Getting Started with the Splunk HPC App

Jon Stearley
Computer Science Research Institute
Sandia National Laboratories[1]
jrstear@sandia.gov

I. Introduction

The "HPC" Splunk App facilitates the analysis of logs from high performance computing (HPC) systems. Developed by Sandia National Laboratories, it employs a wide variety of Splunk features, providing helpful tools to both system administrators and researchers. This document provides rough notes on how to *get started* with set-up and use; this is *not* a polished or exhaustive manual.

II. Setup

A. Getting Data In

1) One Index Per System: One index should be setup for each system (each of which might consist of thousands of nodes). The indexes should be named `hpc_SYSTEM` where `SYSTEM` is a name (e.g. RedSky, Cielo, BlueWaters, etc). Several dashboards assume this naming convention and list systems based on existing index names.

2) Sourcetypes: Parsing rules for several sourcetypes have been described in `props.conf`. In the below list, the first word is the `sourcetype` name.

- moabstats - Arguably the most important because job lookup tables are built using them, these logs are typically found in irritatingly-named files such as `/var/log/moab/stats/events.Mon_Oct_1_2012` on the MOAB node. They are documented at http://www.adaptivecomputing.com/resources/docs/mwm/16.3.3workloadtrace.php#workload. These are written at job submit, schedule, launch, signal, and completion.
- moab - MOAB control daemon logs, typically found in /var/log/moab/moabd.log.
- slurmctld - SLURM control daemon logs, typically found in `/var/log/slurm/slurmdctld.log` on the SLURMD node.
- joblog - SLURM job logs, typically found in `/var/log/slurm/joblog` on the SLURMD node. These are only written when a job ends.
- cray - CRAY XT3/XE6/etc event logs, typically found in files like `/craylog/eventlog.301.00089` on the SMW node. Parsing of these include index-time transformations to get the time and host names right.

B. Lookups

The first word indicate the `lookup` table in Splunk (eg `| lookup job`).

- job - the `jobid` each node is running at any given time. There is only entry at job start, and one at job end, such that if no job is running on a job (at the time of the event being viewed), the `jobid` field will be empty. This is updated every five minutes via the `Admin: updateJobLookup` scheduled search. It is reset to a window of two weeks, every night, via the `Admin: resetJobLookup` scheduled search.
- jobstart - this is the same as `job` above, except only the job start entries are present. The effect is that when this lookup table is used, the value of the `jobid` field is the jobid which most recently started on the node (relative to the event being viewed). It updated/reset at the same time as the `job` lookup table, using the same scheduled searches. **Automatic lookups and reports use this lookup (instead of job).**
- nodes - this optional lookup associates extra stuff with node names. It must be manually created, for which `bin/genders2csv` may be useful (exposes all available genders info to Splunk, so it can be looked up for reports or searches (via reverse lookups, which are very nifty!). On a CRAY, `bin/xtprocadmin2csv` serves a similar purpose. The resulting `nodes.csv` files provides a convenient means to report or search on physical, nid, or host name spaces (as well as role, X/Y/Z coordinates, etc).
- hostlist - this is a scripted lookup based on ttp://www.nsc.liu.se/~kent/pyton-hostlist/, providing expansion and compression of hostlist strings common among SLURM, pdsh, powerman, genders, and MOAB (on CRAYs). See the `job` macro for an example of its use, eg `| lookup hostlist short AS hosts OUPUT long` would do hostlist expansion of the `hosts` field.

C. Tracking Host States (Component Operations Status, COS)

This `app` includes an **experimental** custom command `statechange` to track host states. See [1] for more details. The `summary` index is updated via the `hostStateChanges` saved search (scheduled every 16 minutes), which the searches in the `COS` menu draw from in order to report on things like mean time to failure, etc.

[1] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

1) Backfilling the Summary Index: The `bin/backfill_statechange.pl` script is useful for backfilling the summary index. To use it, historic data should be indexed into Splunk, the `hostStateChanges` scheduled search disabled, the backfill script run, then `hostStateChanges` enabled.

2) Defining your State Machine: To modify the existing, or create your own, state machine logic, simply modify/create eventtype's having a name format of `cos_oldState-newState` where `oldState` the the state being transitioned from, and `newState` is the state being transitioned to.

III. The “Summary” Dashboard

This dashboard is the front page of the HPC app.

A. Logs in a job

To see all the events in a certain `jobid`, enter it into the second pane and hit return. It uses the `job` macro underneath (eg `` `job(jobid)` ``), which searches for the most recent event with `jobid` in `sourcetype=moabstats`, identifies the start and end times, and the set of hosts the job ran on, and expands it via a subsearch. See [2] for more details.

B. Hosts Of Interest

This pane shows hosts in an UnscheduledDowntime state. Clicking on a row brings up the logs for that host, from five minutes before it went down to one minute after. It is using the `hostdownwin(host,before,after)` macro underneath.

C. Messages of Interest

This pane shows hosts and how many of which types of events have occurred lately. Events reported are those having `tag=moi` (messages of interest). Click on a row to see those events for that host.

IV. Using Jobid's

Typical use is to see bad events on the Summary dashboard, click on a row to see them, and use workflow actions to then look at the other events in the job.

A. Workflow Action

The previously described `jobid` field also has a workflow action. For a message having a non-empty `jobid` value, pull-down the jobid menu, and select the job macro.

B. MOI by job,host

This dashboard is a useful summary of what MOI are happening on what hosts, for what users. It is under the MOI menu in the black bar at top.

References

[1] J. Stearley, R. Ballance, and L. Bauman, “A State-Machine Approach to Disambiguating Supercomputer Event Logs,” in *Proceedings of the 2012 Workshop on Managing Systems Automatically and Dynamically*. USENIX, 2012.

[2] J. Stearley, K. Lord, and S. Corwell, “A State-Machine Approach to Disambiguating Supercomputer Event Logs,” in *Proceedings of the 2012 Workshop on Managing Systems Automatically and Dynamically*. USENIX, 2012.