Time-Critical Scheduling on a Well Utilised HPC System at ECMWF using Loadleveler with Resource Reservation

Graham Holt ECMWF Shinfield Park Shinfield Road Reading Berkshire UK RG2 9AX

ECMWF (European Centre for Medium-Range Weather Forecasts) was founded in 1973 and is funded by 25 European Countries (18 original participating Member States and 7 cooperating Member States) where Medium-Range Weather Forecasts concentrate on the period 4 – 10 days ahead. The European Weather Centre, as it is more commonly known, is located 35 miles west of London, England on the outskirts of a town called Reading. See http://www.ecmwf.int/about/overview/ for more information.

email: <u>G.Holt@ecmwf.int</u>

Abstract

This article is written in the context of running a suite of time-critical operational numerical weather prediction batch jobs along with a substantial number of research batch jobs on a large IBM Cluster 1600 system. The batch subsystem used is IBM's LoadLeveler incorporating a little known feature called Resource Reservation.

The article describes how the mixture of operational and research parallel batch jobs are scheduled to run on the 117 nodes provided and how Resource Reservation for operational jobs is performed without reference to job class. Where research parallel batch jobs are jobs requesting more than 1 CPU and must run consistently to ensure resources are released predictably. Note - information is given explaining how consistent runtimes are achieved.

1. Background information.

Before 2001, ECMWF had no experience of Loadleveler, having previously used systems from CDC, CRAY (NQE) and Fujitsu (NQS). So ECMWF's experience of Loadleveler is limited to the needs of the system described, and the scheduling strategy devised in early 2002 was kept simple to minimise the learning curve/time. More recently additional IBM Server systems again using Loadleveler have been installed, but experience has shown there is little common ground between the philosophy of scheduling batch and interactive work on Servers and the philosophy of scheduling high-performance parallel batch jobs on a Cluster. So my Loadleveler/scheduling experience is therefore fine-tuned in a blinkered way to the needs of well balanced, highly parallelised batch jobs on a large Cluster to ensure good performance and consistent runtimes. Importantly the over-subscription of processes per node is not allowed, shared memory use is only permitted by up to 4 jobs per node, maximum 8 processes (as long as the total real memory requested by all the jobs does not exceed the total real memory available), and memory paging when detected is reported as very undesirable, even more so if job performance (CPU utilisation) appears to be compromised.

In 2002 the Loadleveler feature Resource Reservation was not known to ECMWF. At that time when operational jobs were about to be run nodes were 'reserved' by draining 'user' batch job classes so that no new 'user' jobs could start. But this manual, time consuming and often complicated method was frequently found to be wasteful, with nodes being left idle unnecessarily. Then in 2003 a decision was taken to introduce a tighter operational schedule and as a result we needed to develop an automated resource based scheduling scheme that would also overcome the known weaknesses in the manual system in use.

It was clear even in 2002 that predictable runtimes were essential for backfill to maximise node utilisation and for predictable node release. So it was agreed the scheduling scheme like backfill should be knowledge based and use

predicted wall_clock_limits derived from historical run-time data. A lot of work had already gone into creating tools and displays that enabled runtime data to be captured, visualised, and made available to enhance job selection and empower backfill. Importantly, having no previous understanding of IBM backfill, tools had already been created to monitor the results of backfill so that the way it worked could be studied and understood. So plans were made to enhance these displays for use during operational periods. But most importantly a dynamic reservation based scheduling scheme was sought, a scheme independent of physical nodes and physical classes. For this IBM suggested using Resource Reservation.

2. Overview.

ECMWF runs many types of operational forecasts on a daily, weekly and monthly basis, and hundreds of thousands of products are sent (disseminated) to the Member States each day. Twice a day for about 90 minutes all 117 nodes (936 CPUs) are reserved for and used by operational batch jobs (see Figure 1 below)



Figure 1 - the average number of CPU's allocated to parallel operational jobs, plotted over 12 minute intervals, where for parallel jobs 936 CPUs (the red line) is the maximum possible.

Please note – when all 936 CPUs are reserved for operational jobs, some CPUs become unallocated for a few seconds as jobs complete and new jobs start, and at times due to the job-mix at least 1 node (8 CPUs) may not be used for some minutes. So the average maximum use of around 920 CPUs out of 936 is seen as a very good achievement.

However when operational batch jobs are not being run or do not require all of the 'parallel' nodes, every attempt is made to fully utilise 'resources' by running research user's batch jobs

3. A mixture of parallel operational and research batch jobs.

Please note there is little difference between the computational needs and performance characteristics of research batch jobs and their operational equivalent jobs. The codes used are almost identical. The big difference is operational suites of jobs are run usually twice or four times a day and each time use as input the most recent world-wide observations acquired in the preceding few hours. But research experiments differ in that they use historical data (not real-time data acquisition), have no need to create end-user products and over a period of many days submit 28 data assimilations and forecasts, using alternately 00z and 12z data over a 14 date period. Normally there are from 10 to 15 research experiments running simultaneously and these plus other smaller jobs easily utilise the 117 nodes available (see Figure 2 below) whenever resources are not being used operationally. Please note – 00z data is collected globally and simultaneously at 00z GMT, likewise 12z data at 12z GMT.



CPU's allocated on HPCB by NON operational parallel jobs Twe 23 March 2004

Figure 2 - the average number of CPUs allocated to non-operational jobs, before Resource Reservation was introduced, plotted over 12-minute intervals, where 936 CPUs (the red line) is the maximum possible.

Please note in Figure 2 above at 02:48, a 12-minute period when all 936 CPUs are allocated.

4. Scheduling requirements.

One additional function of pre-operational testing is to detect research jobs that do not perform well or do not exhibit consistent runtimes in keeping with the existing operational schedule. Quite simply, jobs (new code) from research experiments cannot progress beyond the research stage to become operational if they do not perform well enough. So a lot of effort has been put into ensuring all parallel jobs have consistent runtimes, and as a result each job confirms by running in the given time that there are no I/O bottlenecks, the GPFS filesystems used are performing consistently, the high-performance internal switch network is performing consistently, that the jobs have not been slowed by paging to local disk and lastly that the jobs (the new code) perform as expected too.

As a result, in a research experiment of jobs run 28 times, at least 95% of the forecast jobs are not expected to vary by more than $\pm 1\%$, with the preceding sets of 28 data assimilation jobs varying by no more than $\pm 3\%$. The big bonus being, once 3 or 4 sets of jobs for an experiment have completed, the subsequent 24 sets of jobs become health checks (each a diagnostic) for good system performance. It is true when a job has run less than twice it has an unknown runtime (shown on displays as a wall_clock_limit of *1-day) but other jobs with similar job-names (for example ifstraj_uptraj_0) are likely to have a known run-time, giving an indication when new jobs might complete.

Importantly the golden rule before submitting an experiment or individual parallel job is – if the data needed by a parallel job is not on a GPFS filesystem, a serial job must first be run to obtain the initial data, and the data obtained must be written to a high-performance GPFS filesystem. In this way all parallel jobs in the Cluster can be scheduled independent of all other systems.

If at any time a parallel job runs for longer than expected or the job is flagged as idle, the job is displayed in red on an operator display and immediately investigation begins. What has to be determined is - is the problem a function of the job, the environment or the system?

5. The mission - to keep the system fully utilised yet run operational work to a tight schedule.

5.1 Fully utilised

Keeping the system lightly loaded to make it easy to run operational work on time is not our style (is for wimps). Users in the Research Department have always been able to expand their experimentation to fully utilise the system, and giving them the service they want by fully utilising the system is very rewarding.

5.2 Keeping to operational schedules.

The plans made in 2003 to run the operational suites to a more demanding schedule made the introduction of an automatic and class-independent resource reservation system essential. The previous 'on demand' scheduling scheme could not guarantee consistent start-times, the variability often exceeded the 15 minute requirement, and as a result there was often insufficient recovery time should jobs fail and need to be rerun.

6. The main objective.

The main objective of the reservation system was then defined like this: Resource reservation is needed to ensure that the day-to-day variation in the end-times of operational forecasts and the generation of products does not exceed 15 minutes compared with the predicted optimum time, whilst keeping node occupancy close to 96% (which is what had been achieved beforehand).

