Greenplum Workload Manager

Pivotal® Greenplum Database

Pivotal™

Agenda

- Workload Manager Overview
- Workload Management
  - Services and Agents
  - Actions, Conditions, and Rule Framework

Learning Objectives

- Learn how to Configure Limits
- Enable Workload Management

Workload Manager (WLM)

Workload Manager Motivation

- Enables operators to maximize value of investment
- When the system is under load, WLM can throttle processes or kill low-priority queries based on operator-defined rules
- Conserves system resources for high-priority operations

Workload Manager

- Tool to monitor and manage queries
  - Monitor Greenplum Database queries and host utilization statistics
  - Log when a query exceeds a threshold
  - Throttle the CPU usage of a query when it exceeds a threshold
  - Terminate or cancel a query
  - Detect memory, CPU, or disk I/O skew occurring during the execution of a query
  - Create detailed rules to manage queries

Workload Manager Docs

Workload Manager Installation

Run the gp-wlm.bin installer on the GPDB Master Server

- Installs on the GPDB Master Server
- Automatically distributes software to all Segment Servers
- When upgrading, automatically rolls back if upgrade fails

```
./gp-wlm.bin --install=<DIR> [ --force ] [ --install-concurrency=<COUNT> ]
[ --no-remove-old ] [ --skip-health-check ] [ --dbname-records=<database_name> ]
[ --tool-manifest=<FILE> ]
```

- Installation Sanity Check with "cluster-health-check" utility (automatically run by installer)

Pivotal™

Services

- Installs and runs five services
  - agent
  - cfgmon
  - rabbitmq
  - rulesengine
  - svcmon (service monitor)
- Command line **interactive** mode with "gp-wlm"
- Graphical interface (curses interface) "gptop”

Manage Services

- Manage with "svc-mgr.sh" where

```
 INSTALLDIR/gp-wlm/bin/svc-mgr.sh \
 --service=SVCNAME \
 --action=ACTION
SVCNAME = agent | cfgmon | rabbitmq | rulesengine | all
ACTION (defaults to local unless prefixed with "cluster-" )
ACTION = start | cluster-start
stop | cluster-stop
status | cluster-status
restart | cluster-restart
enable |cluster-enable
disable |cluster-disable
```

Using Workload Manager Rules

- Trigger Actions on events
- Rule specifies action to execute when specific conditions are detected
- Best Practice guideline – watch your logs first to determine what rules make sense for you
  - Problem queries
  - Throttle queries that consume too much CPU
  - Terminate queries that disrupt DB operations

Actions

- Acts like a "do <action-exp> WHEN <condition-exp>"
- Action Expressions
  - `gpdb_record`: record a custom message and details of the database query process in the gp_wlm_records database table
  - `host:throttle_gpdb_query:` throttle a Greenplum Database query on a specified host
  - `host:pg_cancel_backend:` cancel the current query in a session on a host by calling the PostgreSQL pg_cancel_backend() function
  - `pg_terminate_backend:` terminate a session by calling the PostgreSQL pg_terminate_backend() function

Conditions

- Greenplum scoped-datum
  - E.g. session_id:host:pid:runtime
    scoping = session_id:host:pid
    datum = runtime
- Example of an Action / Condition

  pg_terminate_backend() when session_id:host:pid:runtime > 120

- Datums are data items collected by the agent, and include operating system statistics, OS process statistics, and database query data

List of All Datums, scopes, and data formats

Rule framework

- Add Rule
  ```
  rule add [transient] <name> <action-name>(<action-args>)
  when <expression> [including <datum_list>]
  ```
- Show Rules – `rule show { all | rule-name }`
- Delete Rule - `rule delete rule-name`
- Modify Rule - `rule modify [transient] name action-name (action-args) when expression`
- Save Rule to Disk - `rule dump path`
- Import Rules from Disk - `rule import path`
- Restore Rules from Disk - `rule restore path`

Example Rules

```
log_query gpdb_record(message="Query > 10 seconds") when session_id:host:pid:runtime > 10 and session_id:host:pid:current_query != '<IDLE>'

kill_query pg_terminate_backend() when session_id:host:pid:runtime > 1200 and session_id:host:pid:current_query != '<IDLE>'

throttle_q host:throttle_gpdb_query(max_cpu=10) when session_id:host:pid:current_query =~ /.*select count.*/

record_skew gpdb_record(message="Skew") when session_id:disk_write_bytes_per_sec_skew >= 90 and session_id:host:pid:usename =~/.*/ and (session_id:host:pid:current_query =~ /.*insert.*/ or session_id:host:pid:current_query =~ /.*create.*select./)
```

Wrapping Up

In this module we covered:

- An overview of Workload Manager
- The different services and agents that are installed
- The concept of Actions and Conditions
- And, how to set up rules