LDMS User’s Group Meeting

August 5, 2019

Agenda

- Job Information Spank Plugin
- Job Information LDMS Sampler
- Multi-Tenant Spank Plugin
- Multi-Tenant Slurm LDMS Sampler
- Multi-Tenant Store LDMS Plugin

Job Information Spank Plugin

- Implements a SPANK plugin
- Specified in the .../slurm/etc/plugstack.conf file
- Supports a single job on a compute
- Works in concert with the LDMS Job Information Sampler

SPANK Plugins

- Loaded by slurmd and slurmstepd
- Implement an API where each *well-known* function name is called by slurm/slurmstepd at key points in a job's lifetime
- It is permissible to simply not define API that the plugin is not interested in having called

Slurm SPANK API

slurmd

- call slurm_spank_init('local')
- call slurm_spank_job_prolog ()
- **fork/exec slurmstepd**
  - **wait**
- call slurm_spank_job_epilog()
- call slurm_spank_exit('local')

slurmstepd

- call slurm_spank_init ('remote')
- drop privileges (initgroups(), seteuid(), chdir())
- call slurm_spank_user_init ()
- for each task
  - fork ()
  - reclaim privileges
  - call slurm_spank_task_init_privileged ()
  - become_user
  - call slurm_spank_task_init ()
  - **execve** ('/your/program')
- reclaim privileges
- for each task
  - call slurm_spank_task_post_fork ()
- for each task
  - wait ()
  - call slurm_spank_task_exit ()
- call slurm_spank_exit ('remote')

Spank Plugin Gotchas

- Interfaces are called from different process contexts
  - There is a plugin instance for each process in the job on the node
- Maintaining global state in the plugin will not

LDMS jobinfo_slurm plugin

- Available in 4.0+
- Implements the slurm_spank_init and slurm_task_exit callbacks
- Writes to a file configured with the LDMS_JOBINFO_DATA_FILE environment variable
- Obtains job information from the spank_get_item() and spank_getenv() interfaces
- Writes key=value pairs to the configured text file
- Supports a single job per node

LDMS_JOBINFO_DATA_FILE

Variable	Slurm Item (spank_get_item)	Set by Event
JOB_ID	S_JOB_ID	slurm_spank_init
JOB_STEP_ID	S_JOB_STEPID	slurm_spank_init
JOB_STATUS	JOB_STARTED (1) | JOB_EXITED (2)	slurm_spank_init, slurm_spank_exit
JOB_APP_ID	0	
JOB_START	time()	slurm_spank_init
JOB_END	time()	slurm_spank_exit
JOB_EXIT	S_TASK_EXIT_STATUS	slurm_spank_exit
JOB_NNODES	S_JOB_NNODES	slurm_spank_init
JOB_LOCAL_TASK_COUNT	S_JOB_LOCAL_TASK_COUNT	slurm_spank_init
JOB_NCPUS	S_JOB_NCPUS	slurm_spank_init
JOB_NAME	slurm_getenv("SLURM_JOB_NAME")	slurm_spank_init
JOB_USER_ID	S_JOB_UID	slurm_spank_init
JOB_USER	user name via getpwuuid(S_JOB_UID)	slurm_spank_init

jobinfo_slurm Spank Configuration

- Location of the library is specified in the Slurm plugstack.conf file
- Example:

```
# /opt/slurm/etc/plugstack.conf
# required/optional     plugin-path     args
required /opt/ovis/lib64/ovis-ldms/libjobinfo_slurm.so
```

LDMS jobinfo Sampler Plugin

- Uses getenv(“LDMS_JOBINFO_DATA_FILE”) to locate data file
- Starts a thread to monitor updates to the job data file:
    - while (true)
        - inotify_add_watch(watch_fd, LDMS_JOBINFO_DATA_FILE, …)
        - read(watch_fd)
        - open(LDMS_JOBINFO_DATA_FILE)
        - read(LDMS_JOBINFO_DATA_FILE)
        - update jobinfo metric set
        - close(LDMS_JOBINFO_DATA_FILE)
- The plugin’s *sample* API is a no-op
- It is not necessary to configure an updtr to cause this sampler to work

LDMS jobinfo Keys/Metric Set Schema

Key in Text File	Metric Name	Metric Type
JOB_ID	job_id	LDMS_V_U64
JOB_STATUS	job_status	LDMS_V_U64
JOB_APP_ID	app_id	LDMS_V_U64
JOB_START	job_start	LDMS_V_U64
JOB_END	job_end	LDMS_V_U64
JOB_EXIT	job_exit_status	LDMS_V_U64
JOB_NAME	job_name	LDMS_V_CHAR_ARRAY[256]
JOB_USER_ID	user_id	LDMS_V_U64
JOB_USER	job_user	LDMS_V_CHAR_ARRAY[LOGIN_NAME_MAX/256]