Fortunately the basic building blocks were all in place based on the predicted run-time for most jobs, the standard design of most experiments and the repetitive consistently running job mix, giving the ability to accurately assess node release and node availability times.

6.1 The complexity of operational suites should not be underestimated.

Until now I have trivialised operational suites at ECMWF by talking only about the more substantial time-critical computational jobs. The reality is each suite contains 1000's of jobs that run on multiple systems, have complex interdependencies, multiple-event triggers, time triggers and late flags. Amongst other things, operational jobs acquire observations (data), analyse and check the validity of this data, compute the relevant initial datasets used by the forecast, execute the forecasts, save data at many stages throughout the process, create end-user products, verify the results, archive the results, plot data, send products to users with many operations taking place in parallel. One concern with such a complex set-up, was tighter schedules would lead to new bottlenecks and a complete loss of flexibility, which would reduce efficiency and system utilisation. So the brief was altered subtly to request an automatic system that used the 15 minutes of flexibility if this increased system utilisation.

7. Resource Reservation.

7.1 Defining the need for resources and benefits of Resource Reservation.

The CPU allocation averages shown previously in Figure 1 indicate that most of the time none of the Cluster resources are used by operational jobs, and show during the main periods (08:20 to 11:40 and 19:45 to 23:35) the need for the resources provided to increase in 3 stages from 0% to 30%, from 30% to 60%, and finally from 60% to100% for about 90 minutes. But what Figure 1 does not show are the 2 small periods when CPU use drops for just a few minutes from 30% to 0% before rising to 60%, and later from 60% down to 30% before rising up to 100%. If resources are not reserved or classes drained, user jobs flood the system.

Past experience showed it is possible, but by no means easy, to drain classes on some nodes to limit the use of resources by non-operational jobs to 40%, whilst the operational use dips briefly from 60% to 30% before rising to 100%. By comparison dynamic reservation independent of class using Resource Reservation provides a much needed simplicity. One might even say Resource Reservation brings elegance to the solution.

And by the same token with one operation, as soon as all operational jobs to be run are submitted, and because operational jobs are selected first, by virtue of highest SYSPRIO, it is possible to cancel (reset) reservation, make all non-operational jobs candidates for selection before the final operational jobs have started and allow backfill to take place.

7.2 Describing Resource Reservation.

Resource Reservation for the timely running of operational jobs combined with optimum system utilisation has as a core function the ability to request in advance a reduction to the resources available to non-operational jobs, irrespective of the class the jobs are in, whilst taking into account

- . the requirements of operational jobs (expected start-time, CPUs needed x elapsed time),
- the release of resources (expected elapsed time remaining for research jobs in execution)
- . the estimated start-times (or delays if research jobs appear to end soon enough)

Additionally the scheme should take into account

. the flexibility of the operational schedule

- . how close to or far behind the optimum schedule today's operational runs are
- how consistent the historical runtime data is for the jobs that are running,
- that it is possible to force some jobs to write checkpoint files then kill them without loss.

Finally the scheme should release resources for use by non-operational jobs whenever possible to optimise the use of all resources as shown by Figure 3 below.



Figure 3 - the allocation of CPU's to all jobs before Resource Reservation was introduced, plotted over 12minute intervals, where the red line (936 CPUs) is the maximum number that parallel jobs can use.

8. Summary so far.

The key to everything is predictability. ECMWF's business relies on the timely production of weather forecast products (our weather predictions) so not surprisingly ECMWF has a strong desire for all jobs to run predictably. Otherwise how do you keep to a schedule. And once the causes of variable job runtimes are eliminated (as they have been) and jobs run with little run-time variation, the data available enables

- . backfill to work most effectively.
- . the runtime and end-time of user batch jobs to be monitored.
 - most batch jobs to act as diagnostic checks on system performance as well as on job performance.
- resources to be reserved sufficiently in advance to ensure operational jobs start optimally.

9. Defining the core elements required for effective Resource Reservation.

Analysis of the associated issues have identified 4 main elements, which are listed below so that they may be checked and ticked off. In doing so one can be sure all problems have been identified and solutions put in place. It is true there is some duplication of what has gone before but this is necessary. The 4 elements being

A. The need to know in advance what resources are required by operational jobs and the time at which these jobs should start.

B. The need to ensure that non-operational jobs in execution can be guaranteed to complete and release the resources in a predictable way (there is little point doing this if, in order to achieve it, many jobs have to be cancelled prematurely).

C. The need to have processes that take the information about operational jobs and running jobs and use this along with other information to dynamically reserve resources, the purpose being to restrict or stop non-operational jobs from starting, then release resources.

D. The need to be able to visualise what is happening and verify that it is working as designed.

10. Expanding on the core elements for effective Resource Reservation.

Taking the heading from 9. (and covering A. and B. very quickly) we have:-

A. The need to know in advance what resources are required by operational jobs and the time at which these jobs should start.

The requirements of operational jobs, in terms of resources, start time, elapsed time and dependencies on other factors are well known. Nothing more needs to be added.

B. The need to ensure that non-operational jobs in execution can be guaranteed to complete and release the resources in a predictable way.

As mentioned before, predictability is paramount. There is no point reserving resources if job end-times are unreliable. Good utilisation is compatible with tight schedules only if you have a predictable set of non-operational jobs. The key elements of ECMWF's success are the independence of parallel jobs, consistent job performance and consistent system performance. And to illustrate the point Figure 4, which follows, shows historical run-time data for an assimilation job and the predicted Min and Max runtimes. Where the predicted 95% Max value is used for scheduling and backfill purposes. The job selected does not quite have the quoted runtime range of $\pm 3\%$ and this is commented on later.

F	-¤ hpcb			• • ×
	Job: rdx-ehmv_ifsmin_uptraj_1-/hpcb/rdx_di	ir/log/stz/ehmv/an/main/xx	:/an/4dvar/uptraj_1/ifsmin	
	Run times Date	Notes		
	00:51:52 Tue Mar 23 07:33:23 2004 00:52:29 Mon Mar 22 20:59:16 2004 00:50:35 Mon Mar 22 13:40:56 2004 00:44:00 Sat Mar 20 17:40:11 2004 00:44:50 Sat Mar 20 17:40:11 2004 00:44:58 Sat Mar 20 03:03:38 2004 00:44:58 Sat Mar 20 03:03:38 2004 00:44:58 Fri Mar 19 06:31:05 2004 00:48:54 Fri Mar 19 06:31:05 2004 00:50:50 Fri Mar 19 06:31:05 2004 00:48:57 Thu Mar 18 09:03:41 2004 00:48:55 Thu Mar 18 02:56:31 2004 00:50:27 Wed Mar 17 17:17:12 2004 00:51:32 Wed Mar 17 02:17:54 2004 00:51:15 Wed Mar 17 02:17:54 2004	hpcb2301.1469889.0 Nodes hpcb2401.1469889.0 Nodes hpcb2401.1468401.0 Nodes hpcb2301.1466756.0 Nodes hpcb2401.1459329.0 Nodes hpcb2401.1458507.0 Nodes hpcb2401.1452953.0 Nodes hpcb2401.1451611.0 Nodes hpcb2301.1451611.0 Nodes hpcb2301.1447545.0 Nodes hpcb2401.1446125.0 Nodes hpcb2401.1446125.0 Nodes hpcb2401.1446125.0 Nodes hpcb2401.1444640.0 Nodes hpcb2401.1444640.0 Nodes	16 16 16 16 16 16 16 16 16 16 16 16 16 1	
	00:47:41 Tue Mar 16 07:21:50 2004 00:49:20 Mon Mar 15 21:24:19 2004	hpcb2301.1439114.0 Nodes hpcb2301.1437568.0 Nodes	16 16	
	Discarding extreme values 00:52:29 and 00: Runs Min Avg Max 17 00:44:58 00:49:27 00:51:	44:00 52		
	ESTIM.70% 00:47:43 00:51: ESTIM.95% 00:46:00 00:52: Std Dev : 00:01:43 Spread : 3% Standard Deviation to Average hpcb2501{/home/cos/newops/bin}:69\$	10 53 : Ratio		:

Figure 4 - historical run-time data and predicted run-time estimate (95%/Max) for job ehmv_ifsmin_uptraj_1

Figure 4 shows 17 runtimes. Discarding the shortest and longest, the average runtime (under Avg) is shown to be 00:49:27. The expected $\pm 3\%$ would give Min 00:47:59 and Max 00:50:55. Sadly 6 runtimes fall outside this range. The larger than expected spread of 00:44:58 to 00:51:52 means the predicted 95% Max runtime of 00:52:53 exceeds by 24 seconds the known longest run of 00:52:29, and exceeds the average by almost 3.5 minutes. But this is no disaster. It means successive jobs will end about 3.5 minutes before the predicted run-time and as a result 16 nodes out of 117 may be idle for 3.5 minutes longer than expected before operational jobs run. Less-than ideal but never the less still very acceptable.

