

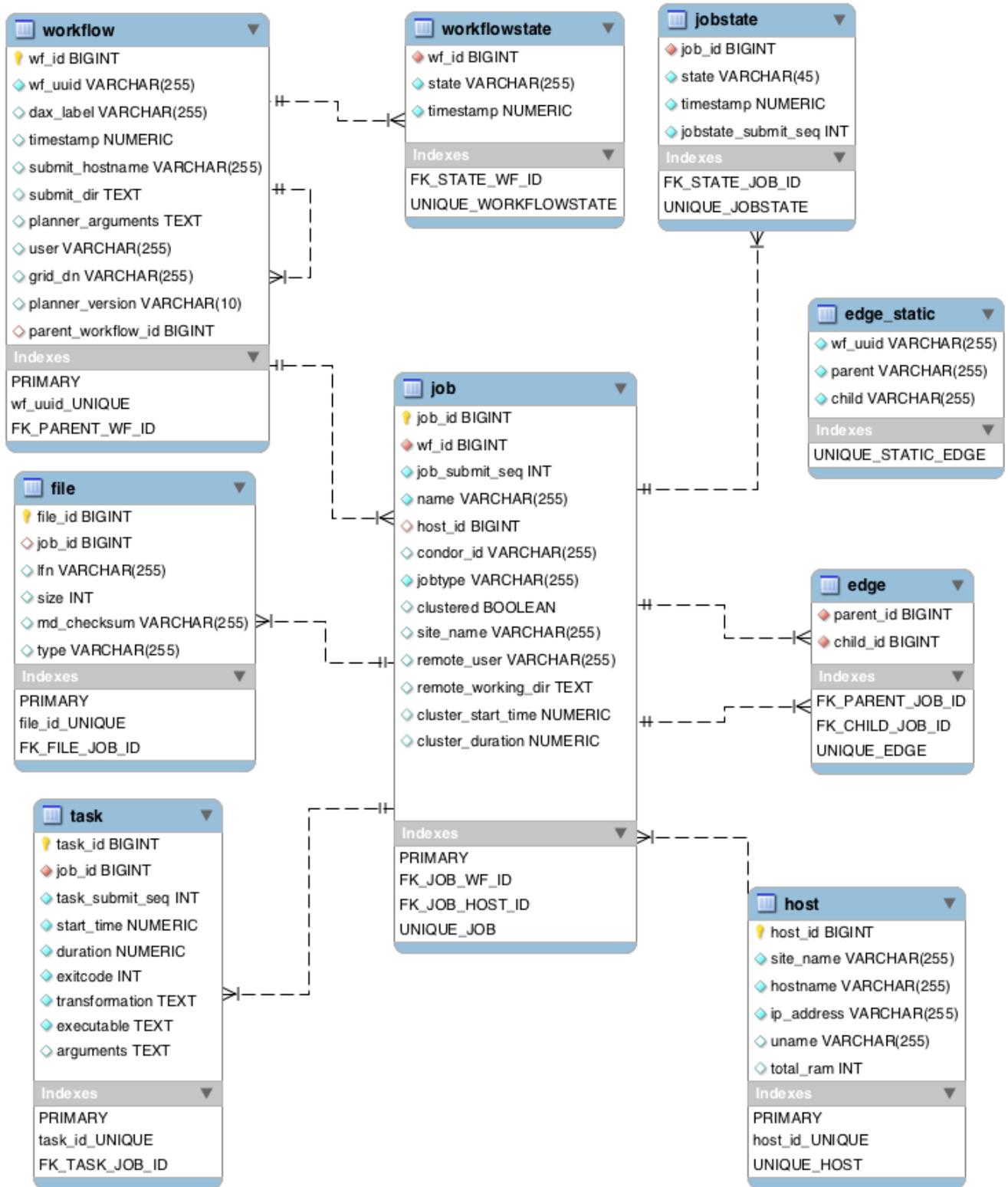
Stampede Data Model

Table Of Contents:

