

# Pegasus statistics python version

- Introduction
- Pegasus Statistics Output
  - Workflow Summary
    - Workflow status ( Shows the last retry details)
    - Workflow statistics (Shows cumulative of all retries)
    - Workflow wall time :
    - Workflow cumulative job wall time :
    - Cumulative job walltime as seen from submit side:
  - Workflow statistics file
    - Workflow status ( Shows the last retry details)
    - Workflow statistics (Shows cumulative of all retries)
    - Workflow wall time :
    - Workflow cumulative job wall time :
    - Cumulative job wall time as seen from submit side:
  - Jobs statistics file
  - Transformation statistics file
- Pegasus Statistics Queries
  - Workflow Summary (Across workflow)
    - Workflow status ( Shows the last retry details)
      - Query for finding all the workflow id's( i.e all sub workflows and top level workflow ) by passing the wf\_uuid
      - Total jobs

## Introduction

pegasus-statistics is a command line tool for generating workflow execution statistics.

## Pegasus Statistics Output

pegasus-statistics generates the following statistics.

Workflow Summary :- Summary of the workflow run. If the given workflow has sub workflows it iteratively parses the sub workflow to generate the summary statistics. It is shown on the command line console.

Workflow statistics file :- A file containing statistics of individual workflows separated by their respective workflow uuid. If a given workflow has sub workflows it is considered as a single job, it won't iteratively parse the sub workflow. The file is named 'workflow.txt'

Job statistics file :- A file containing job statistics of individual workflows separated by their respective workflow uuid. The file is named 'jobs.txt'

Transformation statistics file :- A file containing transformation statistics of individual workflows separated by their respective workflow uuid. The file is named 'breakdown.txt'

The document uses the examples described below to explain statistics information.

**Note :** The example in Figure 1 is a diamond workflow with 4 tasks in the dax. And the pegasus plan creates three jobs in the dag with B2 and B3 clustered. During the execution of the workflow the clustered job fails after 3 retries . For the clustered job B2 runs , but B3 fails in all the retries.

The example in Figure 2 is a hierarchical work flow with 4 tasks in DAX A and 4 tasks in DAX B.A3 is sub workflow task.

The example in Figure 3 is a hierarchical work flow with 4 tasks in DAX A and 4 tasks in DAX B.However the A3 sub workflow tasks fails at the Prescript which results in DAX B workflow not getting planned . So the database is not populated with DAX B workflow details.

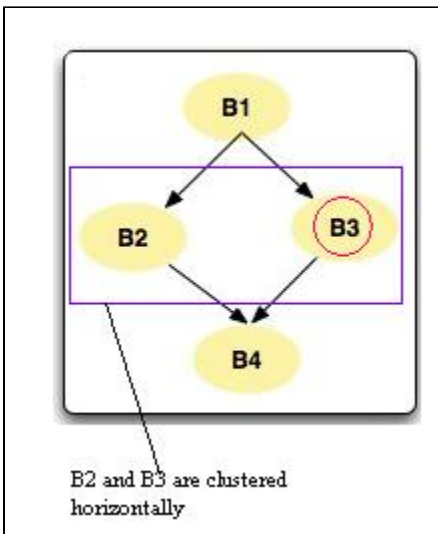



Figure 1 :- Diamond workflow [Failed Run]