What is clear however is the basic requirements exist for the introduction of a resource reservation system. We know in advance the resources needed by operational jobs and the time these jobs should start. We have confidence that the non-operational jobs have well predicted run-times and will finish as required/expected.

C. The need to have processes that take the information about operational jobs and running jobs and use this along with other information to dynamically reserve resources, the purpose being to restrict or stop non-operational jobs from starting, then release resources.

To reserve resources IBM suggested ECMWF use a little-used Loadleveler feature called "floating resources", where "floating resources" are in effect cluster-wide licenses to use CPUs. The floating resources we use are defined as "FloatingCpus", the unit of which is a physical CPU. These are requested by non-operational jobs in the same way as other resources, such as ConsumableCpus and ConsumableMemory via the "resources" LoadLeveler directive. e.g.

@ resources = ConsumableCpus(4) ConsumableMemory(3600Mb) FloatingCpus(4)

And as it is imperative that all non-operational jobs contain such a directive and request the correct number of FloatingCpus all jobs are passed through a "job filter" and the job filter inserts the request. Then by reducing the number available, it is possible to restrict all non-operational jobs to a limited number of CPUs, and ensure that only operational jobs, which by design do not request floating resources, can use the remaining CPUs.

With 117 nodes, each node an 8-way SMP system, making 936 CPUs in total, the following directive in LoadLeveler's configuration file:

FLOATING RESOURCES = FloatingCpus(936)

means that outside operational periods we make all 936 CPUs available for running non-operational work.

Then when it becomes necessary to reduce the non-operational workload, to reserve CPU's for operational jobs, the value of cluster-wide FloatingCpus limit is reduced in LoadLeveler's configuration file, and by signalling the relevant LoadLeveler daemons the change is introduced. As non-operational jobs finish and release nodes, the number of FloatingCpus in use is reduced but no new non-operational jobs will start until the number of FloatingCpus in use becomes less than the value of FloatingCpus set in the LoadLeveler configuration file. Of course, as this is a reservation scheme, the value is reduced some time before the operational jobs are submitted, taking into account the time needed for user jobs to end (which will be covered later).

Ignoring the time in advance calculation for now, to reserve 256 CPUs for the first operational job, the value of FloatingCpus in the LoadLeveler configuration file is reduced from:

FLOATING RESOURCES = FloatingCpus(936)

to

FLOATING RESOURCES = FloatingCpus(680)

and once 256 CPUs become available, they will remain unused until the operational job starts.

However, the skill is in working out when to reduce the value of FloatingCpus. Too early and resources will lie idle, waiting for the operational jobs to be submitted. Too late and the resources will not be released in time. However as the operational schedule has a degree of flexibility a little late is better than a little early.

Clearly much depends on the number of nodes needed (to give time for a running job or running jobs to end) and taking into account the known flexibility in the running of operational jobs. To maximise the use of the system it would be beneficial if shorter non-operational jobs with the most consistent historical run-times were to be selected just before operational periods. Additionally if the operational resources required are needed in stages rather than all at once, a more flexible approach can be made.

ECMWF talked to IBM about this. IBM is proposing to implement a resource reservation scheme that, like LoadLeveler's backfill scheduler, works by using a job's wall_clock_limit. Sadly on a basic IBM system wall_clock_limit is rarely set accurately, and if it is the kill on wall_clock_limit exceeded is a rigid scheme that is not acceptable to ECMWF. So a local fix was needed.

11 The backfill problem.

11.1 First identify the problem.

IBM ask users to set the wall_clock_limit and then use the value given for backfill. Yet users are known to have a poor understanding of the length of time their jobs will take to run and even on the 'best' systems runtimes will vary. So expecting users to keep tabs on the spread in unrealistic. Then add to this the basic philosophy behind wall_clock_limit, which is to kill jobs which over-run, and it is no wonder that users never specify an "accurate" value. You do not have to be a rocket scientist to realise I have no faith in scheduling and backfill using user-specified wall_clock_limits. The system forces users to be inaccurate.

11.2 The ECMWF solution.

- Ask users to not specify a job wall_clock_limit.
- When each job completes, to collect data (including actual run-time) and store it in a database.
- To predict wall_clock_limits using the data collected.

If a job (same name, same user) is submitted more than twice, the job filter uses the historical run-time data to create a predicted run-time, see Figure 4 using the Estim.95%/Max value 00:52:53 for this purpose. Then add one day to this and set the wall_clock_limit to "1+00:52:53". The time added ensures the job is not killed should it over-run its' expected runtime and the 1+ is used because it is very easy to code. Additionally it is very easy to mask out in displays leaving the true predicted run-time for all to see. Please note - the backfill scheduler works by comparing time differences. So provided all wall_clock_limits use the 24-hour baseline, which they do, the backfill scheduler functions as designed. The 1+ is only an offset used to stop backfill from killing jobs, it can otherwise be ignored.

As pointed out earlier, job runtimes are quite consistent, and the predicted run-times are always a slight overestimate using the "Estim.95%/ Max" value, so it is quite rare for jobs to over-run. It is much more common for nonoperational jobs to end about 1 to 2 minutes ahead of the predicted run-time. But should a job run for more than 1 minute longer than the predicted run-time it will be highlighted in red on the operator display, to ensure investigation starts as soon as possible. An example of this comes later with Figure 7.

Up to now I have described all of the elements that enable a system to predict and control the reservation, use and release of CPUs (nodes). I have shown that the data exists to enable nodes to be filled and to ensure nodes are released in line with tight operational schedules. Theory is fine, but how is it done in practise?

12. When does the value of FloatingCpus get modified in the LoadLeveler configuration file?

Analysis of the workload is used to determine the average time for nodes to be released. And from this we have determined the time in advance (in minutes) for each reduction of FloatingCpus. The values chosen can be seen under the heading 'time' in Figure 6. A more sophisticated scheme could be used but this simple scheme works and has not yet been improved on.

13. How does the value of FloatingCpus get modified in the LoadLeveler configuration file?

A utility was written to modify the LoadLeveler configuration file "/loadl/config". This utility "floating_cpus" takes care of file locking to ensure that the file is not being manipulated by other utilities concurrently. At present "floating_cpus" is executed via "cron" at the times shown in Figure 6. The call to run "floating_cpus" comes with 2 parameters, the number of **CPUs** to be reserved and **time** in advance the new value is being applied.

Interim State Sta		• • ×
File Sessions Settings Help		
crontab -1	CPUs time	4
06 8 * * * /loadl/floating_cpus 29 9 * * /loadl/floating_cpus 50 9 * * /loadl/floating_cpus 00 11 * * * /loadl/floating_cpus 21 19 * * * /loadl/floating_cpus 44 20 * * * /loadl/floating_cpus 15 22 * * /loadl/floating_cpus 15 22 * * /loadl/floating_cpus	 256 11 >> /loadl/crontab.log 2>&1 544 16 >> /loadl/crontab.log 2>&1 928 19 >> /loadl/crontab.log 2>&1 0 0 >> /loadl/crontab.log 2>&1 256 11 >> /loadl/crontab.log 2>&1 544 16 >> /loadl/crontab.log 2>&1 928 19 >> /loadl/crontab.log 2>&1 0 0 >> /loadl/crontab.log 2>&1 	
New Konsole		

Figure 6 - the crontab runtimes where the first parameter is CPUs reserved and the second value is time (minutes), that is minutes in advance of the operational need.