- 1 Stampede Database Schema For Pegasus 3.0
 - 1.1 Schema Picture
 - 1.1.1 Condensed Schema Picture
 - 1.2 Information Source for Each Table
 - 1.2.1 st_workflow
 - 1.2.2 st_job
 - 1.2.3 st_jobstate
 - 1.2.4 st_host
 - 1.2.5 st_task
 - 1.2.6 st_file
 - 1.3 Sample NetLogger Events
 - 1.3.1 workflow.plan event
 - 1.3.2 workflow.start event
 - 1.3.3 workflow.finish event
 - 1.3.4 job.prescript.start event
 - 1.3.5 job.prescript.finish event
 - 1.3.6 job.mainjob.start event
 - 1.3.7 job.mainjob.finish event
 - 1.3.8 job.postscript.start event
 - 1.3.9 job.postscript.finish event
 - 1.3.10 job.state event
 - 1.3.11 task.prescript, task.mainjob, task.postscript events
 - 1.3.12 host event

Stampede Database Schema For Pegasus 3.0

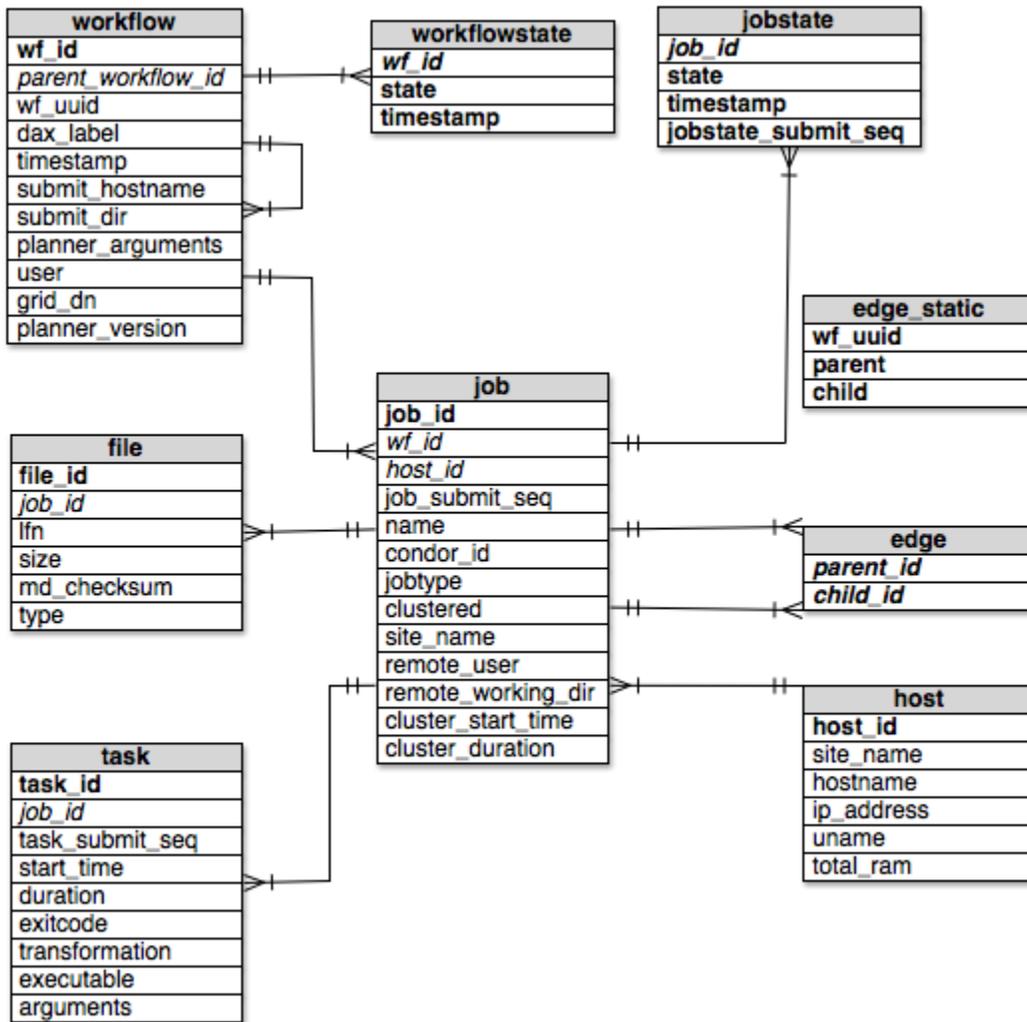
Schema Picture



[Download Mysql EER Workbench File](#)

