


BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF
ENERGY



Analysis of Job Traces from Carver, Hopper and Edison

May 2014

G. P. Rodrigo 2014 - gprodrigoalvarez@lbl.gov


Advanced Computing for Science

This work is one of the first steps to understand the challenges of scheduling in the next generation of super computers and other high performance systems. It intends to understand what is the current job submission behavior and trend in the scientific users at NERSC.

If any question comes out, refer it to:

gprodrigoalvarez@lbl.gov

iramakrishnan@lbl.gov



Machines summary

Machine	Edison	Hopper	Carver
Nodes	5,576 nodes	6,384 nodes	1,202 nodes
Cores	133,824 (24 cores per node)	153,216 (24 cores per node)	9,984 cores (Mixed)
CPU type	2 x Intel 'Ivy Bridge' 2.4-GHz processors	2 x AMD 'MagnyCours' 2.1-GHz processors	2.0Ghz-2.67Ghz Intel
Memory per node	64 GB DDR3 1333-MHz	32/64 GB DDR3 1333-MHz	24,48,48,1024 GB (mixed)
Scheduling System	Moab over Torque over Alps	Moab over Torque over Alps	Moab over Torque
Apps	Pure HPC. No server applications	Pure HPC. No server applications	Mixed Linux, MPI, High throughput, HPC
Allocation Granularity	Allocation per Node (Minimum granularity 24 cores)	Allocation per Node (Minimum granularity 24 cores)	Allocation per Core


G. P. Rodrigo 2014 - gprodrigoalvarez@lbl.gov

Advanced Computing for Science


This work analyzes the workload from 3 of the high performance systems at NERSC. Edison and Hopper represent a classical super-computer model. Carver is a example of a cluster approach to scientific computation.

Highlight on allocation:

- Edison only allocates full nodes.
- Carver and Hopper contains a subset of nodes that can be shared by different users at the same time. The rest are allocated to a single task at a time.



BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY



U.S. DEPARTMENT OF
ENERGY

Data Source Summary

- Execution logs from two supercomputers and one high-end cluster at NERSC:
 - Edison Cray XC30
 - Hopper Cray XE6
 - Carver IBM iDataPlex
- Logs generated from the different resource & workload management suites: Moab, Torque and Alps.
- Log files time-span & Tasks Scheduled:

– Edison:	01 Jan 2014 – 1 May 2014	474,361 tasks
– Hopper:	01 Jan 2013 – 31 Dec 2013	1,378,031 tasks
– Carver :	01 Jan 2013 – 31 Dec 2013	4,356,616 tasks
- Information captured/analyzed per task in this work: time dimension (running wall clock, wait time) and resource dimension (number of nodes, number of cores per node).
- Final analysis cut down to #Cores and Wallclock time.
- Regular queues have been grouped for the “Submission queue analysis”

G. P. Rodrigo 2014 - gprodrigo@lbl.gov


Advanced Computing for Science

All systems use MOAB (Scheduler) over Torque (Resource Manager). Edison and Hopper add an extra layer beneath to manage the nodes: ALPS (Cray system).


This work is based on the logs of Torque: register of every single job that ends.

Time analyzed: stable times for all machines. Edison was recently brought into production a good analysis case for user behavior formation.

Since this is a scheduling focused work it reduced the analysis to the geometrics in the jobs: length (Duration) and width (#cores allocated)




BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY



U.S. DEPARTMENT OF
ENERGY

Analysis summary



Workload analysis is focused on 2 variables for each task: **Wall-clock time & number of cores.**

- **Task Distribution:** Histogram analysis of the distribution of tasks by wall-clock time, number of cores and memory.
- **Queue analysis:** Characteristics of the tasks in each running queue.
- **Task analysis:** Clustering analysis of the tasks in the workload, defining groups of similar tasks under two variables: wall-clock time, and number of cores.
- **Wall clock accuracy:** Prediction vs. reality of tasks wall-clock.
- **Time Cycle Analysis:** How/when jobs are submitted.

Queue and cluster analysis are targeted towards understanding mapping of tasks and execution queues

G. P. Rodrigo 2014 - gprodrigoalvarez@lbl.gov

Advanced Computing for Science

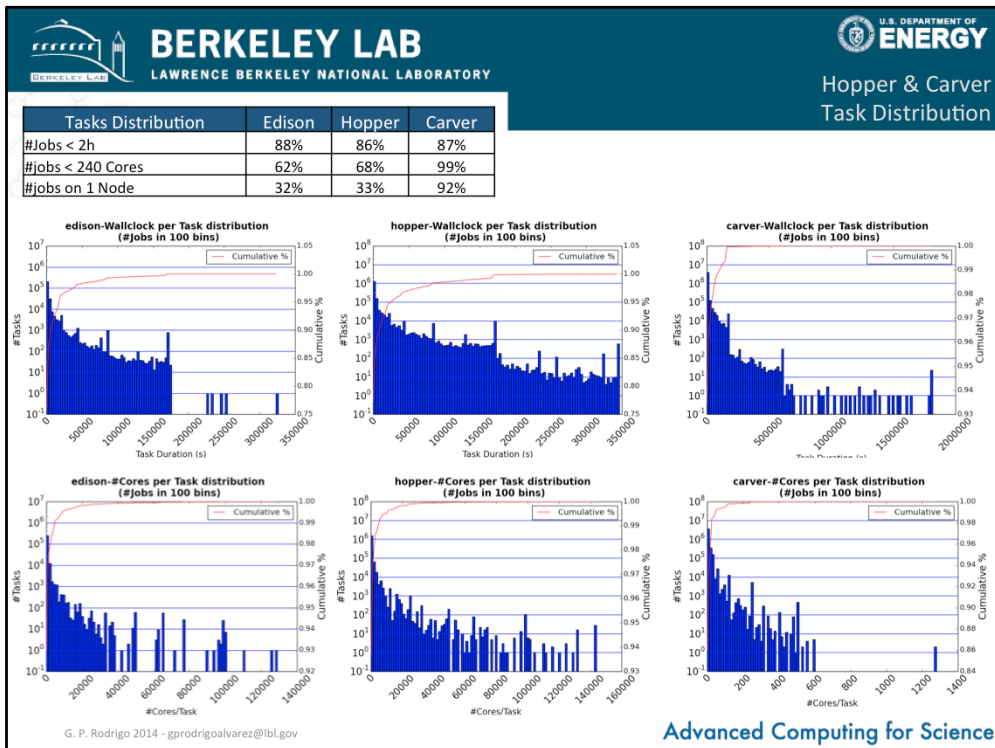
Clusters: as statistical clusters of jobs with similar length/width



- **Target: to group similar tasks, understanding how many groups (Clusters) are present in the workloads.**
 - Looking for patterns on **Wallclock time** and **#Cores** per task. (Relevant to scheduling)
- **Applied statistical analysis methods:**
 - Each task is represented by two component vectors.
 - K-Means clustering technique to group the similar tasks.
 - Final groups: no task should be more than 110% of the mean away from the centroid of the group.
- **Method:**
 - Based on a workload analysis method from Google ^[1]
 - Looking for the smallest set of uniform groups.
 - We consider groups to be uniform if all their points comply with: $(\text{Mean})/(\text{Std Deviation}) \leq 1.1$
- **Analysis:**
 - How are the Queues mapped on the Clusters?
 - Do queues contain similar jobs? Or are they mixed?

