Outline

SLURM HA Support

Event Types

STRIGGER Mechanism

BULL Usage
HA Support is an Integral Part of SLURM

- Monitoring of System Events
  - periodic checks
  - explicit service calls
  - status checking for communication attempts

- Notification Mechanism
  - Recognition of condition 'triggers' event notification

- strigger Command for Event Notification
  - must be slurmuser
    --set register for event notification
    --get displays all registered triggers
    --clear clears a specific trigger
Types of Events Monitored

- Node
- Job
- slurmctld
- slurmdbd
- database
- front_end
- block_error
STRIGGER Commands

--set register for event notification

- specify the event type
- options
- location of program

--get displays registered event triggers

- active triggers (default)
- options for filtering
  (> strigger --get -j 41)

--clear delete a registered event trigger

- trigger id
- jobid
- user
  (> strigger --clear --user=slurm)
STRIGGER Event Types

Node Events

- **n, --node[=host]**

- **d, --down** node **DOWN**

- **D, --drained** node becomes **DRAINED**

- **F, --fail** node is **FAILING**

- **I, --idle** node remains **IDLE**

- **o, --offset=#** trigger's **offset time** from event

- **u, --up** node returned from **DOWN** state

- **r, --reconfig** **configuration** changes
STRIGGER Event Types

front_end  FrontEnd node state changes

- BlueGene and Cray architectures \textit{only}

- \texttt{-d, --down}  node DOWN
- \texttt{-u, --up}  node returned DOWN state

block_err  block error

- BlueGene architecture \textit{only}
STRIGGER Event Types

Job Events

- `j, --jobid=id` specific jobid

- `f, --fini` finish

- `t, --time` time limit

- `d, --down` node went DOWN
  - node (default)
  - --front_end

- `u, --up` node returned from DOWN state
  - node (default)

- `F, --fail` expected node failure
STTRIGGER Event Types

Slurmctld Events

- a, --primary_slurmctld_failure
- A, --primary_slurmctld_resumed_operation
- b, --primary_slurmctld_resumed_control
- B, --backup_slurmctld_failure
- c, --backup_slurmctld_resumed_operation
- C, --backup_slurmctld_assumed_control
- e, --primary_slurmctld_acct_buffer_full
STRIIGGER Event Types

Slurmdbd Events

-\texttt{g, --primary_slurmdbd_failure}

-\texttt{G, --primary_slurmdbd_resumed_operation}

Database Events

-\texttt{h, --primary_database_failure}

-\texttt{H, --primary_database_resumed_operation}
STRIGGER Mechanism

- Event notification registration with **strigger** command
  - identifies the event **type**
  - options
  - designate **program** location

- **Script** for setting multiple triggers

- Multiple triggers for the same event

- Condition detection sets notification flag

- Trigger processing isn't immediate
  - periodic check of event notification flags
  - multiple events can occur in the same interval
  - processing clears registration
STRIGGER Mechanism

- Trigger program executed once in interval

- Program executed on node used by slurmctld daemon

- Program responsible for logging of event information

- Program responsible for trigger reset

- Triggers saved in State Save Directory by SLURM

- System restart restores triggers
BULL Usage of STRIGGER

• Directory containing trigger programs
  –

• Cluster start invokes trigger registration script

• One `strigger` command for each event
  – identifies the event `type`
  – options
  – designate `program` location
BULL Usage of STRIGGER

- BULL cluster Event Manager monitors an event log
  - Customized version of SEC (Simple Event Correlator)
  - Contains event action rules

- Event Manager logs data in cluster database

- Rule based event actions
  - Start program execution
  - Send messages

- Event priority *escalation* supported
BULL Usage of STRIGGER

program that registers triggers

#!/bin/bash
#
## set the SLURM event triggers
#
# set trigger for primary_slurmctld_failure
strigger --set -a -p /etc/slurm/triggers/triggerascript.sh
# set trigger for primary_slurmctld_resumed_operation
strigger --set -A -p /etc/slurm/triggers/triggeraascript.sh
# set trigger for primary_slurmctld_resumed_control
strigger --set -b -p /etc/slurm/triggers/triggerbscript.sh
# set trigger for backup_slurmctld_failure
strigger --set -B -p /etc/slurm/triggers/triggerbbscript.sh
# set trigger for backup_slurmctld_resumed_operation
strigger --set -c -p /etc/slurm/triggers/triggerccscript.sh
# set trigger for backup_slurmctld_assumed_control
strigger --set -C -p /etc/slurm/triggers/triggerccscript.sh
# set trigger for primary_slurmctld_acct_buffer_full
strigger --set -e -p /etc/slurm/triggers/triggerescript.sh
# set trigger for primary_slurmdbd_failure
strigger --set -g -p /etc/slurm/triggers/triggergscript.sh
# set trigger for primary_slurmdbd_resumed_operation
strigger --set -G -p /etc/slurm/triggers/triggergscript.sh
# set trigger for primary_database_failure
strigger --set -h -p /etc/slurm/triggers/triggerhscript.sh
# set trigger for primary_database_resumed_operation
strigger --set -H -p /etc/slurm/triggers/triggerhscript.sh
# set trigger for any node going down
strigger --set -n -d -p /etc/slurm/triggers/triggerndscript.sh
BULL Usage of STRIGGER

- SLURM trigger fire invokes registered program
- Program gathers and formats event information
  - Uses Syslog Protocol standard track format
- Entry written to cluster event log
  - Uses `logger` command
- Program resets event trigger
BULL Usage of STRIGGER

program for trigger -a (primary_slurmctld_failure)

#!/bin/bash
#
# trigger a - primary_slurmctld_failure
#
datetime=$((date --rfc-3339=ns))
time=$(echo $datetime#*
)date=$(echo $datetime%*)
#
boslurmctld=$(cat /etc/slurm/slurm.conf |grep BackupController=)
boslurmctld=$(echo $boslurmctld##*BackupController=)
#
fqdn=$(ping -c1 $boslurmctld |grep PING)
fqdn=$(echo $fqdn#*PING)
fqdn=$(echo $fqdn%*)
#
slurmctld=$(cat /etc/slurm/slurm.conf |grep ControlMachine)
slurmctld=$(echo $slurmctld##*ControlMachine=)
#
logger "<27>1"$datetime $time $fqdn slurm backupslurmctld ID55 [Slurm@Bull
component="slurmctld"\eventType="failed" role="primary" slurm_nodename=""$slurmctld\"
#
# reset the fired trigger
#
strigger --set -a -p /etc/slurm/triggers/triggerascript.sh