LDMS jobinfo Configuration

```
load name=jobinfo
configure name=jobinfo \
    component_id=${COMPONENT_ID} \
    producer=${HOSTNAME} \
    instance=${HOSTNAME}/jobinfo \
    uid=0 gid=0
```

- The metric set schema name is “jobinfo”

Multi-Tenant Job Support - Goals

- Support more than a single job on a node
- Provide additional information to plugins, including:
  - Process ID for each and number of local tasks
  - Global task id for each local task (i.e. job ‘rank’)
  - Local node number and number of nodes for the job
  - User-defined application and plugin configuration information
- This allows plugins (e.g. papi) to attach perforsmance counters to Process ID

MT-Spank Plugin – Slurm Notifier

- Available in LDMS 4.3+
- Implements the spank_init, spank_task_privileged_init, spank_task_exit, and spank_exit interfaces
- Obtains job information from the spank_get_item() and spank_getenv() interfaces
- Uses the ldmsd_stream_publish() interface to notify LDMS plugins of Spank events on an authenticated LDMS transport
    - All events are JSON formatted text
- Configured in the plugstack.conf configuration file

Slurm Notifier Events

- All events are JSON text
- “init” – Sent when slurm_spank_init is called
- “task_init” – Sent when slurm_spank_task_init_privileged is called
- “task_exit” – Sent when slurm_spank_task_exit is called
- “exit” – Sent when slurm_spank_exit is called

Slurm Notifier “init” event

```
{"schema“ : "slurm_job_data",
  "event“ : "init",
  "timestamp“ : 1565009670,
  "data“ : {
    "job_id“ : 90215, "job_name“ : "run-3x9.sh",
    "nodeid“ : 0, "ncpus“ : 16, "nnodes“ : 3,
    "local_tasks“ : 9, "total_tasks“ : 27,
    "uid":1002, "gid":1002,
    "subscriber_data“ : {
        "papi_sampler“ : {
            "file“ : "/opt/ovis/etc/papi-config.json",
         },
         "instance_data“ :
            "MACHINE=orion NUM_NODES=3 NUM_TASKS=27 PART=ldms“
        },
    }
}
```

Slurm Notifier “task_init_priv” event

```
{
  "schema‟ : "slurm_job_data",
  "event‟ : "task_init_priv",
  "timestamp‟ : 1565009670,
  "data‟ : {
    "job_id‟ : 90215,
    "task_id‟ : 0, "task_global_id‟ : 0, "task_pid‟ : 25419,
    "nodeid‟ : 0,
    "uid‟ : 1002, "gid‟ : 1002,
    "ncpus‟ : 16, "nnodes‟ : 3,
    "local_tasks‟ : 9, "total_tasks":27
  }
}
```

Slurm Notifier “task_exit” event

```
{
  "event‟ : "task_exit",
  "timestamp‟ : 1565009911,
  "data‟ : {
     "job_id‟ : 90215,
     "task_id‟ : 8, "task_global_id‟ : 8,
     "task_pid":25419,
     "nodeid‟ : 0,
     "task_exit_status":0
  }
}
```

Slurm Notifier “exit” event

```
{
  "schema“ : "slurm_job_data",
  "event“ : "exit",
  "timestamp“ : 1565009911,
  "data“ : {
    "job_id“ : 90215,
    "nodeid":0
  }
}
```

Slurm Notifier Configuration

```
# /opt/slurm/etc/plugstack.conf
required /opt/ovis/lib64/ovis-ldms/libslurm_notifier.so \
      host=<LDMS-hostname> \
      xprt=<LDMS-xprt-name> \
      port=<LDMS-listen-port> \
      auth=[munge,ovis,none] \
      stream=<default is 'slurm'> \
      timeout=<give up wait time (default 5s)>
```

Multi-Tenant Slurm Sampler

- Subscribes to the “slurm” stream, receives stream events on a callback function *stream_recv_cb*
- JSON data is parsed by the ldmsd streams infrastructure and delivered to a callback function as a json_entity_t
- Processes each event in the stream, updating the job data in the metric set with the data from the events

MT-Slurm Metric Set Schema

Metric	Type	Description
component_id	U64_ARRAY[job_count]	An array of component_id, *job_count* is the maximum number of concurrent jobs configured for the sampler
job_id	U64_ARRAY[job_count]	An array of job_id
app_id	U64_ARRAY[job_count]	An array of app_id
job_slot_list_tail	U32	Index in job_slot_list of the most recently created job
job_slot_list	S32_ARRAY[job_count]	Array of job slots, the contents is the job_slot number 0..job_count. A -1 indicates the entry is unused.
job_state	U32_ARRAY[job_count]	State of each job: JOB_FREE, JOB_STARTING, JOB_RUNNING, JOB_STOPPING, JOB_COMPLETE
job_size	U32_ARRAY[job_count]	Array of job size for each job, i.e total task count
job_uid	U32_ARRAY[job_count]	Array of user-id for each job
job_gid	U32_ARRAY[job_count]	Array of group-id for each job
job_start	U32_ARRAY[job_count]	Array of job start times
job_end	U32_ARRAY[job_count]	Array of job end times