[1] Asit K. Mishra, Joseph L. Hellerstein, Walfredo Cirne, and Chita R. Das. 2010. Towards characterizing cloud backend workloads: insights from Google compute clusters. SIGMETRICS Perform. Eval. Rev. 37, 4 (March 2010)

Target: to build groups of similar jobs. Grouping

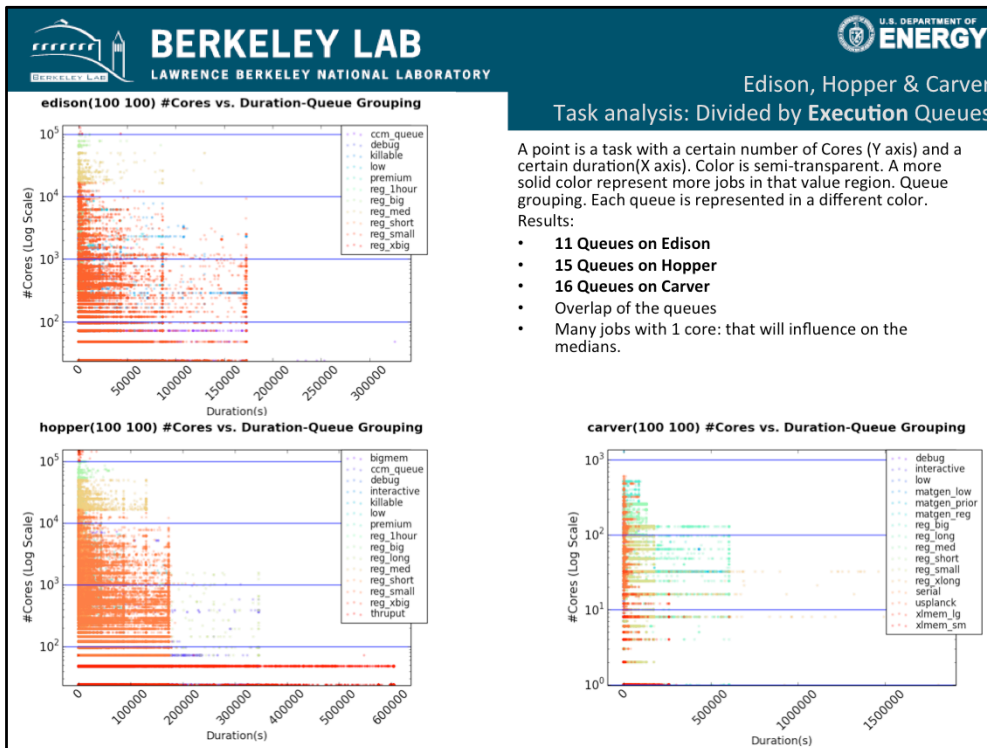


Distribution histograms on two variables: job duration and width.

Sample Explanation of the first graph:

- X axis: duration of a job in seconds
- Y axis: number of jobs
- Example: if there is a vertical bar in $x=50000s$ and $y=10^2$ it means that there were 100 jobs with that duration
- The red line is the aggregation of the "blue bars". If in $x=7200s$ $y=88\%$ it means that 88% of all jobs are 7200s or less seconds long.

The table on top gives some simple values that allow to understand in which duration and width values are the jobs concentrating.



Edison, Hopper & Carver
Task analysis: Divided by Execution Queues

A point is a task with a certain number of Cores (Y axis) and a certain duration(X axis). Color is semi-transparent. A more solid color represent more jobs in that value region. Queue grouping. Each queue is represented in a different color.

- Results:
- 11 Queues on Edison
 - 15 Queues on Hopper
 - 16 Queues on Carver
 - Overlap of the queues
 - Many jobs with 1 core: that will influence on the medians.

From this point and on, we consider 2 groups of queues:

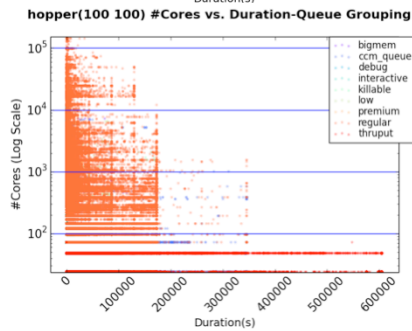
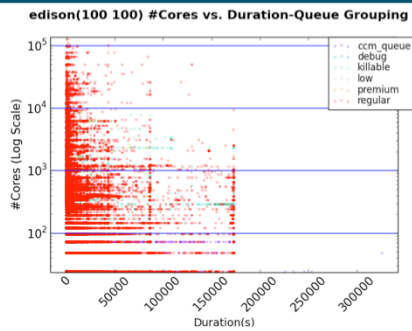
- Submission queues: seen by the users. They choose the queue.
- Execution queues: the system divides the submission queues (in some cases) in other queues. Jobs are classified by their geometric characteristics.

This slide looks into the execution queues.

Explanation of these graphs:

- Each color dot in the graph represents a task
- If a task belong to a queue its dot will be colored in the corresponding queue color.
- The colors are “transparent” so an area with more solid color implies more tasks than an area with “fainter” colors.
- Each dot is positioned depending on its geometry: X-axis corresponds to the job length (running time). The Y-axis corresponds to the width (#cores allocated)

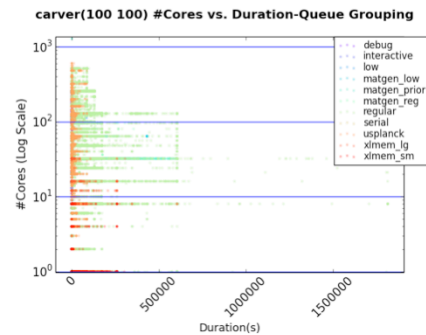
We can observe some division depending on the queues, but still queue jobs seem to overlap.