# HIERARCHAL WORKFLOWS

INCREASING LEVEL OF RECURSION 

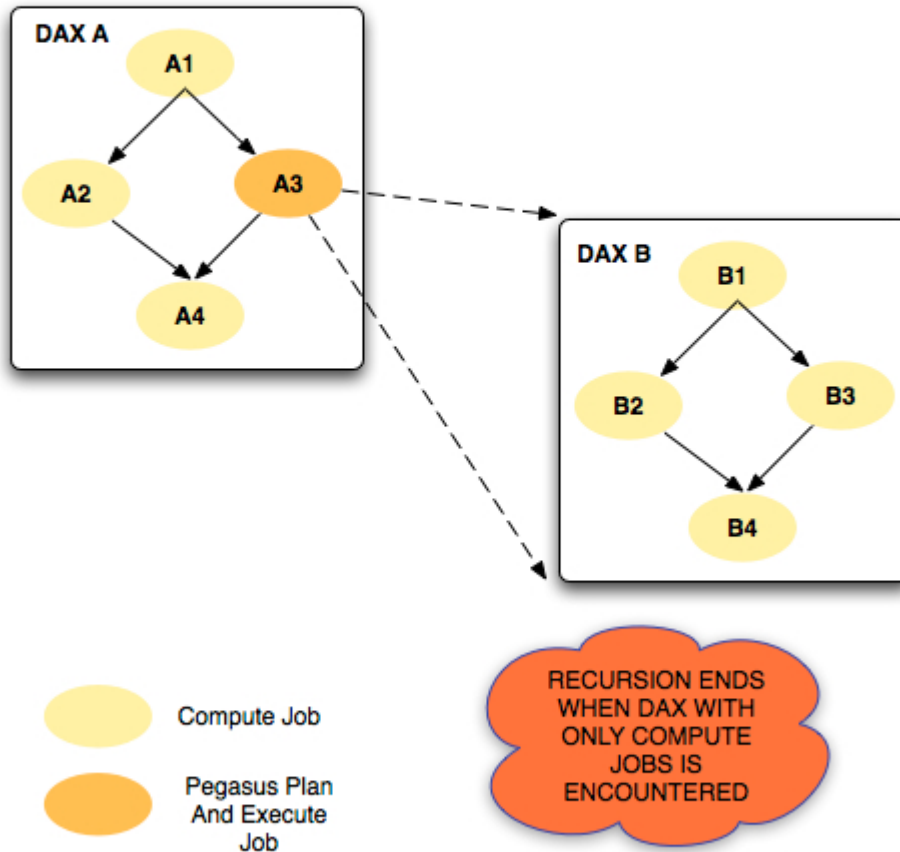


Figure 2 :- Hierarchal Workflow [Successful Run]

# HIERARCHAL WORKFLOWS

INCREASING LEVEL OF RECURSION →

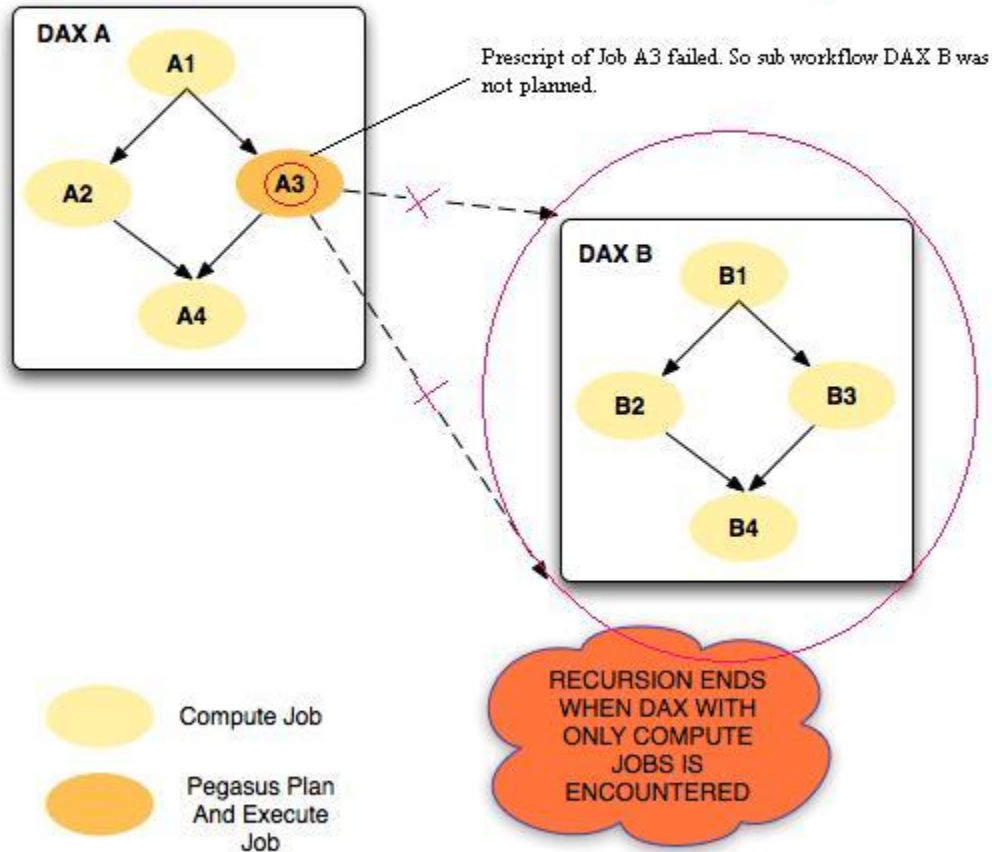


Figure 3 Hierarchal workflow [Failed Run]

## Workflow Summary

Workflow summary is summary of the statistics information of the workflow that is shown on the command line output . It recursively parses the sub workflow to generate the statistics information.

### Workflow status ( Shows the last retry details)

Workflow status table contains the information about the planned jobs and task.

The job information is obtained from the **jobs** table .Information about the job status i.e failed, succeeded etc is obtained from the **jobstate** table by looking at the state of the last retry.

The task information is obtained from the tasks table . Information about the task status is obtained from the **invocation** table . The query should combine task, **job** , **job instance** and **invocation** table using task\_id and job\_id.