MT-Slurm Metric Set Schema - continued

Metric	Type	Description
node_count	U32_ARRAY[job_count]	Array of node counts for each job
task_count	U32_ARRAY[job_count]	Array of local tasks for each job
task_pid_0	U32_ARRAY[task_count]	Array of Process ID for each local task in the job
…		
task_pid_N	U32_ARRAY[task_count]	
task_rank_0	U32_ARRAY[task_count]	An array of global task id, i.e. rank for each local task in the job
…		
task_rank_N	U32_ARRAY[task_count]	
task_exit_status_0	U32_ARRAY[task_count]	An array of exit status for each local task in the job
…		
task_exit_status_N	U32_ARRAY[task_count]	

MT-Slurm Metric Set Example

```
# ldms_ls -h orion-01 -p 10000 -a munge -E .*/slurm -l
orion-01-10000/slurm: consistent, last update: Mon Aug 05 07:58:31 2019 -0500 [597745us]
D u64[]      component_id                                 10001,10001,10001,10001,10001,10001,10001,10001
D u64[]      job_id                                       90208,90211,90215,0,0,0,0,0
D u64[]      app_id                                       0,0,0,0,0,0,0,0
D u32        job_slot_list_tail                           2
D s32[]      job_slot_list                                0,1,2,-1,-1,-1,-1,-1
D u8[]       job_state                                    0x04,0x04,0x04,0x00,0x00,0x00,0x00,0x00
D u32[]      job_size                                     27,8,27,0,0,0,0,0
D u32[]      job_uid                                      1002,1002,1002,0,0,0,0,0
D u32[]      job_gid                                      1002,1002,1002,0,0,0,0,0
D u32[]      job_start                                    1565009319,1565009560,1565009670,0,0,0,0,0
D u32[]      job_end                                      1565009555,1565009660,1565009911,0,0,0,0,0
D u32[]      node_count                                   3,2,3,0,0,0,0,0
D u32[]      task_count                                   9,4,9,0,0,0,0,0
D u32[]      task_pid_0                                   24609,24615,24621,24627,24633,24639,24645,24651,24657,0,0,0,0,0,0,0
. . .
D u32[]      task_pid_7                                   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
D u32[]      task_rank_0                                  0,1,2,3,4,5,6,7,8,0,0,0,0,0,0,0
. . .
D u32[]      task_rank_7                                  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
D u32[]      task_exit_status_0                           0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
. . .
D u32[]      task_exit_status_7                           0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

MT-Slurm Sampler Configuration

```
load name=slurm_sampler
config name=slurm_sampler \
    producer=${HOSTNAME} \
    instance=${HOSTNAME}/slurm \
    component_id=${COMPONENT_ID} \
    job_count=4 \
    stream=slurm
```

- The MT-Slurm sampler *sample* function is a no-op, all actions are driven from the stream_recv_cb function
- There is no need to configure an updtr for the MT-Slurm Sampler plugin

MT-Slurm Store

- The MT-Slurm metric set contains sequences of metrics where each metric value is an array
- Storing this data directly would make analysis tedious and difficult
- The MT-Slurm Store converts arrays in the MT-Slurm metric set into multiple records in a SOS data store
- Multiple output formats supports:
- Summary - Stores two records for each job
  - One for “init” and one for “exit”
- Rank – Stores a record for each task in the job
  - One for each “task_init/task_exit” event in the stream

Example *Rank* output of SOS data

```
-bash-4.2$ sos_cmd -C /DATA15/orion/ldms-data -qS mt-slurm \
    -X job_rank_time -V job_id -V job_size -V task_rank -V component_id -V task_pid
job_id             job_size     task_rank    component_id       task_pid
------------------ ------------ ------------ ------------------ ------------
             89135           27            0              10001        47502
             89135           27            1              10001        47503
. . .
             89135           27            8              10001        47545
             89135           27            9              10002        33106
. . .
             89135           27           17              10002        33154
             89135           27           18              10003        47824
             89135           27           19              10003        47825
. . .
             89135           27           26              10003        47866
. . .
```

MT-Slurm Store Configuration

```
load name=store_slurm
config name=store_slurm path=${CONTAINER_PATH} \
    verbosity=RANK
strgp_add name=slurm plugin=store_slurm \
    container=${CONTAINER_NAME} \
    schema=mt-slurm
strgp_start name=slurm
```