Taking the first cron entry from Figure 6

CPUs time

06 8 * * * /loadl/floating_cpus 256 11

At 08:06 the crontab job /loadl/floating_cpus is run to reserve 256 **CPUs**. This is done by altering the value of FloatingCpus in the /loadl/config file, reducing the value from 936 to 680 with the knowledge that the 256 CPUs requested will be needed. The value of **time** (11 minutes) means the first operational job is expected to be submitted at 08:17.

In the future, the script 'floating_cpus' will be executed by an event that is an integral part of the operational suite of jobs. Then if operational schedules are changed, the reservation times will automatically change too. However the "cron" mechanism will be retained as a safety net. If the start of operational activity is delayed significantly and the event-linked reservation of nodes does not run, a "cron" run a little later will ensure that resources are reserved so that when the operational activity eventually starts it is not be delayed further. Although potentially wasteful, on the few occasions when forecasts run very late it is essential that the resources needed are already available.

14. When does the value of FloatingCpus get modified in the /loadl/config file?

The utility "floating_cpus" that sets the limit of FloatingCpus in the configuration file, runs ahead of the time reserving resources based on our experience of the normal average release of nodes. We plan to change this "rule of thumb" mechanism to one that analyses running jobs, and pre-selects the 'right' jobs. But currently this is not done.

It is obvious that schemes to alter job selection before operational periods will influence the use of nodes, the release of nodes and the optimum time in advance that resource reservation should be made. The opportunity for complexity is large. However we keep this simple by using average node release patterns and checking to see what actually happens. There is some waste, but to start with if there is less than a 15-minute variation to the runtimes of the operational suites we have achieved our primary aim. Slightly better node utilisation could be obtained but not a lot. Integrating Resource Reservation into the operational suite comes first.

15. Summarising A. B. and C.

The components of Resource Reservation are

- CPU resources called FloatingCpus defined in the /loadl/config file
- the job-filter ensures all non-operational jobs request FloatingCpus
- known requirements for all operational jobs
- historical run-time data for most user jobs
- A predicted run-time for most user jobs
- Confidence factors (see Figures 4 and later Figure 7 Std. Dev and Spread) in the predicted run-times.
- User job wall_clock_limits set to 1+ predicted run-times.
- Efficient backfill but with kill on wall_clock_limit disabled.
- A utility called floating_cpus that sets new values of FloatingCpus in the config file
- No need to drain or resume job class globally or node by node.
- A utility that gives information about the average frequency of released nodes
- cron jobs that ensure the value of FloatingCpus is set sufficiently in advance

16. Visualisation and human intervention.

So finally we get to **D**. The need to be able to visualise what is happening and verify that it is working as designed.

Visualising job selection and backfill activity is done with a single display called ll_jobs. Please note - ll_jobs is a display program and does no data gathering. A perl program called nll_sched provides the information displayed. nll_sched runs every 10 seconds and uses the Loadleveler API to get the data needed then processes the data.

ll_jobs has 2 modes - non-operational and operational, where operational is triggered by FloatingCpus <936. By watching the ll_jobs display over time, it is possible to be confident that what is supposed to happen actually happens. Importantly when resources have been reserved, but the display shows CPUs are not going to be released because the active jobs will run for too long, detailed information is displayed from which action can be taken. And when action needs to be taken, it is shift staff taking the action. They make sure the nodes get released before a delay of more than 15 minutes occurs.

So next a display of ll_jobs in non-operational mode and other related information.