**Note :** For workflow of workflows the original job count will include the jobs of the sub workflow if sub workflow was invoked. Otherwise the original job will consider sub dag or sub dax job as single job and the status as 'Failed' . i.e Only if entries corresponding to a sub workflow is present, the count of the workflow jobs will be added to the total original count. Tables below shows the workflow status for examples described above.

	Original	Succeeded	Failed	Unsubmitted	Unknown
Jobs	3	1	1	1	0
Tasks	4	2	1	NA	NA

Case 1 : Refer Figure 1 [Diamond Failed Run]

	Original	Succeeded	Failed	Unsubmitted	Unknown
Jobs	7	7	0	0	0
Tasks	8	8	0	NA	NA

Case 2: Refer Figure 2 [Hierarchical Successful Run]

	Original	Succeeded	Failed	Unsubmitted	Unknown
Jobs	4	2	1	1	0
Tasks	4	2	1	NA	NA

Case 3: Refer Figure 3 [Hierarchical Failed Run]

### Workflow statistics (Shows cumulative of all retries)

Workflow statistics table contains the information about the jobs and task actually executed during workflow run.

This information is obtained from the **job instance** and **invocation** table respectively. Tables below shows the workflow statistics for examples described above.

	Actually Run	Succeeded	Failed
Jobs	4	1	3
Tasks	7	4	3

Case 1 : Refer Figure 1 [Diamond Failed Run]

	Actually Run	Succeeded	Failed
Jobs	7	7	0
Tasks	8	8	0

Case 2 : Refer Figure 2 [Hierarchical Successful Run]

	Actually Run	Succeeded	Failed
Jobs	5	2	3
Tasks	5	2	3

Case 3 : Refer Figure 3 (B3 fails after 3 retries) [Hierarchical Failed Run]

### Workflow wall time :

The walltime from the start of the workflow execution to the end as reported by the DAGMAN. In case of rescue dag the value is the cumulative of all retries.

### Workflow cumulative job wall time :

The sum of the walltime of all jobs as reported by kickstart. In case of job retries the value is the cumulative of all retries. For workflows having sub workflow jobs (i.e SUBDAG and SUBDAX jobs), the walltime value includes jobs from the sub workflows as well. The value is obtained from the *remote\_runtime* in the **invocation** table

### Cumulative job walltime as seen from submit side:

The sum of the walltime of all jobs as reported by DAGMan. This is similar to the regular cumulative job walltime, but includes job management overhead and delays. In case of job retries the value is the cumulative of all retries. For workflows having sub workflow jobs (i.e SUBDAG and SUBDAX jobs), the walltime value includes jobs from the sub workflows as well. The value is obtained from the *local\_duration* in the **job\_instance** table

## Workflow statistics file

Workflow statistics file contains statistics information of each individual workflow. The parent workflow doesn't recursively calculate sub workflows jobs . Each sub workflow (SUB DAX, SUB DAG ) job is counted as a single job.

The information in these file is calculated similarly to the summary information . However, it is calculated only for the given workflow, if it has SUB DAX and SUB DAG jobs they are not recursively parsed.

**Note** : Job means the non sub workflow jobs.

### Workflow status ( Shows the last retry details)

	Original	Succeeded	Failed	Unsubmitted	Unknown
Jobs	4	2	1	1	0
SUB DAX	0	0	0	0	0
SUB DAG	0	0	0	0	0
Tasks	4	2	1	NA	NA

Case 1 : Refer Figure 1[Diamond Failed Run]

	Original	Succeeded	Failed	Unsubmitted	Unknown
Jobs	3	3	0	0	0
SUB DAX	1	1	0	0	0
SUB DAG	0	0	0	0	0
Tasks	4	4	0	NA	NA

Case 2 : Refer Figure 2 (DAX A workflow) [Hierarchal Successful Run]

	Original	Succeeded	Failed	Unsubmitted	Unknown
Jobs	3	2	0	1	0
SUB DAX	1	0	1	0	0
SUB DAG	0	0	0	0	0
Tasks	4	2	1	NA	NA

Case 3 : Refer Figure 3 (DAX A workflow ,B3 fails after 3 retries)[Hierarchical Failed Run]

### Workflow statistics (Shows cumulative of all retries)

	Actually Run	Succeeded	Failed
Jobs	4	1	3
SUB DAX	0	0	0
SUB DAG	0	0	0
Tasks	7	4	3

Case 1 : Refer Figure 1[Diamond Failed Run]

	Actually Run	Succeeded	Failed
Jobs	3	3	0
SUB DAX	1	1	0
SUB DAG	0	0	0

Tasks	4	4	0
-------	---	---	---

Case 2 : Refer Figure 2 (DAX A workflow ) [Hierarchical Successful Run]

	Actually Run	Succeeded	Failed
Jobs	2	2	0
SUB DAX	3	0	3
SUB DAG	0	0	0
Tasks	5	2	3

Case 3 : Refer Figure 3(DAX A workflow ,B3 fails after 3 retries)[Hierarchical Failed Run]

### Workflow wall time :

The walltime from the start of the workflow execution to the end as reported by the DAGMAN.In case of rescue dag the value is the cumulative of all retries.