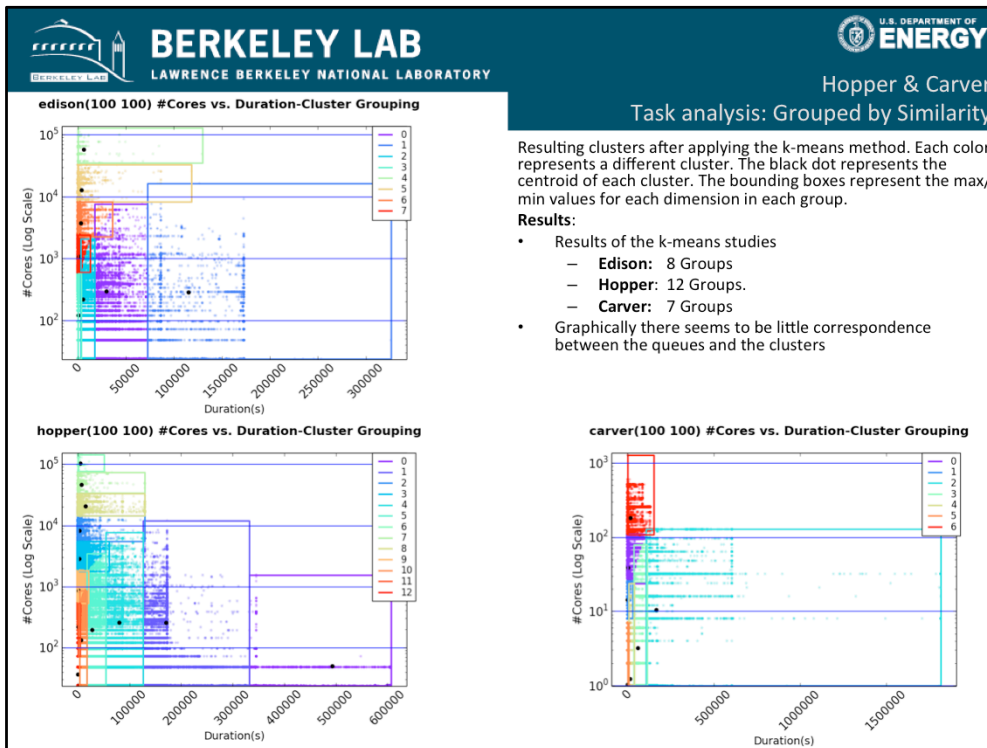
A point is a task with a certain number of Cores (Y axis) and a certain duration(X axis). Color is semi-transparent. A more solid color represent more jobs in that value region. Queue grouping. Each queue is represented in a different color.

Results:

- **6 Queues on Edison**
- **9 Queues on Hopper**
- **11 Queues on Carver**
- Overlap of the queues
- Many jobs with 1 core: that will influence on the medians.



Analysis was repeated on the submission queues.




This is the result of the previously described clustering exercise: group similar tasks. For each machine we could infer different numbers of clusters, indicating a difference on the heterogeneity of the job mix: more clusters found mean more diversity in the job mix of the system.

The graph representation is similar to the previous slides but:


- Now each color correspond to the detected clusters. Tasks are colored with the color of the cluster they are included in.
- Each “cluster” is surrounded by a bounding box indicating the max/min job length/width detected for the jobs contained. Bounding boxes may overlap as a both dimensions are used to assert if a task belongs to a certain cluster.

If we move backwards and forward between this slide and the previous ones, we can observe how the clustering offers a more clear separation between the tasks than the queues.

We consider that 2 jobs in the same clusters are more similar than 2 jobs which belong two different clusters.



BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY



Edison, Hopper & Carver
Task analysis: Mapping Queues over Groups

Queue Edison	50% of the queue mapped on just one cluster
ccm_queue	No
debug	Yes
killable	No
low	Yes
premium	Yes
Regular	Yes
1hour	Yes
big	Yes
med	Yes
short	No
small	No
xbig	Yes

Queue Hopper	50% of the queue mapped on just one cluster
bigmem	No
ccm_queue	Yes
debug	Yes
interactive	Yes
killable	No
low	No
premium	No
Regular	No
1hour	Yes
big	Yes
long	No
med	Yes
short	No
small	No
xbig	Yes
thruput	No

Queue Carver	50% of the queue mapped on just one cluster
debug	No
interactive	Yes
low	No
matgen_low	Yes
matgen_prior	No
matgen_reg	Yes
Regular	No
1hour	No
big	Yes
med	Yes
short	Yes
small	Yes
xbig	Yes
serial	Yes
usplanck	No
xlmem_lg	Yes
xlmem_sm	Yes

Why

- To which groups tasks in queues belong to?
- Tasks in queue should be mapped on the same group (Or mostly in the same cluster).

The results:

- Edison:** 4 out of 6 user queues have 50% of their tasks mapped on the same group. (7 out of 11 for the execution queues)
- Hopper:** 3 out of 9 user queues have 50% of their tasks mapped on the same group. (7 out of 14 for the execution queues)
- Carver:** 6 out of 11 user queues have 50% of their tasks mapped on the same group. (11 out of 19 for the execution queues)
- Some groups are a "workload mix": Debug, Interactive, ..
- More diversity mapping in Hopper.