es Avail gy/np 3/3 Drained 0/0 Draining 0/0 Tue Mar 23 14:91:99 2004 Modes who is a second of the interval o]	-₩ hj	ocb												
Ding Models W.C. (Ock. PP10 P111 ehba_ifstraj_uptraj_0 16 hpb2301.1471557.0 np 00:16 13700-50 -00:00 smdel_an 1 hpb2301.1471557.0 np 00:16 13700-50 -00:00 smdel_an 1 hpb2301.1471557.0 np 00:16 13700-50 -00:00 smdel_an 1 hpb2301.1471530.0 np 00:10 36600-50 00:00 smdel_an 1 hpb2301.1471573.0 np 00:10 36600-50 00:00 smdel_an 1 hpb2301.1471573.0 np 00:10 36600-50 00:00 smdel_an 1 hpb2301.1471573.0 np 00:00 13700-70 00:00 ehdy_indel_fc 4 hpb2301.1471593.0 np 00:00 13700-70 00:00 ehdy_indel_fc 4 hpb2301.14711984.0 np 00:00 13700-70 00:00 ehdy_indel_fc 4 hpb2301.14711984.0 np 00:01 13700-70 00:00 ehdy_indel_fc 4 hpb2301.14711984.0 np 00:00 <td< td=""><td></td><td>Node</td><td>s Avail</td><td>op/np</td><td>3/3</td><td>Dra</td><td>ined</td><td>0/0</td><td>Dry</td><td>aining</td><td>; O/C</td><td>) 1</td><td>ue Mar</td><td>23 14:49:</td><td>59 2004</td></td<>		Node	s Avail	op/np	3/3	Dra	ined	0/0	Dry	aining	; O/C) 1	ue Mar	23 14:49:	59 2004
c hba_ifstraj_uptraj_0 16 hpcb201.1471693.0 op 00:13 3600-50 00:03 s model_an 1 hpcb201.1471694.0 op 00:04 38600-50 00:03 s model_an 1 hpcb201.1472001.0 op 00:04 38600-50 00:03 s model_an 1 hpcb201.1472001.0 op 00:04 38600-50 00:03 s model_an 1 hpcb201.1472002.0 op 00:04 38600-50 00:03 s model_an 1 hpcb201.1471698.0 np 00:014 38600-50 00:02 s model_an 1 hpcb201.1471980.0 np 00:11 13700-70 00:06 eht2_ifstraj_uptraj_0 16 hpcb201.1471980.0 np 00:06 13700-70 00:07 eht4_ifstraj_uptraj_0 16 hpcb201.1471980.0 np 00:00 13700-70 00:02 eht4_ifstraj_uptraj_0 16 hpcb201.1471980.0 np 00:00 13700-70 00:12 eth4_ifstraj_uptraj_0 16 hpcb201.1471980.0 np 00:00 13700-70 00:12 eth4_ifstraj_uptraj_0 <td></td> <td>кunn rdx</td> <td>ehnO_</td> <td>ifstra</td> <td>ri_4dvai</td> <td>r</td> <td>NOQ 16</td> <td>es hpc</td> <td>b230</td> <td>1.1471</td> <td>1557.0</td> <td>) np</td> <td>00:15</td> <td>19700-7:</td> <td>F1615</td>		кunn rdx	ehnO_	ifstra	ri_4dvai	r	NOQ 16	es hpc	b230	1.1471	1557.0) np	00:15	19700-7:	F1615
s model_ant i hplos2301.47/1534.0 gP 00103 32200-50 010 0 s model_ant i hplos2301.47/1534.0 gP 00104 32200-50 010 0 s model_ant i hplos2301.47/2001.0 gP 00104 32200-50 010 0 s model_ant i hplos2301.47/2001.0 gP 00104 32500-50 010 0 s model_ant i hplos2301.47/2001.0 gP 00104 32500-50 010 0 s model_ant i hplos2301.47/1980.0 np 00114 13700-70 00106 ehbu ifstrai_uptraj_0 i 6 hplos2301.47/1980.0 np 00116 13700-70 00100 ehbu ifstrai_uptraj_1 i 6 hplos2301.47/1980.0 np 00116 13700-50 00107 ehbu ifstrai_uptraj_1 i 6 hplos2301.47/1902.0 np 00116 13700-50 00107 ehbu.ifstrai_uptraj_1 i 6 hplos2301.47/1902.0 np 00154 13700-50 00107 ehbu.ifstrai_uptraj_1 i 6 hplos2301.47/1902.0 np 001054 13700-72 00170 ehbu.ifstrai_uptraj_0 i 6 hplos2401.47/1902.0 np 00100 13700-72 00170 ehbu.ifstrai_uptraj_0 i 6 hplos2401.47/1902.0 np 00104 13700-50 02126 ehbu.ifstrai_uptraj_0 i 6 hplos2401.47/1903.0 np 00104 13700-50 02126 ehbu.ifstrai_uptraj_0 i 6 hplos2401.47/1903.0 np 00114 13700-50 02126 ehbu.ifstrai_uptraj_0 i 6 hplos2401.47/1903.0 np 00104 13700-50 02126 ehbu.model_fc i 6 hplos2401.47/1930.0 np 00104 13700-50 02126 ehbu.model_fc i 6 hplos2401.47/1930.0 np 00104 13700-50 02126 ehbu.model_fc i 16 hplos2401.47/1930.0 np 00104 13700-50 02126 ehbu.model_fc i 16 hplos2401.47/1930.0 np 00104 13700-50 02126 ehbu.model_fc i 16 hplos2401.47/1930.0 np 00171 13700-50 00126 ehbu.model_fc i 16 hplos2401.47/1950.0 np 00161 13700-50 00126 ehbu.model_fc i 16 hplos2401.47/1950.0 np 00161 13700-50 00126 ehbu.model_fc i 16 hpl		rdx	ehba_	ifstra	j_uptr	aj_O	16	hpc	b240	1.1471	269.0	np	00:16	19700-5	0 -00:0
s model_an 1 hptb2401.1472001.0 pp 00:04 38600-50 01:02 s model_an 1 hptb2401.1472002.0 pp 00:04 38600-50 01:02 s model_an 1 hptb2401.1472003.0 pp 00:04 38600-50 01:02 ehftC_1fsmin_uptraj_0 8 hptb2401.1471898.0 np 00:14 18700-71 01:02 ehty_indel_rC 4 hptb2301.1471499.0 np 00:18 19700-70 01:02 ehty_indel_rC 4 hptb2401.1471901.0 np 00:16 19700-50 01:71 ehty_indel_rC 4 hptb2401.1471994.0 np 00:16 19700-72 01:10 ehty_indel_rC 4 hptb2401.1471994.0 np 00:16 19700-72 01:10 ehty_indel_rC 1 16 hptb2301.1471493.0 np 00:16 19700-72 01:10 ehty_indel_rC 1 16 hptb2301.1471493.0 np 00:16 19700-72 01:16 ehty_indel_rC 1 16 hptb2401.147195.0 np 00:16 19700-72 01:16 ehty_indel_rC 1 16 hptb2401.147195.0 np 00:10 19700-72 01:16 ehty_indel_rC 1 16 hptb2401.147195.0 np 00:10 19700-72 01:16 ehty_indel_rC 1 16 hptb2401.147195.0 np 00:10 19700-70 01:24 ehty_indel_rC 1 16 hptb2401.147195.0 np 00:10 19700-70 01:32 ehty_ifstraj_uptraj_0 12 hptb2401.147195.0 np 00:10 19700-50 01:24 ehty_idstraj_uptraj_0 16 hptb2401.147195.0 np 00:10 19700-50 01:24 ehty_idstraj_uptraj_0 16 hptb2401.147195.0 np 00:10 19700-50 01:25 ehty_idstraj_uptraj_0 16 hptb2401.147195.0 np 00:10 19700-50 01:25 ehty_idstraj_uptraj_0 16 hptb2401.147195.0 np 00:10 19700-50 01:26 ehty_idstraj_uptraj_0 16 hptb2401.147195.0 np 00:10 19700-50 01:26 ehty_idstraj_uptraj_0 16 hptb2401.147195.0 np 00:10 19700-50 01:26 ehtz_idstraj_uptraj_0 16 hptb2401.147150.0 np 00:10 19700-50 01:26 ehtz_idstraj_uptraj_0 16 hptb2401.147155.0 np 00:01 19700-50 01:26 ehtz_idstraj_uptraj_0 16 hptb2401.147155.0 np 00:14 19700-50 01:26 ehtz_idstraj_uptraj_0 16 hptb2401.147157.0 nodes 16 01:15:21 Tue Mar 16 03:42:50 2004 hptb2401.147157.0 Nodes 16 01:15:24 Tue Mar 16 03:42:50 2004 hptb2401.147157.0 Nodes 16 01:15:24 Tue Mar 16 03:42:50 2004 hptb2401.147157.0 Nodes 16 01:15:2		emos	model	_an _an			1	hpc	:0230 :0230	1, 1471	1593. U 1594. C) op) op	00:03	36600-50	0 00:0
s model_an s model s model		emos	model	an			1	hpo	b240	1.1472	2001.0) op	00:04	36600-50	0 00:0