### Workflow cumulative job wall time :

The sum of the walltime of all jobs as reported by kickstart. In case of job retries the value is the cumulative of all retries. For workflows having sub workflow jobs (i.e SUBDAG and SUBDAX jobs), the walltime value **doesn't** include jobs from the sub workflows . The value is obtained from the *remote\_runtime* in the **invocation** table

### Cumulative job wall time as seen from submit side:

The sum of the walltime of all jobs as reported by DAGMan. This is similar to the regular cumulative job walltime, but includes job management overhead and delays. In case of job retries the value is the cumulative of all retries. For workflows having sub workflow jobs (i.e SUBDAG and SUBDAX jobs), the walltime value **doesn't** includes jobs from the sub workflows . The value is obtained from the *local\_duration* in the **job\_instance**

### Jobs statistics file

Jobs file contains the following information about jobs in the individual workflow.

**Job** - the name of the job

**Site** - the site where the job ran

**Kickstart(sec.)** - the actual duration of the job in seconds on the remote compute node. In case of retries the value is the cumulative of all retries.The value is obtained from the *remote\_runtime* in the **invocation** table

**Post(sec.)** - the postscript time as reported by DAGMan .In case of retries the value is the cumulative of all retries. The value is calculated as [POST\_SCRIPT\_TERMINATED - POST\_SCRIPT\_STARTED/JOB\_TERMINATED].The information is obtained from **jobstate** table

**DAGMan(sec.)** - the time between the last parent job of a job completes and the job gets submitted.In case of retries the value of the last retry is used for calculation.The value is calculated as [SUBMIT] - last parent job's [POST\_SCRIPT\_TERMINATED] .The information is obtained from **jobstate** table

**CondorQTime(sec.)** - the time between submission by DAGMan and the remote Grid submission. It is an estimate of the time spent in the condor q on the submit node .In case of retries the value is the cumulative of all retries.The value is calculated as [GRID\_SUBMIT/GLOBUS\_SUBMIT/EXECUTE - SUBMIT].The information is obtained from **jobstate** table

**Resource(sec.)** - the time between the remote Grid submission and start of remote execution . It is an estimate of the time job spent in the remote queue .In case of retries the value is the cumulative of all retries.The value is calculated as [EXECUTE -GRID\_SUBMIT/GLOBUS\_SUBMIT].The information is obtained from **jobstate** table

**Runtime(sec.)** - the time spent on the resource as seen by Condor DAGMan . Is always >=kickstart .In case of retries the value is the cumulative of all retries.The value is obtained from the *local\_duration* in the **job\_instance**

**Seqexec(sec.)** - the time taken for the completion of a clustered job .In case of retries the value is the cumulative of all retries. This value is obtained from the *cluster\_duration* in the **job\_instance** table

**Seqexec-Delay(sec.)** - the time difference between the time for the completion of a clustered job and sum of all the individual tasks kickstart time .In case of retries the value is the cumulative of all retries. This value is obtained as the difference between the *cluster\_duration* in the **job\_instance** table and sum of all the *corresponding task's remote\_runtime* in the **invocation** table

### Transformation statistics file

The transformation statistics file contains the following information about each transformation in individual workflow.

**Transformation** - name of the transformation.

**Count** - the number of times the transformation was executed.

**Mean(sec.)** - the mean of the transformation runtime. The value is obtained from the *remote\_runtime* in the **invocation** table

**Variance(sec.)** - the variance of the transformation runtime. Variance is calculated using the on-line algorithm by Knuth ([http://en.wikipedia.org/wiki/Algorithms\\_for\\_calculating\\_variance](http://en.wikipedia.org/wiki/Algorithms_for_calculating_variance)).

**Min(sec.)** - the minimum transformation runtime value.

**Max(sec.)** - the maximum transformation runtime value.

**Total(sec.)** - the cumulative of transformation runtime.

## Pegasus Statistics Queries

This section contains the queries that are used for fetching the statistics information from the stampede DB.

### Workflow Summary (Across workflow)

#### Workflow status ( Shows the last retry details)

Query for finding all the workflow id's( i.e all sub workflows and top level workflow ) by passing the wf\_uuid

```
select wf_id
from
workflow as wf
where
wf.root_wf_id = (
    select wf_id from workflow where wf_uuid = '1e8b9ab6-8cdd-4e90-95cb-989f246dab56'
)
```

**Note:** All the queries uses the wf\_id which is obtained by passing the wf\_uuid to the above query. This will avoid the need for joining the workflow table to each query.

#### Total jobs

```
select
(
```