StashCache Tester Documentation

Release 0.0.2

Derek Weitzel

July 21, 2016

Contents

The StashCache Tester is designed to run periodically to test site's ability to interact with the StashCache.

Contents:

CHAPTER 1

Tutorial

In this tutorial, you will learn how to run the StashCache tester.

1.1 Requirements

StashCache Tester requires HTCondor in order to run tests. StashCache submits tests to HTCondor as a DAG.

Additionally, it requires the HTCondor Python Bindings.

1.2 Installing

The StashCache tester is distributed as a python package in PyPi. It is recommended that you install the tester inside a virtual enviornment.

The setps to install are:

```
$ virtualenv tester
$ . tester/bin/activate
$ pip install --upgrade setuptools
$ pip install stashcache_tester
```

The pip installation could take a while. It requires the compilation and installation of several packages including matplotlib and numpy.

1.3 Running StashCache

StashCache comes with an executable script, `stash-test` which will begin the submission of test jobs. A configuration file is required by `stash-test`. An example configuration file is located in `etc/stashcache-tester/tester.conf`. You can test with this configuration:

```
$  stash-test -c tester/etc/stashcache-tester/tester.conf run
```

This will submit the DAG to the cluster.

1.4 Debugging StashCache Tester

A log file is produced that will contain the debugging and error messages.

CHAPTER 2

Output Types

StashCache Tester can produce different outputs by subclassing the GeneralOutput class.

class stashcache_tester.output.generalOutput.**GeneralOutput**(*sitesData*)
The GeneralOuptut class should be subclassed by the output plugin.

Parameters **sitesData** (*dict*) – The data from sites in the form of a dictionary. The keys should be the sites, and the values should be an array of times for the transfers.

An example structure for sitesData is:

```
sitesData = {
    "UCSDT2": [
        {'starttime': "140192910", 'endtime': "140204950", 'successful': True},
        {'starttime': "140105910", ...}
    ],
    "Nebraska": [
        {'starttime': ...}]}
        ...
```

The initialize function should also be used to initialize any structures required for processing.

startProcessing()
This is called when the the output plugin should begin processing the *sitesData* data.

2.1 Example Outputs Processors

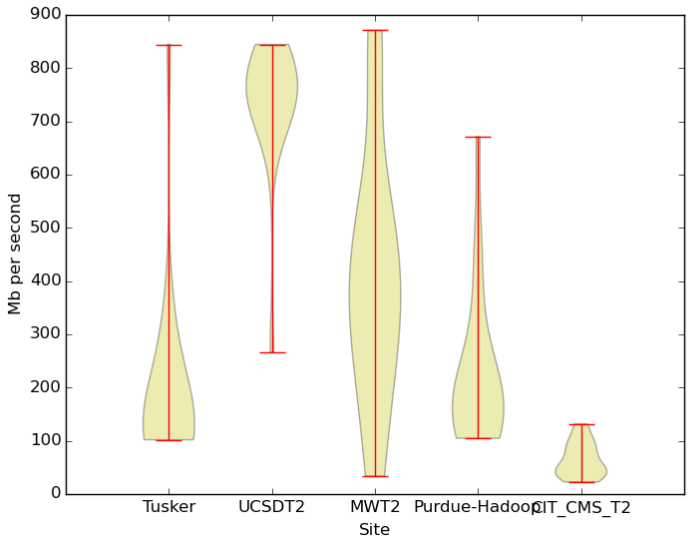
An example of an output processor is the MatplotlibOutput processor.

class stashcache_tester.output.matplotlibOutput.**MatplotlibOutput**(*sitesData*)

startProcessing()
This function will create plots using python's matplotlib. Currently, it will make:

1. A violin plot of the distribution of download times for each site given in *sitesData*.

A violin plot example:



And another example, the GithubOutput processor.

class `stashcache_tester.output.githubOutput.`**`GithubOutput`**(*sitesData*)

Parameters **`sitesData`** (*dict*) – Dictionary described in *sitesData*.

This class summarizes and uploads the download data to a github account. The data will be stored in a file named `data.json` in the git repo under the directory in the configuration. The format of `data.json` is:

```
{
    "20150911": [
        {
            "average": 364.76526180827,
            "name": "Tusker"
        },
        {
            "average": 75.99734924610296,
            "name": "UCSDT2"
        },
        ...
    ],
    "20150913": [
        {
            "average": 239.02169168535966,
            "name": "Tusker"
        },
        ...
    ],
    ...
}
```

Github output requires an SSH key to be added to the github repository which is pointed to by the *repo* configuration option.

Github output requires additional configuration options in the main configuration in the section *[github]*. An example configuration could be:

```
[github]
repo = StashCache/stashcache.github.io.git
```

```
branch = master
directory = data
ssh_key = /home/user/.ssh/id_rsa
```

The configuration is:

repo The git repo to commit the data to.

branch The branch to install repo.

directory The directory to put the data summarized files into.

maxdays The maximum number of days to keep data. Default=30

ssh_key Path to SSH key to use when checking out and pushing to the repository.

startProcessing()
Begin summarizing the data.