s model an 1 hptp2401.472103.0 of 00:04 36600-60 00:02 chWu 1f5traj_uptraj_0 8 hptp2401.4721983.0 np 00:14 19700-70 00:06 chWu 1f5traj_uptraj_1 16 hptp2301.4721070.0 np 00:22 19700-50 00:06 chJ_lmodel_fc 4 hptp2401.4721902.0 np 00:54 19700-72 00:17 chtp4_1f5traj_uptraj_1 16 hptp2301.4721994.0 np 00:55 19700-72 00:17 chtp4_1f5traj_uptraj_1 16 hptp2301.4721994.0 np 00:56 19700-72 00:17 chtp4_1f5traj_uptraj_0 12 hptp2401.4721994.0 np 00:00 19700-71 00:00 chtp4_1f5traj_uptraj_0 12 hptp2401.4721994.0 np 00:14 19700-70 00:00 chtp4_1f5traj_uptraj_0 12 hptp2401.4721994.0 np 00:14 19700-50 02:24 chtp2_1f5traj_uptraj_0 15 hptp2401.4721994.0 np 00:16 19700-71 00:00 chtp4_1f5traj_uptraj_0 16 hptp2401.4721994.0 np 00:14 19700-50 02:14 chtp2_1f5traj_uptraj_0 16 hptp2801.4721994.0 np 00:14 19700-50 02:14 chtp2_1f5traj_uptraj_0 16 hptp2801.4721994.0 np 00:14 19700-50 02:14 chtp2_1f5traj_uptraj_0 16 hptp2801.4721994.0 np 00:14 19700-50 01:24 chtp2_1f5traj_uptraj_0 16 hptp2801.4721994.0 np 00:14 19700-50 01:24 chtp2_1f5traj_uptraj_0 16 hptp2801.4721994.0 np 00:17 19700-50 01:24 chtp2_1f5traj_uptraj_0 16 hptp2801.4721994.0 np 00:17 19700-50 00:52 chtp2_1f5traj_uptraj_0 16 hptp2801.4721983.0 np 00:10 19700-50 00:22 chtp2_1f5traj_uptraj_0 16 hptp2801.4721983.0 np 00:14 19700-50 00:25 chtp2_1f5traj_uptraj_0 16 hptp2801.4721983.0 np 00:01 19700-50 00:22 chtp2_1f5traj_uptraj_0 16 hptp2801.4721983.0 np 00:01 19700-50 00:20 chtp2501{/home/ops/cog}:58\$ runt hptp2301.1471557.0 chtp2201.4721444.0 Nodes 16 00:15:24 Tue Mar 16 03:42:50 2004 hptp2301.44719157.0 Nodes 16 00:15:24 Tue Mar 16 03:42:50 2004 hptp2301.44719157.0 Nodes 16 00:15:24 Tue Mar 15 03:42:50 2004 hptp2301.44719157.0 Nodes 16 00:15:24 Tue Mar 16 03:42:50 2004 hptp2301.44719157.0 Nodes 16 00:15:24 Tue Mar 16 03:42:50 2004 hptp2301.44719157.0 Nodes 16 00:15:24 Tue Mar 16 03:42:50 2004 hptp2301.44719157.0 Nodes 16 00:15:24 Tue Mar 16 03:42:50 2004 hptp2301.44719157.0 Nodes 16 00:15:24 Tue Mar 16 03:42:50 2004 hptp2301.44719157.0 Nodes 16 00:15:24 Tue Mar 16 03:42:50 2004 hptp23		emos	model	_an _an			1	hpc	.0240 :0230	1.1472	2002.C) op) op	00:04	36600-50	0 00:0
<pre>c entc_lTSMn_uptraj_0</pre>		emos	model	an			1	hpc	b240	1.1472	2003.0) op	00:04	36600-50	0 00:0
: chj2_ifstraj_uptraj_0 16 hpcb2301.4471970.0 np 00:16 19700-50 00:0 chj2_model_fc 4 hpcb2401.4471990.0 np 00:25 19700-52 00:17 : chp4_ifsmin_uptraj_1 16 hpcb2401.4471990.0 np 00:56 19700-72 00:17 : chp4_ifsmin_uptraj_1 16 hpcb2301.4471990.0 np 00:05 19700-72 00:17 : chp4_ifsmin_uptraj_0 17 : chp4_ifstraj_uptraj_0 16 hpcb2401.4471993.0 np 00:00 19700-71 00:02 chth_ifstraj_uptraj_0 16 hpcb2401.4471993.0 np 00:00 19700-70 02:27 chth_ifstraj_uptraj_0 16 hpcb2401.4471993.0 np 00:01 19700-70 02:17 : chth_ifstraj_uptraj_0 16 hpcb2401.4471993.0 np 00:14 19700-50 02:17 : chth_ifstraj_uptraj_0 16 hpcb2401.4471937.0 np 00:14 19700-50 02:17 : chth_ifstraj_uptraj_0 16 hpcb2401.4471937.0 np 00:14 19700-50 02:17 : chth_ifstraj_uptraj_0 18 hpcb2401.4471937.0 np 00:14 19700-50 02:17 : chth_ifstraj_uptraj_0 18 hpcb2401.4471937.0 np 00:11 9700-50 01:22 : chth_ifstraj_uptraj_0 18 hpcb2401.4471937.0 np 00:17 19700-50 01:22 : chth_ifstraj_uptraj_0 18 hpcb2401.4471937.0 np 00:10 19700-50 01:22 : chth_ifstraj_uptraj_0 16 hpcb2401.4471937.0 np 00:10 19700-50 00:52 : chth_ifstraj_uptraj_0 16 hpcb2401.4471950.0 np 00:05 19700-50 00:52 : chth_ifstraj_uptraj_0 16 hpcb2401.4471950.0 np 00:05 19700-50 00:52 : chth_ifstraj_uptraj_0 16 hpcb2401.4471950.0 np 00:01 19700-50 00:22 : chth_ifstraj_uptraj_0 16 hpcb2401.4471950.0 np 00:01 19700-50 00:52 : chth_ifstraj_uptraj_0 16 hpcb2401.4471950.0 np 00:01 19700-50 00:52 : chth_ifstraj_uptraj_0 16 hpcb2401.4471950.0 np 00:03 19700-50 00:52 : chth_ifstraj_uptraj_0 16 hpcb2401.4471950.0 ndes 16 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.4471957.0 Ndes 16 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.4471957.0 Ndes 16 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.4471957.0 Ndes 16 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.447557.0 Ndes 16 00:15:01 Nu Mar 15 19:23:09 2004	j	rdx rdx	ehtc_	ifstra	_uptra <u>:</u> i untra]_0 ai 1	16	hpc	b240 b230	1,1471 1,1471	988.0 579.0) np	00:11	19700-70	0 00:00
<pre>c ehy_model_fc 4 hpcb2401.14719302.0 np 00:52 19700-72 00:71 c ehp4_ifsmin_uptraj_1 16 hpcb2401.14719302.0 np 00:56 19700-72 00:71 c ehp4_ifsmin_uptraj_1 16 hpcb2301.1471493.0 np 00:56 19700-72 00:71 c ehp4_ifstraj_uptraj_0 12 hpcb2401.1471934.0 np 00:00 19700-71 00:00 c ehpk_ifstraj_uptraj_0 12 hpcb2401.1471934.0 np 00:01 19700-70 00:00 c ehpk_ifstraj_uptraj_0 12 hpcb2401.1471934.0 np 00:14 19700-50 02:14 c ehbu_model_fc 16 hpcb2301.1471393.0 np 00:10 19700-50 02:14 c ehbu_model_fc 16 hpcb2301.1471393.0 np 00:10 19700-50 02:14 c ehbu_model_fc 16 hpcb2301.1471393.0 np 00:10 19700-50 01:22 c ehbu_model_fc 16 hpcb2301.1471393.0 np 00:10 19700-50 01:22 c ehbu_model_fc 16 hpcb2301.1471393.0 np 00:114 19700-50 01:22 c ehbu_model_fc 16 hpcb2301.1471393.0 np 00:114 19700-50 01:22 c ehbu_ifstraj_uptraj_0 16 hpcb2401.1471934.0 np 00:17 19700-50 00:22 c ehbu_ifstraj_uptraj_0 16 hpcb2401.1471930.0 np 00:03 19700-50 00:22 c ehbu_ifstraj_uptraj_0 16 hpcb2301.1471593.0 np 00:31 19700-50 00:22 c ehbu_ifstraj_uptraj_0 16 hpcb2301.1471597.0 no 00:3 19700-50 00:20 c hpcb2501{/home/ops/cog}:58\$ runt hpcb2301.1471507.0 nodes 16 cn15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1439147.0 Nodes 16 cn15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1437440.0 Nodes 16 cn15:24 Mon Mar 15 19:23:00 2004 hpcb2301.1437440.0 Nodes 16 cn15:24 Mon Mar 15 19:23:00 2004 hpcb2301.1437280.0 Nodes 16 cn15:41 Mon Mar 15 19:23:00 2004 hpcb2301.14374280.0 Nodes 16 cn15:41 Mon Mar 15 19:23:00 2004 hpcb2301.14374280.0 Nodes 16 cn15:41 Mon Mar 15 19:23:00 2004 hpcb2301.1437280.0 Nodes 16 cn15:41 Mon Mar 15 19:23:00 2004 hpcb2301.1437280.0 Nodes 16 cn15:41 Mon Mar 15 19:23:00 2004 hpcb2301.1437280.0 Nodes 16 cn15:41 Mon Mar 15 19:23:00 2004 hpcb2301.1437280.0 Nodes 16 cn15:41 Mon Mar 15 19:23:00</pre>	1	°dx	ehj2_	ifstra	j_uptr	aj_O	16	hpc	b230	1.1471	070.0) np	00:16	19700-50	0 00:0
chp4_ifsmin_uptraj_1 16 hpcb2301.1471493.0 np 00:56 19700-72 00:17 utd Nodes W.Clock Prio Valt ch14_ifstraj_ifswar 8 hpcb2401.1471494.0 np 00:00 19700-72 00:10 chth_ifstraj_uptraj_0 1 hpcb2401.1471494.0 np 00:00 19700-50 02:24 chth_ifstraj_uptraj_0 1 hpcb2401.1471495.0 np 00:01 19700-50 02:24 chth_ifstraj_uptraj_0 1 hpcb2301.1471394.0 np 00:011 19700-50 02:17 chth_ifstraj_uptraj_0 1 hpcb2301.1471394.0 np 00:011 19700-50 02:24 ethw_imdel_fc 16 hpcb2301.1471394.0 np 00:011 19700-50 01:25 ethw_imdel_fc 16 hpcb2301.1471394.0 np 00:011 19700-50 00:55 ethw_imdel_fc 16 hpcb2401.1471393.0 np 00:011 19700-50 00:25 ethw_imdel_fc 16 hpcb2401.1471893.0 np 00:011 19700-50 00:02 ethw_imstraj_uptraj_0 16 hpcb2401.1471602.0 np 00:03 19700-50 00:05		rdx rdv	ehjx_ eha4	model_ ifsmir	tC Untra	i 1	4	hpc	b240 b240	1.1471 1.1471	1901.0 1902.0) np	00:22	19700-50	0 00:13