G. P. Rodrigo 2014 - gprodrigo@lbl.gov

Advanced Computing for Science

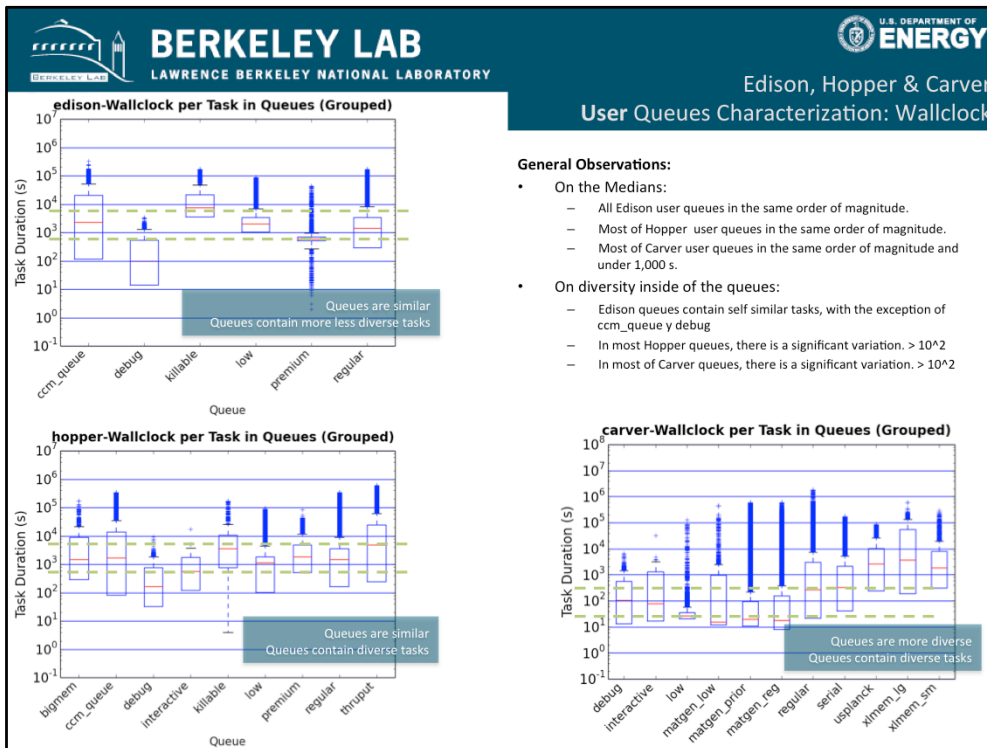
Still, it is hard to get the real dimension of how the queues are mapped over the clusters. The whole idea is to understand two different points:

- Are the jobs inside of the queues similar between them?
- Are the queues different between them?

(Both things are desired for the scheduling system to make good quality allocation decisions).

For this we made a simple measure:

- If we look into the jobs of a queue... on how many different clusters are these jobs coming from? If a queue contains jobs from many different clusters, it will mean that it contains very diverse jobs. If the jobs in a queue are coming from 1 cluster, it will mean that the jobs contained are very similar.
- In the table in this slide we use the following criteria: if at least 50% of the jobs belong to the same cluster, we will determine that it is quite "uniform" and marked in green. Otherwise we will consider to have a certain heterogeneous job mix and mark it in red.



The next 4 slides are statistical analysis on the task duration and job width on: execution queues, user queues and clusters.

Let's explain the first graph on this slide:

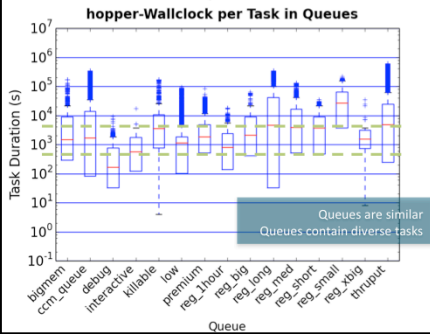
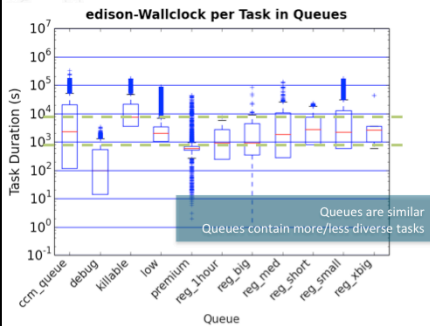
- X axis: one boxplot per queue
- Y axis: task duration
- The red line represents the median of the jobs on the queue: meaning that 50% of the jobs have that duration or less.
- The length of the box: represents how much variance is happening in that queue, if it is long it means that there is a big difference between the longest and shortest job.
- The blue crosses are outliers.

On understanding the graphs:

- If the medians of the queues/clusters are similar: the queues/clusters are quite similar between them
- If the boxplot for a queue/cluster are long: the jobs in that queue/cluster are quite diverse.

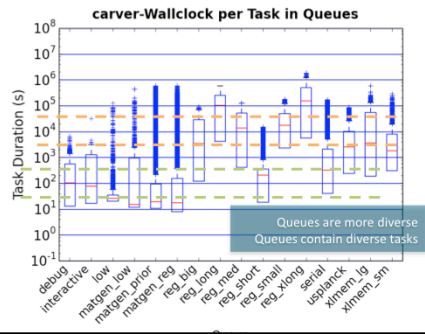
As a summary for the next slides: queues are quite similar between them, but containing an heterogeneous job mix. Clusters are more differentiated but containing more similar jobs.

Edison, Hopper & Carver
Execution Queues Characterization: Wallclock

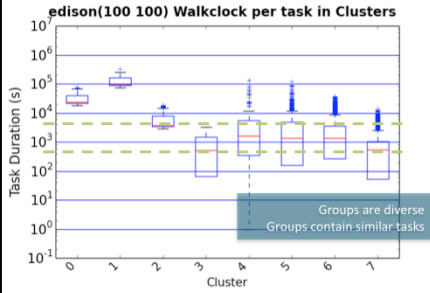


General Observations:

- On the Medians:
 - All Edison queues in the same order of magnitude.
 - Most of Hopper queues in the same order of magnitude.
 - Two differentiated groups.
- On diversity inside of the queues:
 - In Edison there is more diversity on the execution queues than in the user queues
 - In most Hopper queues, there is a significant variation. $> 10^2$
 - In most of Carver queues, there is a significant variation. $> 10^2$



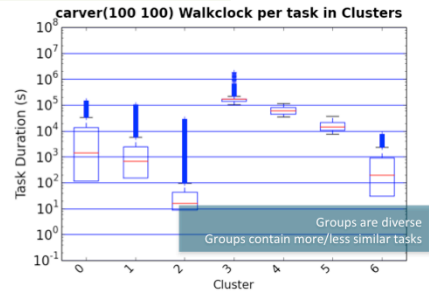
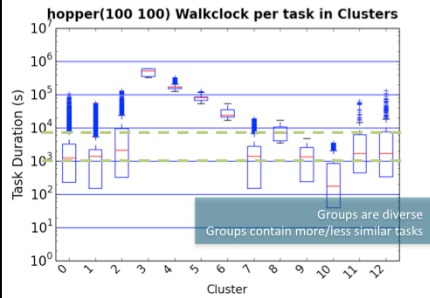
Edison, Hopper & Carver
Task Groups Characterization: Walkclock