[Download SQLite schema file](#)

Condensed Schema Picture



The entries that make up a primary key are in Bold.
Foreign Keys are in italics

[Download Omni Graffle File](#)

Information Source for Each Table

st_workflow

==> Information comes from braindump.txt file

wf_uuid = generated by pegasus-plan (currently missing)
 dax_label = label
 timestamp = pegasus_wf_time
 submit_hostname = (currently missing)
 submit_dir = run
 planner_arguments = (currently missing)
 user = (currently missing)
 grid_dn = (currently missing)
 planner version = pegasus version
 parent_workflow_id = wf_id of parent workflow

st_job

==> Information comes mainly from kickstart output file, but also from dagman.out file

job_id = autogenerated
job_submit_seq = integer <generated by tailstatd, and guaranteed to be unique within a workflow>
host_id = <hostname from invocation element>
name = <jobname from dagman.out file>
condor_id = <condor id from dagman.out file>
jobtype = <from .sub file pegasus_job_class>
clustered = boolean (true if jobname begins with merged_)
site_name = <resource from invocation element>
remote_user = <user from invocation element>
remote_working_dir = <cwd element>
cjob_start_time = (only for clustered job, struct entry of .out file, start)
cduration = (only for clustered job, struct entry of .out file, duration)

st_jobstate

==> Same information that currently goes into jobstate.log file, obtained from dagman.out file

job_id = from st_job table (autogenerated)
state = from dagman.out file (3rd column of jobstate.log file)
timestamp = from dagman.out file (1st column of jobstate.log file)

st_host

==> Information from kickstart output file

site_name = <resource, from invocation element>
hostname = <hostname, from invocation element>
ip_address = <hostaddr, from invocation element>
uname = <combined (system, release, machine) from machine element>
total_ram = <ram_total from machine element>

==> Note that old kickstart records may not have a machine element, need to handle it gracefully

st_task

==> Information comes from kickstart output file

task_id = autogenerated here
job_id = from st_job, autogenerated
start_time = <start from mainjob element>
duration = <duration, from mainjob element>
exitcode = <regular exitcode, from status element>
transformation = <transformation from invocation element>
arguments = <argument vector, joined by single space>

st_file

==> Information will come from kickstart output file

Sample NetLogger Events

As tailstatd parses the dagman.out file, it will generate NetLogger events that can be used to populate a database using the Stampede schema. All events have the "stampede." prefix. Here are examples for each of these events:

workflow.plan event

```
ts=2009-02-21T00:09:12.000000Z event=stampede.workflow.plan level=Info wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 submit_dir=/home/fabio/testing/dags/vahi/pegasus/montage/run0004 dax_label=montage planner_version=2.3.0cvs
```

This event is generated when tailstatd parses braindump.txt. The wf.id field is generated by Pegasus and is guaranteed to be unique. The ts field contains the timestamp the workflow was planned.

workflow.start event

```
ts=2010-02-20T23:09:13.000000Z event=stampede.workflow.start level=Info wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28
```

This event is generated by tailstatd when it detects that DAGMan has started. The ts field contains the timestamp DAGMan started.

workflow.finish event

```
ts=2010-02-20T23:25:28.000000Z event=stampede.workflow.finish level=Info wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28
```

This event is generated by tailstatd when it detects that DAGMan has finished. The ts field contains the timestamp DAGMan ended.

job.prescript.start event

```
ts=2010-02-20T23:25:28.000000Z event=stampede.job.prescript.start level=Info wf.id=wftest-id name=pegasus-plan_ID000001 job.id=2
```

This event is generated by tailstatd whenever it detects the start of a prescript for a new job. This event is similar to the job.mainjob.start event (see below), but it does not contain the condor_id field (as it is not yet assigned one). The ts field contains the timestamp the prescript started.

job.prescript.finish event

```
ts=2010-02-20T23:14:11.000000Z event=stampede.job.prescript.finish level=Info wf.id=wftest-id name=pegasus-plan_ID000001 job.id=2
```

This event is generated by tailstatd whenever it detects the end of a prescript for a job. The ts field contains the timestamp the prescript ended.