ued Nodes W.Clock Prio Walt ehhl_ifstraj_uptraj_0 1 hpcb2401.1471994.0 np 00:00 19700-50 00:22 ehhl_ifstraj_uptraj_0 16 hpcb2401.1471954.0 np 00:01 19700-50 02:24 ehhl_ifstraj_uptraj_0 16 hpcb2301.1471393.0 np 00:01 19700-50 02:17 ehhl_ifstraj_uptraj_0 16 hpcb2301.1471393.0 np 00:01 19700-50 01:25 ehhl_ifstraj_uptraj_0 16 hpcb2301.1471393.0 np 00:01 19700-50 01:25 ehhl_ifstraj_uptraj_0 16 hpcb2301.1471393.0 np 00:01 19700-50 00:25 ehnl_ifstraj_uptraj_0 16 hpcb2401.1471893.0 np 00:01 19700-50 00:25 ehnl_ifstraj_uptraj_0 16 hpcb2401.1471893.0 np 00:01 19700-50 00:22 ehnl_ifstraj_uptraj_0 16 hpcb2401.1471893.0 np 00:01 19700-50 00:22 ehnl_ifstraj_uptraj_0 16 hpcb2401.1471803.0 np 00:14 19700-50 00:22 is	1	rdx	ehp4_	ifsmir	_uptra	j_i	16	hpc	b230	1.1471	499.0	np	00:56	19700-72	2 00:1
<pre>c eh14_fistraj_ifsuar</pre>	ł	oueu	ed				Nod	95					W.Clock	Prio	Wait
<pre>c hpt_lfstraj_uptraj_0 12 hptb2401.447155.0 np 00103 13700-50 02:24 ehbt_lfstraj_uptraj_0 16 hptb2301.447139.0 np 00104 13700-50 01:3 ehbu_model_fc 16 hptb2301.447139.0 np 00114 13700-50 01:3 ehbu_model_fc 16 hptb2301.447139.0 np 00114 13700-50 01:24 ehbu_model_fc 16 hptb2301.447139.0 np 00117 13700-50 01:24 ehbu_model_fc 16 hptb2301.447139.0 np 00117 13700-50 01:24 ehbu_model_fc 16 hptb2401.147184.0 np 00117 13700-50 01:24 ehbu_model_fc 16 hptb2401.147184.0 np 00101 13700-50 00:55 ehpu_model_fc 16 hptb2401.147189.0 np 00101 13700-50 00:25 ehpu_fstraj_uptraj_0 16 hptb2401.147187.0 np 00114 13700-50 00:25 ehpu_fstraj_uptraj_0 16 hptb2401.147187.0 np 00101 13700-50 00:26 ehpu_fstraj_uptraj_0 16 hptb2301.147157.0 ethu_fstraj_uptraj_0 16 hptb2301.14318272.0 Nodes 16 0015:01 Tue Mar 16 03:42:50 2004 hptb2301.1438272.0 Nodes 16 0015:21 Tue Mar 16 01:04:31 2004 hptb2301.1438272.0 Nodes 16 0015:21 Tue Mar 15 16:41:45 2004 hptb2301.1437236.0 Nodes 16 0015:41 tim.sSt 00:15:23 00:15:24 im.70x 00:15:07 00:15:23 00:15:44 im.70x 00:15:07 00:15:23 00:15:44 im.70x 00:15:07 00:15:23 00:15:44 im.70x 00:15:07 00:15:23 00:15:54 im.sSt 20:14:52 00:04 hptb2301.1437256.0 Nodes 16 00:15:21 Tue Mar 16 0:104:31 2004 hptb2301.143757.0 Nodes 16 00:15:21 Tue Mar 16 0:3:42:50 2004 hptb2301.143757.0 Nodes 16 00:15:21 Tue Mar 16 0:3:42:50 2004 hptb2301.143757.0 Nodes 16 00:15:21 Tue Mar 16 0:42:32 0001 hptb2301.143757.0 Nodes 16 00:15:21 Tue Mar 16 0:42:32 0004 hptb2301.143757.0 Nodes 16 00:15:21 Tue Mar 16 0:42:42 0004 hptb2301.143757.0 Nodes 16 00:15:21 Tue Mar 16 0:42:42 0204 hptb2301.1437257.0 Nodes 16 00:15:24 Mon Mar 15 16:41:45 2004</pre>		rdx	eh14_	ifstra	j_ifsv≀	ar	8	hpc	:b240	1.1471	1994.0) np	00:00	19700-7	1 00:0
athbo_model_fc 16 hpcb2301: 1471207:0 np 00:04 19700-60 02:17 athbo_model_fc 16 hpcb2301: 1471333.0 np 00:14 19700-60 01:25 athbo_model_fc 16 hpcb2301: 1471333.0 np 00:10 19700-60 01:25 athbo_model_fc 16 hpcb2301: 1471307.0 np 00:10 19700-60 01:25 athbo_model_fc 16 hpcb2301: 1471507.0 np 00:07 19700-60 00:57 athbo_model_fc 16 hpcb2301: 1471507.0 np 00:07 19700-60 00:56 athbo_model_fstraj_uptraj_0 16 hpcb2401: 1471933.0 np 01:41 19700-50 00:02 athbo_fstraj_uptraj_0 16 hpcb2401: 1471937.0 np 00:141 19700-50 00:02 athbo_fstraj_uptraj_0 16 hpcb2301: 1471602: 0 np 00:03 19700-50 00:00 athbo_fstraj_uptraj_0 16 hpcb2301: 1471602: 0 np 00:03 19700-50 00:00 athbo_fstraj_uptraj_0 16 hpcb2301: 1471602: 0 np 00:03 19700-50 00:00 athbo_fstraj_uptraj_0 16 hpcb2301: 14735440 <td< td=""><td>ľ</td><td>rdx ∙dx</td><td>ehpk_ ehht</td><td>ifstra ifstra</td><td>g_uptra d untra</td><td>a<u>]_</u>O ai_O</td><td>12</td><td>hpc</td><td>b240 b240</td><td>1.1471 1.1471</td><td>563.0 581.0</td><td>) np) nn</td><td>00:03</td><td>19700-50</td><td>0 02:2/ 0 02:4/</td></td<>	ľ	rdx ∙dx	ehpk_ ehht	ifstra ifstra	g_uptra d untra	a <u>]_</u> O ai_O	12	hpc	b240 b240	1.1471 1.1471	563.0 581.0) np) nn	00:03	19700-50	0 02:2/ 0 02:4/
chk0_model_rC 16 hptpb2301.47/339.0 np 02:17 1970-50 01:28 chbv_model_rC 16 hptpb2301.47/398.0 np 00:10 1970-50 01:25 chbv_model_rC 16 hptpb2301.47/398.0 np 00:110 1970-50 00:55 cepq_ifstraj_uptraj_0 16 hptp2301.47/1980.0 np 00:07 19700-50 00:50 cehnz_ifstraj_uptraj_0 16 hptp2301.47/1980.0 np 00:07 19700-50 00:50 cehnz_ifstraj_uptraj_0 16 hptp2301.47/1980.0 np 00:07 19700-50 00:00 cehnz_ifstraj_uptraj_0 16 hptp2301.47/1971.0 np 00:14 19700-50 00:00 cehnz_ifstraj_uptraj_0 16 hptp2301.147/102.0 np 00:03 19700-50 00:00 ptp25501{/home/ops/cog}:58\$ runt hptp2301.147162.0 np 00:03 1970-50 00:00 ptp25501{/home/ops/cog}:58\$ runt hptp2301.1471632.0 np 00:03 1970-50 00:00 ptp25501{/home/ops/cog}:58\$ runt hptp2301.1471632.0 np 00:04 16 10:15:17 times Date Notes 16 10:15:17 10:15:14 10:15:14 10:15:14<	ł	rdx	ehbv_	model_	fc.	-J_0	16	hpc	b230	1.1471	207.0) np	00:04	19700-50	0 02:1:
abbmodel_rc 16 hpcb2301:1471338.0 mp 00:10 19700-50 01:22 c ehbifstraj_uptraj_0 16 hpcb2401.1471898.0 mp 00:17 19700-50 00:55 c embmodel_rc 16 hpcb2401.1471898.0 mp 00:10 19700-50 00:52 c ehbifstraj_uptraj_0 16 hpcb2401.1471930.0 mp 10:17 19700-50 00:22 c ehbifstraj_uptraj_0 16 hpcb2401.1471937.0 mp 00:14 19700-50 00:22 c ehbifstraj_uptraj_0 16 hpcb2401.1471937.0 mp 00:14 19700-50 00:23 c ehbifstraj_uptraj_0 16 hpcb2301.1471602.0 np 00:3 19700-50 00:00 pcb2501{/home/ops/cog}:565 runt hpcb2301.1471602.0 np 00:3 19700-50 00:00 pcb2501{/home/ops/cog}:508 runt hpcb2301.1471602.0 np 00:3 19700-50 00:00 pcb2501{/home/ops/cog}:509 2004 hpcb2401.1479872.0 nodes 16 i15:01 Tue Mar 16 03:42:50 2004 hpcb2301.1437602.0 nodes 16 i0:15:24 Mon Mar 15 16:31:40 00:15:44 00:4139272.0 Nodes 16	F	rdx rdv	ehkO_	model_ ifstra	fC d untr	ai o	16	hpc	b230	1.1471 1.4774	339.0) np	02:17	19700-50	0 01:3