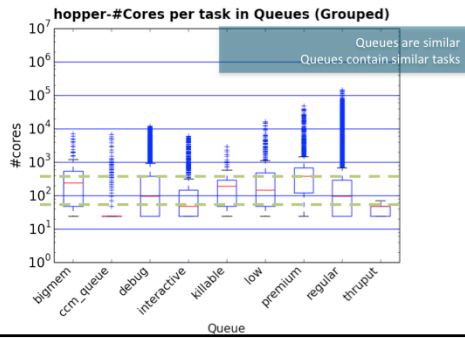
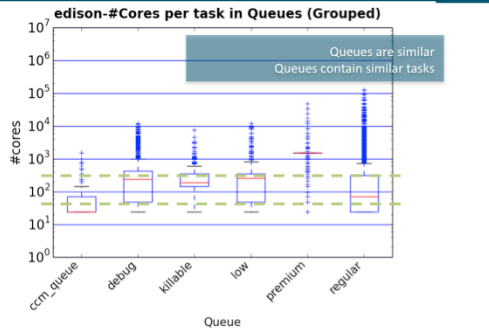
General Observations:

- On the Medians:
 - Edison groups have to ranges of medians.
 - Hopper groups have differentiated medians.
 - Carver groups have differentiated medians.
- On diversity inside of the groups:
 - Edison groups contain as diverse tasks as the queues.
 - In Hopper, diversity is reduced for 4 groups.
 - Most of Carver groups contain similar tasks.

Groups are more differentiated than queues.

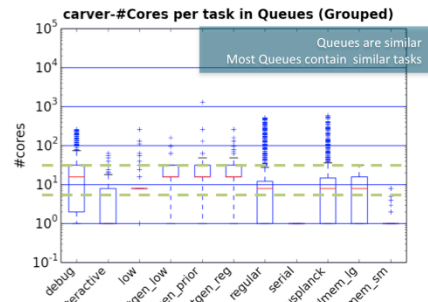


Edison, Hopper & Carver
User Queues Characterization: #Cores

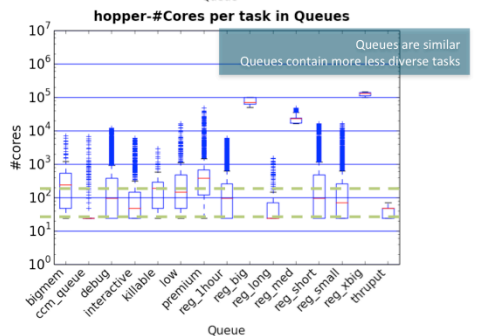
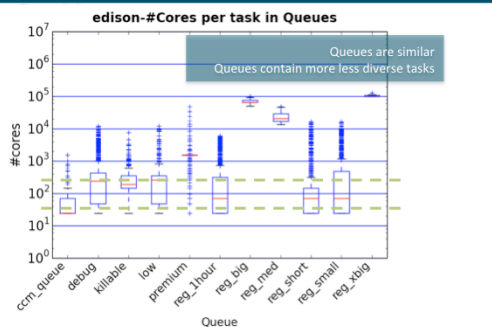


General Observations:

- On the Medians:
 - Most Edison user queues in the same order of magnitude.
 - Most of Hopper user queues in the same order of magnitude.
 - Most of Carver user queues in the same order of magnitude.
- On diversity inside of the queues:
 - In most Edison queues, there is not a great variation.
 - In most Hopper queues, there is not a great variation.
 - In some of Carver queues, there is a significant variation.

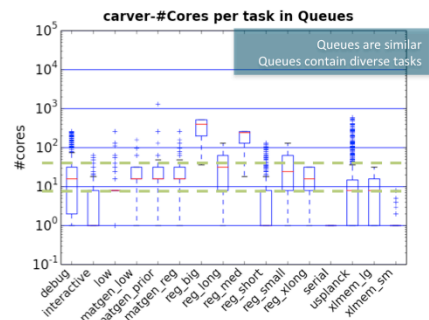


Edison, Hopper & Carver
Execution Queues Characterization: #Cores

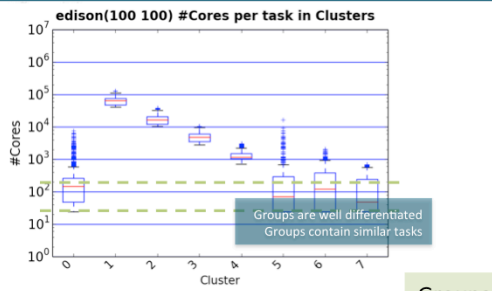


General Observations:

- On the Medians:
 - Most Edison user queues in the same order of magnitude.
 - Most of Hopper user queues in the same order of magnitude.
 - Most of Carver user queues in the same order of magnitude.
- On diversity inside of the queues:
 - In most Edison queues, there is a significant variation. > 10²
 - In most Hopper queues, there is a significant variation. > 10²
 - In most of Carver queues, there is a significant variation. > 10²



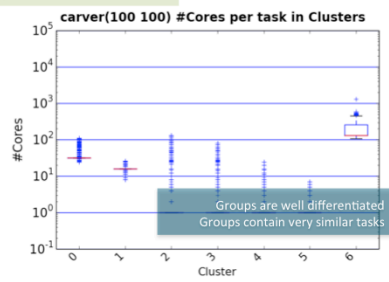
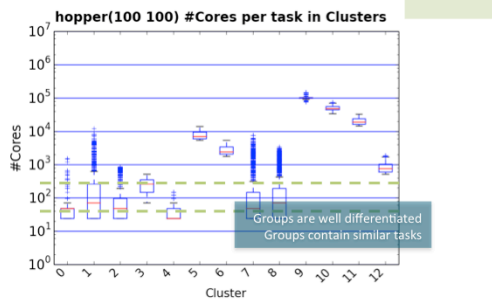
Edison, Hopper & Carver
Task Groups Characterization: #Cores