CHAPTER 3

Development Docs

The StashCache tester is a workflow of multiple steps and tools working together.

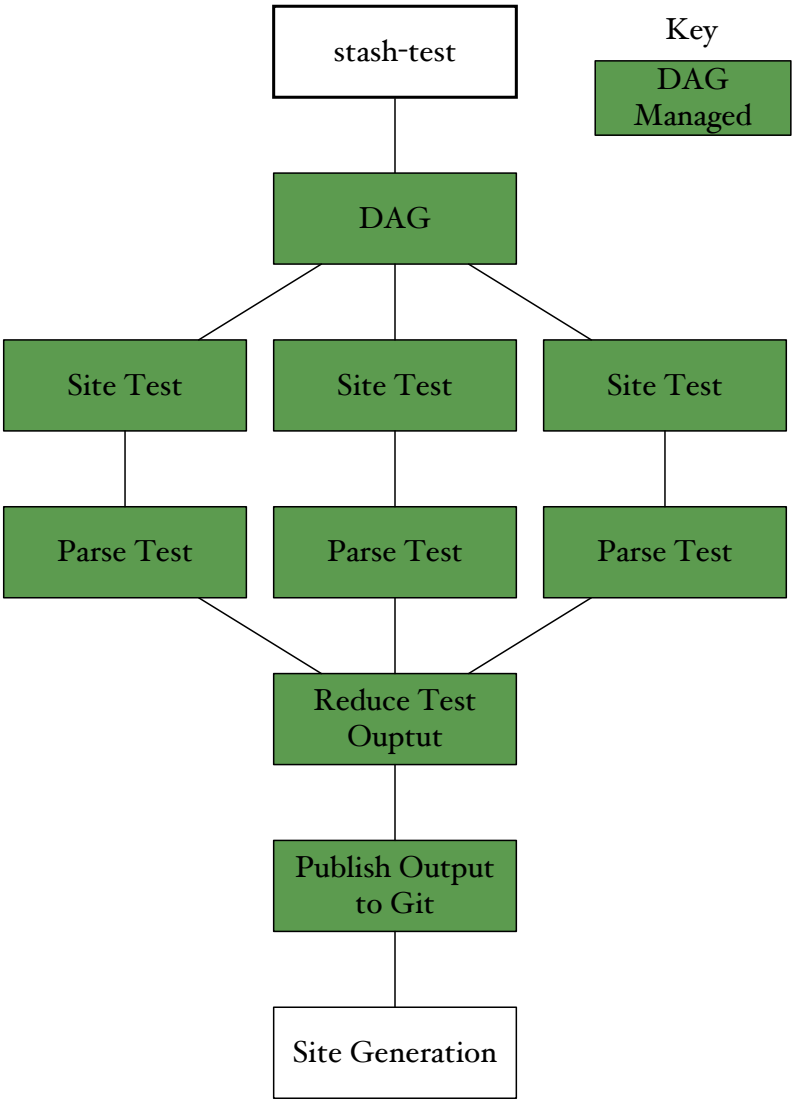


Fig. 3.1: Workflow of stashcach tester when partnered with Git output module.

CHAPTER 4

Changelog

4.1 Version 0.4.0

- Check MD5sum of test file
- Correctly report failed downloads
- Use new module stashcp/3.0

4.2 Version 0.3.0

- Adding the `maxdays` to *`stashcache_tester.output.githubOutput.GithubOutput`* to limit the number of days to keep data.

4.3 Version 0.2.0

- Add caching site to the output data for *`stashcache_tester.output.githubOutput.GithubOutput`*

4.4 Version 0.1.1

- Fix bug in post site processing when certain output exists but is blank.

4.5 Version 0.1.0

- Reconfigure data layout for github output plugin. It will now write to a single file, `data.json`.

4.6 Version 0.0.8

- Remove host key check from the github output type.
- Add condition to remove jobs which have had shadow exceptions more than 5 times.

4.7 Version 0.0.7

- Adding tests directory to contain configurations for testing the tester.
- Changing default test output directory from `tests` to `stashtests` to not conflict with the new tests directory.
- Add stdout and stderr redirection for the `site_post.py` post processing script.

4.8 Version 0.0.5 & 0.0.6

- Small bug fixes from 0.0.4.

4.9 Version 0.0.4

- Add timeout to site test jobs if they are running too long or idle too long.
- Changed the site_post.py to use HTCondor's Python bindings rather than regular expressions.

4.10 Version 0.0.3

- Added plugin based output formation. The output class can now be specified in the configuration variable `outputtype`. The plugin should subclass the *stashcache_tester.output.generalOutput.GeneralOutput* class.
- Adding Git output plugin to upload summarized data to a github repo. It's further documented at *stashcache_tester.output.githubOutput.GithubOutput*

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

S

G

M

S