above.

Like all of these scripts, it relies on variables that are set in the `acc_test.conf` file.

Start up Hadoop

```
./start_hadoop
```

This script will launch Zookeeper and Hadoop based on the settings in `acc_test.conf`.

There is also `stop_hadoop` script which cleanly shuts down the Hadoop processes.

Hadoop Tests

```
./run_pi  
./run_terasort
```

After successful launch of Hadoop, the user can run the map reduce scripts to calculate pi or execute terasort.

Start up Accumulo

```
./start_accumulo
```

This script will launch accumulo and create an accumulo database instance. It will accept an optional command line argument, "split". If split is specified, Accumulo will be started with property files set to accommodate pre-splitting of tables.

There is also `stop_accumulo` script which cleanly shuts down the Accumulo processes.

Accumulo Tests

At this point the user has the option to run the various Accumulo tests that have been scripted. These include:

- **run_ingest** - This script launches ingest clients on the nodes running tservers. These clients will continue to run until the **stop_ingest** script is executed.
- **run_bulk** - This script runs bulk ingest. A map reduce job generates rfiles based on number of table splits setting. This ingest completes after a specified number of loops.
- **run_walk** - This script launches walk clients on the nodes running tservers. These clients will continue to run until the **stop_walk** script is executed.
- **run_batchwalk** - This script launches batchwalk clients on the nodes running tservers. These clients will continue to run until the **stop_batchwalk** script is executed.
- **run_verify** - This script verifies the continuous ingest table via a map reduce job.
- **run_moru** - This script runs a map reduce job that stresses Accumulo by reading and writing the continuous ingest table. This map reduce job will continue to run until the **stop_moru** script is executed.

Accumulo Benchmarks

In addition to the standard tests, the user can run the two Accumulo benchmarking tests that have been scripted.

- **test_ingest_only** - This script utilizes **run_ingest** and **stop_ingest** to measure the ingest rate of Accumulo on the cluster. It accepts "split" as a command line argument. If specified, the script uses a file named "splits" to pre-split the ci table which improves ingest performance. Ingest runs until a specified percentage of the Hadoop file system has been filled. At the conclusion of ingest, a minor compaction is run to flush any data in memory to disk, and then the ci table is taken offline to halt any alteration to the rfiles. The ingest rate is then calculated based on rfile information. Test results are stored to a date stamped log file. Run_verify should be executed after this test to prove the integrity of the ci table.
- **test_walk** - This script utilizes the **run_walk** or **run_batchwalk** and **stop_walk** or **stop_batchwalk** scripts. It scans entries in the ci table and reports on entry read rates. If executed with no option it tests walk. If executed with the "batch" command line option it tests batchwalk. Test results are saved to a date stamped log file.

Helper Programs

The following additional programs are included in this release.

- **create_splits** - Generates a file named "splits" used for pre-splitting the ci table. Pass the number of desired splits in as the single command line argument. The entries in the ci table are hexadecimal numbers. This program generates a series of sixteen digit hex numbers as split points.
- **start_up** - This script runs **acc_test_sw_install**, **start_hadoop**, and **start_accumulo**. It accepts "split" as a command line argument. If specified, **start_accumulo** is executed with the "split" option.
- **shut_down** - This script simply runs **stop_accumulo**, **stop_hadoop**, and **reset** in order. It is very important that **reset** is run before executing

acc_test_sw_install again. Running **reset** restores files and directories to the required state.