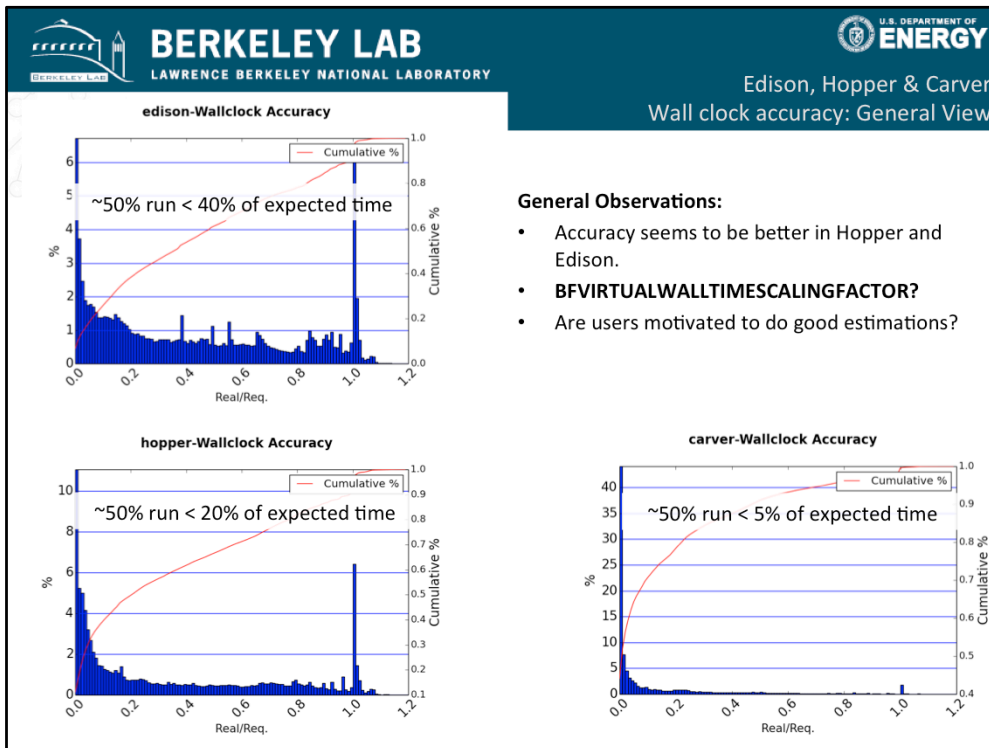


General Observations:

- On the Medians:
 - In Edison 4 groups have similar medians while another 4 are well differentiated.
 - In Hopper only 4 groups have similar medians
 - 4 Groups have 1 core as the median.
- On diversity inside of the groups:
 - In Edison, tasks in the same queues are similar.
 - In Hopper, tasks in the same queues are similar.
 - In Carver, tasks in the same queues are similar.

Groups are more differentiated than queues.





General Observations:

- Accuracy seems to be better in Hopper and Edison.
- **BFVIRTUALWALLTIMESCALINGFACTOR?**
- Are users motivated to do good estimations?


Users require a maximum running time for their jobs (wall clock time). This required wall clock time is used by the backfill functions to calculate what jobs could be forwarded in the queues to fill “execution gaps”.

If this prediction made for the user (if any) is far from reality, the quality of the backfill decisions will be affected.


Here we can observe the study of the the accuracy of the wall clock:
 $(\text{actual wall clock}) / (\text{required wall clock})$ A value close to 1 means that the predictions are close to the actual value (Supposed to be good). A value close to 0 means that the predictions are really far away (Supposed to be bad).

In this slide we do a distribution analysis of the wall clock accuracy: what are the chances (y axis) for the accuracy of a job to be certain value (x axis). Again the red line is the aggregated value=

The results suggest that accuracy is low, specially in carver.



BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY



U.S. DEPARTMENT OF
ENERGY

Edison, Hopper & Carver
Wall clock accuracy: Queues

Edison Req. Wall Clock analysis

Queue	Q. Limit (s)	Median (s)	Median/Limit	Acc. (Med)
ccm_int	1800	1800	1,00	0,06
ccm_queue	172800	21600	0,13	0,19
debug	1800	1800	1,00	0,20
killable	172800	10800	0,06	0,67
low	86400	3600	0,04	0,65
premium	43200	3600	0,08	0,18
reg_big	129600	3600	0,03	0,46
reg_med	129600	10800	0,08	0,50
reg_small	172800	12599	0,07	0,35
reg_xbig	43200	3600	0,08	0,92

Carver Req. Wall Clock analysis

Queue	Q. Limit (s)	Median (s)	Median/Limit	Acc. (Med)
debug	1800	1800	1,00	0,27
interactive	1800	1800	1,00	0,45
low	86400	120	0,00	0,18
reg_big	86400	3600	0,04	0,39
reg_long	604800	360000	0,60	0,34
reg_med	129600	43200	0,33	0,74
reg_short	14400	600	0,04	0,45
reg_small	172800	57600	0,33	0,31
reg_xlong	1814400	43140	0,02	0,19
serial	172800	86400	0,50	0,03
xlmem_lg	259200	90000	0,35	0,02
xlmem_sm	259200	21600	0,08	0,43

Hopper Req. Wall Clock analysis

Queue	Q. Limit (s)	Median (s)	Median/Limit	Acc. (Med)
bigmem	86400	8400	0,10	0,22
ccm_int	1800	1800	1,00	0,17
ccm_queue	345600	41400	0,12	0,28
debug	1800	1800	1,00	0,14
interactive	1800	1800	1,00	0,48
killable	172800	5400	0,03	0,69
low	86400	3600	0,04	0,44
premium	43200	5400	0,13	0,57
reg_1hour	3600	3600	1,00	0,35
reg_big	129600	14400	0,11	0,20
reg_long	345600	259199	0,75	0,53
reg_med	129600	9000	0,07	0,68
reg_short	21600	8400	0,39	0,51
reg_small	172800	86400	0,50	0,50
reg_xbig	43200	5400	0,13	0,08
scavenger	21600	14400	0,67	0,90
thruput	604800	6599	0,01	0,03
xfer	43200	7200	0,17	0,03