job.mainjob.start event

```
ts=2010-02-20T23:09:26.000000Z event=stampede.job.mainjob.start level=Info condor_id=3309.0 wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 name=create_dir_montage_0_viz_glidein job.id=1 jobtype=compute
```

The job.mainjob.start event is generated by tailstatd every time a job is found in the dagman.out file. The job.id tag is generated by tailstatd and starts in 1. The combination of wf_uuid and job.id guarantees a unique job. When a job begins, only certain information will be available. Later, when the job finishes, tailstatd will parse the kickstart output file and send the rest of the information in the job.mainjob.finish event (see below). The ts field contains the timestamp the main job started.

job.mainjob.finish event

```
ts=2010-02-20T23:14:06.000000Z event=stampede.job.mainjob.finish level=Info remote_user=vahi site_name=viz_glidein name=create_dir_montage_0_viz_glidein job.id=1 wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 condor_id=3309.0 remote_working_dir=/nfs/shared-scratch clustered=0 jobtype="create dir"
```

This event is generated by tailstatd whenever a main job finishes. It contains all the remaining information for the job table (which comes from the kickstart output file) that was unavailable at the beginning of the job execution. Note that jobtype now contains the correct value. The ts field contains the timestamp the main job ended.

job.postscript.start event

```
ts=2010-02-20T23:14:06.000000Z event=stampede.job.postscript.start level=Info wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 name=create_dir_montage_0_viz_glidein job.id=1
```

This event is generated by tailstatd when it detects the start of the postscript for a given job. The ts field contains the timestamp the postscript started.

job.postscript.finish event

```
ts=2010-02-20T23:14:11.000000Z event=stampede.job.postscript.finish level=Info wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 name=create_dir_montage_0_viz_glidein job.id=1
```

This event is generated by tailstatd when it detects the end of the postscript for a given job. The ts field contains the timestamp the postscript ended.

job.state event

```
ts=2010-02-20T23:14:06.000000Z event=stampede.job.state level=Info wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 state=EXECUTE name=create_dir_montage_0_viz_glidein job.id=1
```

A job.state event is generated every time a job changes state (e.g. SUBMIT, then EXECUTE, then JOB_SUCCESS, ...). The ts field contains the timestamp the job state changed.

task.prescript, task.mainjob, task.postscript events

```
ts=2010-02-20T23:14:06.000000Z event=stampede.task.mainjob level=Info executable=/nfs/software/pegasus/default/bin/dirmanager name=create_dir_montage_0_viz_glidein job.id=1 wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 task.id=1 start_time=1235175231 arguments="--create --dir /nfs/shared-scratch/vahi/exec/vahi/pegasus/montage/run0004" duration=0.078 transformation=pegasus::dirmanager exitcode=0
```

```
ts=2010-02-20T23:14:11.000000Z event=stampede.task.postscript level=Info executable=/nfs1/software/install/pegasus/default/bin/exitpost name=create_dir_montage_0_viz_glidein job.id=1 task.id=-2 wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 start_time=1266707646 arguments="- Dpegasus.user.properties=/nfs1/work/netlogger/dags/vahi/pegasus/montage/run0004/pegasus.15181.properties -e /nfs1/work/netlogger/dags/vahi/pegasus/montage/run0004/create_dir_montage_0_viz_glidein.out" duration=5 transformation=dagman::post exitcode=0
```

These three events are similar and indicate the termination of a prescript, a mainjob, or a postscript task (respectively). The ts field contains the timestamp the task ended. The task.id field contains the value -1 for prescript tasks, -2 for postscript tasks, and an integer (starting in 1) for each main job task.

host event

```
ts=2010-02-20T23:14:06.000000Z event=stampede.host level=Info site_name=viz_glidein name=create_dir_montage_0_viz_glidein hostname=viz-4.isi.edu job.id=1 wf.id=8bae72f2-31b9-45f4-bdd3-ce8032081a28 uname=linux-2.6.11.7-i686 ip_address=128.9.72.173 total_ram=2125164544
```

This event is generated by tailstatd whenever it parses a kickstart output file. In the case of clustered jobs (when there is more than 1 task in a mainjob), it is generated once per task. The ts field contains the timestamp the task associated with this host ended.