Advanced Computing for Science

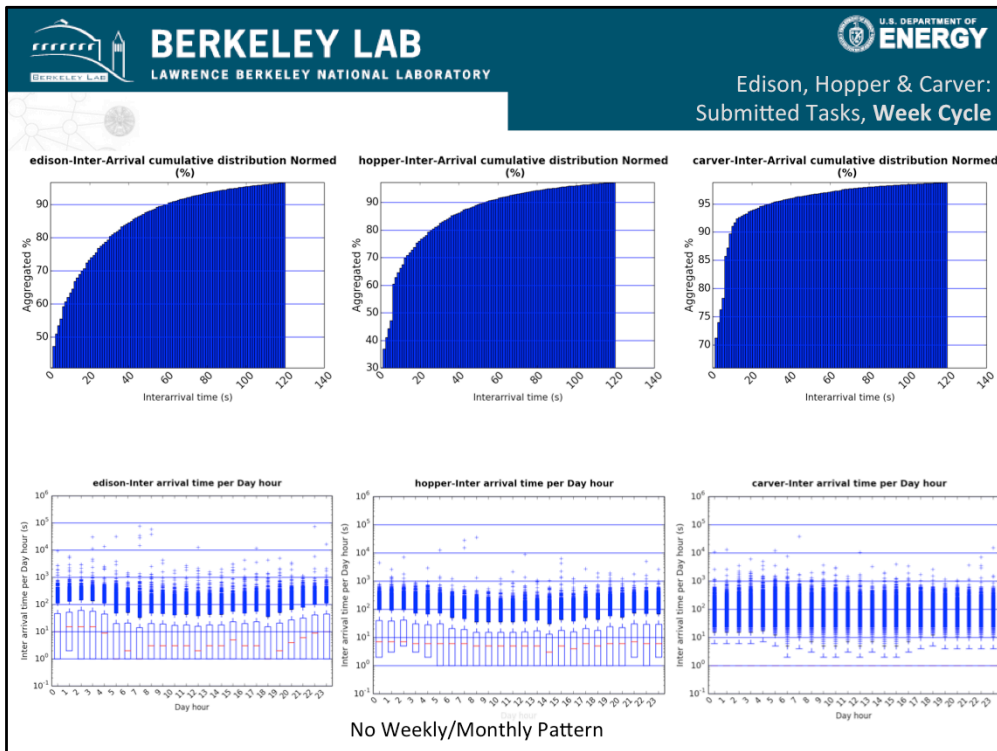
G. P. Rodrigo 2014 - gprodrigo@lbl.gov

After looking into the overall accuracy we looked into the accuracy in jobs in each queue.

This is how to interpret these tables columns:

- Queue: queue studied
- Q. Limit(s) Limit in wall clock imposed by the submission system. Jobs submitted cannot have a required wall clock bigger than that.
- Median (s): Median of the requested wall clock values of the jobs in the queue. If it is 1800 it means that at least 50% are equal or under that value.
- Media/Limit: We divide the two previous columns. This value gives an ide of how close are the requested times to the queue limit. A value of 1 would mean that the users use the queue limit as their default wall clock request for jobs.
- Acc. (Med): median on the accuracy of the jobs in the queue.

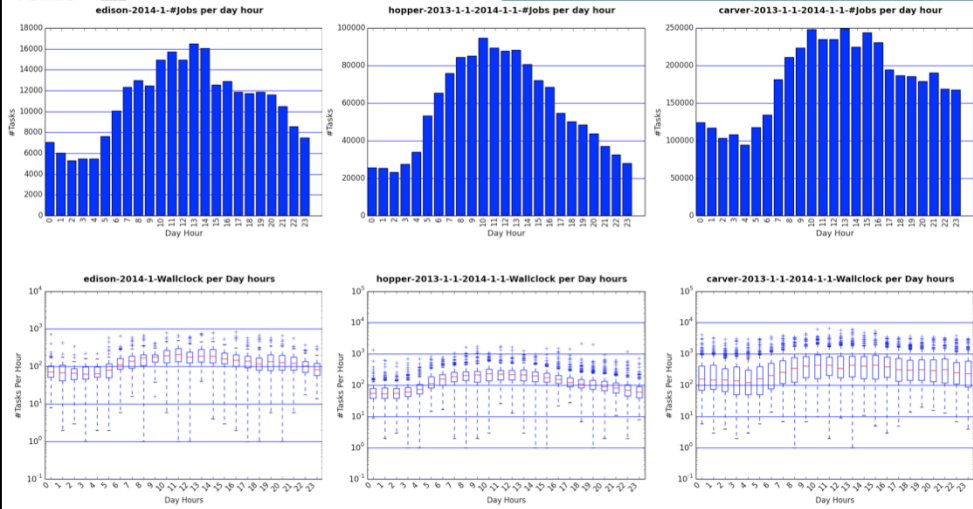
This tables show that in some queues (debug, interactive, etc.) the users are using by default the queue limit for the requested wall clock. And also that in many queues the accuracy is pretty low.



The following slides we can observe time patters on the job submission behavior: Are jobs submitted more on a particular day of the week? Month? Hour of the day? Overall in the year?



Edison, Hopper & Carver:
Submitted Tasks, Day Cycle

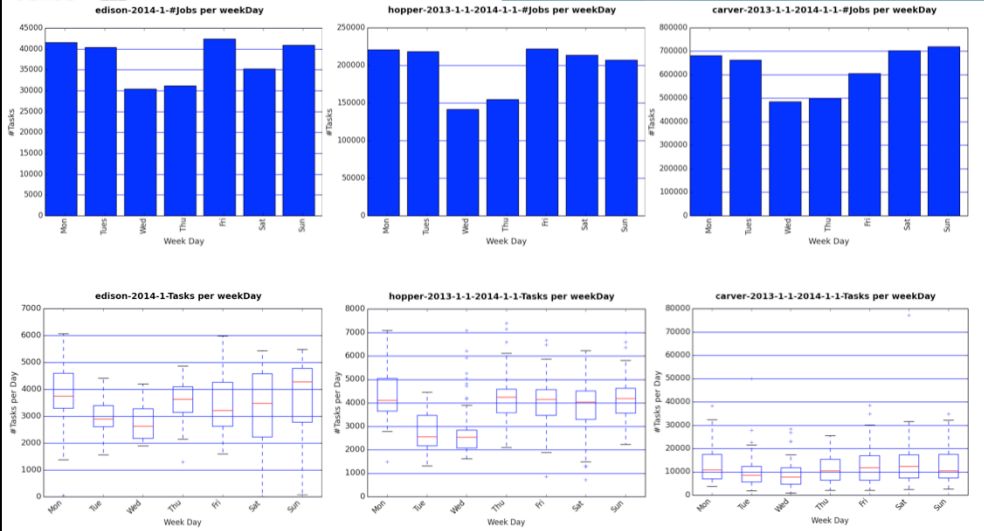


G. P. Rodrigo 2014 - gprodrigoalvarez@lbl.gov

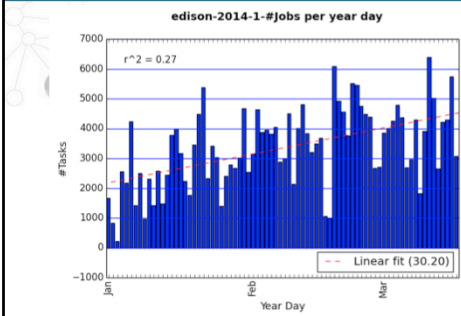
Advanced Computing for Science



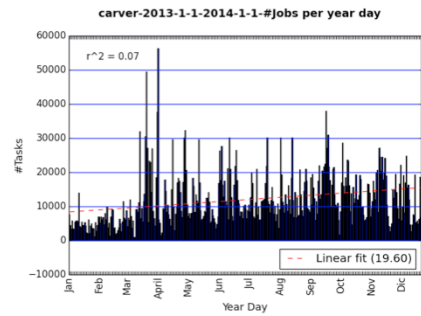
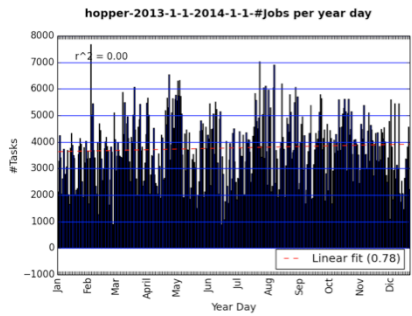
Edison, Hopper & Carver:
Submitted Tasks, Week Cycle

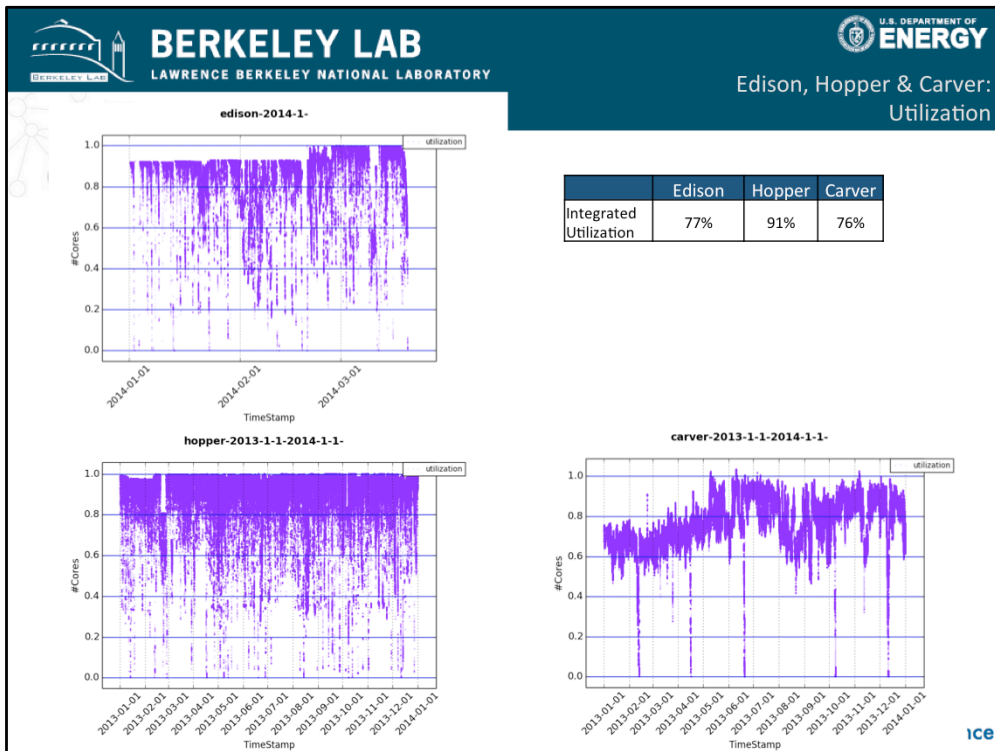


G. P. Rodrigo 2014 - gprodrigovalvarez@lbl.gov



- Increasing trend on #Jobs





Utilization at a moment of time: computed as the number of cores allocated at certain moment of time vs. the total cores in the system. Not taking into account if the cores are used or not or they were allocated because the system doesn't allow node partitioning.

This is a utilization study of the systems:

- The graph represents all changes in utilization along the studied period of time.
- The table has the integrated usage (Surface under the line represented in the graphs) along the studied period.

Important:

- Outages, resource changes were not taken into account.
- The maximum number of cores is taken as the value registered in NERSC site by May 2014. It has not been adapted to the changes in the system.



Task distribution

- Edison has a similar job lengths distribution than Hopper, although Edison present few jobs over 150,000s.
- Edison and Hopper have a similar #Cores distribution, although, again Hopper has more jobs using more cores.
- Carver tasks users less cores (far less) and are shorter.

General workload

- The number of task groups obtained on edison(8), hopper (12) and carver(7) indicates that Hopper has a more heterogeneous workloads.
- In all cases queues are mapped over multiple groups: Not as strong relationship between each queue and the characteristics of the contained tasks.
- Wall clock accuracy: Low. No prediction effort by users (Which is natural)



Walking forward

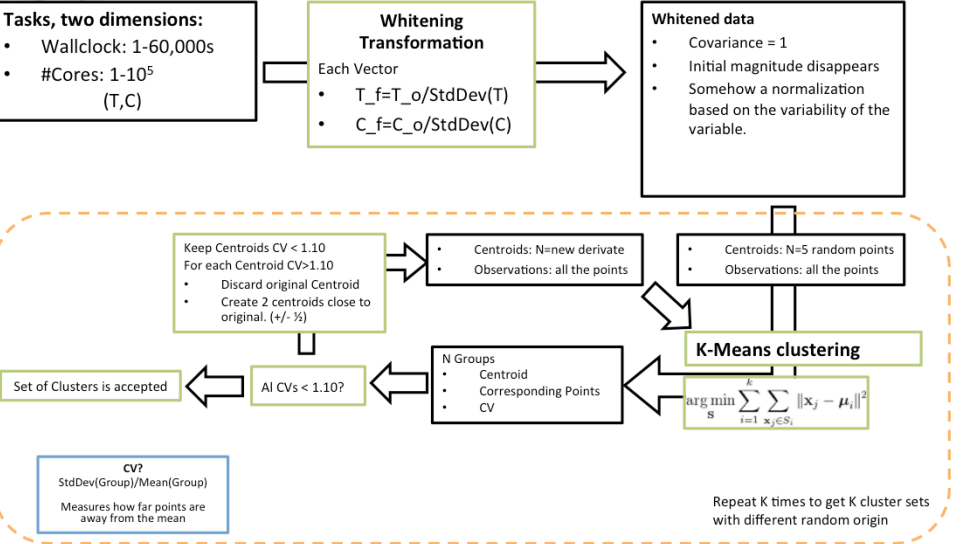
- Setting up a simulation system of the batch scheduler
- Create synthetic workloads.
- Test different scenarios:
 - Different queue definition
 - Wall clock correction
 - Impact on utilization and wait time
- Think about scheduling models



Questions?



On the clustering method



G. P. Rodrigo 2014 - gprodrigo@lbl.gov