<pre>c ehk_ifstraj_uptraj_0 16 hpcb2401.4474847.0 np 00:17 19700-50 00:57 eqq0.ifstraj_uptraj_0 8 hpcb2401.4474830.0 np 00:05 19700-50 00:50 c ehm6_model_fc 16 hpcb2401.4474830.0 np 00:07 19700-50 00:28 c ehr2_ifstraj_uptraj_0 16 hpcb2401.4471971.0 np 00:107 19700-50 00:28 c ehr2_ifstraj_uptraj_0 16 hpcb2401.1471971.0 np 00:13 19700-50 00:20 c ehr2_ifstraj_uptraj_0 16 hpcb2301.1471602.0 np 00:03 19700-50 00:00 pcb2501{/home/ops/cog}:58\$ runt hpcb2301.1471602.0 np 00:03 19700-50 00:00 pcb2501{/home/ops/cog}:58\$ runt hpcb2301.1471657.0 c rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj i times Date Notes 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.1433917.0 Nodes 16 00:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1438272.0 Nodes 16 00:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.14387280.0 Nodes 16 00:15:24 Mon Mar 15 19:24:00 A hpcb2401.1437286.0 Nodes 16 00:15:44 Mon Mar 15 19:24:00 O:01:5:33 im.95% 00:14:52 00:15:33 l Dev : 00:00:15 read : 1% Standard Deviation to Average Ratio b2501{/home/ops/cog}:53\$ g/dar/ehn0/an/main/xx/an/4dvar/ifstraj < c: rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj < c: rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj i times Date Notes 00:17:28 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2301.1473557.0 Nodes 16 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2301.1473557.0 Nodes 16 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2301.14337236.0 Nodes 16 00:15:01 Tue Mar 15 01:41:45 2004 hpcb2301.14337236.0 Nodes 16 00:15:24 Mon Mar 15 18:41:45 2004 hpcb2301.14337240.0 Nodes 16 00:15:44 Mon Mar 15 18:41:45 20</pre>		rdx	ehbv_	model_	fc	ajio	16	hpc	b230	1.1471	398.0) np	00:10	19700-50	0 01:2
<pre>cyq0_risting_uptraj_0 is hptp2401.1471858.0 np 00.00 is hold=50 00:80 ehms_insdel_fc 16 hptp2401.1471933.0 np *1-day 19700-50 00:22 ehm_ifstraj_uptraj_0 16 hptp2401.1471971.0 np 00:03 19700-50 00:00 pptp2501{/home/ops/cog}:58\$ runt hptp2301.1471657.0 runt hptp2301.14371444.0 Nodes 16 runt hptp2301.14372444.0 Nodes 16 runt hptp2301.1437444.0 Nodes 16 runt hptp2301.1437444.0 Nodes 16 runt hptp2301.1437444.0 Nodes 16 runt hptp2301.14</pre>		rdx	ehhx_	ifstra	j_uptr	aj_O	16	hpo	b240	1.1471	847.0) np	00:17	19700-50	
<pre>c ehr2_ifstraj_uptraj_0 16 hpcb2401.1471933.0 np *1-day 19700-50 00:22 c ehhu_ifstraj_uptraj_0 16 hpcb2401.1471937.0 np 00:14 19700-50 00:00 pcb2501{/home/ops/cog}:58\$ runt hpcb2301.1471557.0 c rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj times Date Notes</pre>		rux rdx	ehm6_	model_	j_uptr fc	aj_U	16	hpc	.0240 :0230	1.1471	1889. U 1510. C) np) np	00:05	19700-50	0 00:5
<pre>c ennu_fistraj_uptraj_0</pre>	1	rdx	ehr2_	ifstra	j_uptr	aj_O	16	hpc	b240	1.1471	933.0) np	*1-day	19700-50	0 00:2
pcb2501{/home/ops/cog}:58\$ runt hpcb2301.1471557.0 : rdx=ehn0_ifstraj_4dvar=/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj i times Date Notes 0:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.1439117.0 Nodes 16 0:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1437444.0 Nodes 16 0:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 0:15:44 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 0:15:41 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 0:15:01 00:15:23 00:15:38 im. 95% 00:14:52 00:15:33 1 Dev : 00:00:15 read : 1% Standard Deviation to Average Ratio b2501{/home/ops/cog}:59\$ g/dar/ehn0/an/main/xx/an/4dvar/ifstraj (times Date Notes 10:15:21 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1471557.0 Nodes 16 10:15:21 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1437444.0 Nodes 16 10:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1437444.0 Nodes 16 10:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1437457.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437444.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:48 00:114:04 00:17:32 10:15:49 00:14:04 00:17:32 10:15:40 00:14:56 00:16:40 10:15:40 00:14:56 00:16:40 10:15	i	rax rdx	ennu_ eh8z	ntstra <mark>matchu</mark>	ij_uptra Ip_feed	aj_u back	16	hpc	b240 b230	1,1471 1,1471	1971. U 1602. C) np) np	00:14	19700-50	0 00:14
<pre>: rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj i times Date Notes 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.1439117.0 Nodes 16 00:15:24 Mon Mar 15 10:29:09 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 00:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 00:15:01 00:15:07 00:15:38 1m.95% 00:14:52 00:15:33 1Dev : 00:00:15 red : 13 Standard Deviation to Average Ratio bb2501{/home/ops/cog}:53\$ g/dar/ehn0/an/main/xx/an/4dvar/ifstraj < :: rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj i times Date Notes 10:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1471557.0 Nodes 16 00:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1471557.0 Nodes 16 00:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.147357.0 Nodes 16 00:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.147357.0 Nodes 16 00:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1437444.0 Nodes 16 00:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 19:290 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 19:290 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 19:290 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 16:441:45 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 16:441:45 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437444.0 Nodes 16 00:15:04 00:141:56 00:16:40 1m.95% 00:141:04 00:17:28 1m.70% 00:141:04 00:17:28 1m.70% 00:141:04 00:17:22 Dev : 00:00:52 read : 5% Standard Deviation to Average Ratio bb2501{/home/ops/cog}:60\$</pre>	7	Chp	cb2501{	/home/	ops/co	g}:58\$	runt	hpcb	2301	14715	557.0				
<pre>: Fux-enn0_fistraj_4uvar-/hpcb/rux_uir/iog/uar/enn0/an/main/xx/an/4uvar/fistraj i times Date Notes 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2301.1439272.0 Nodes 16 00:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1437296.0 Nodes 16 00:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437296.0 Nodes 16 00:15:24 Mon Mar 15 19:29:09 2004 hpcb22401.1437296.0 Nodes 16 00:15:44 Mon Mar 15 19:29:09 2004 hpcb22401.1437296.0 Nodes 16 00:15:01 00:15:23 00:15:44 im.70% 00:15:07 00:15:33 1 Dev : 00:00:15 read : 1% Standard Deviation to Average Ratio bb2501{/home/ops/cog}:53\$ g/dar/ehn0/an/main/xx/an/4dvar/ifstraj < :: rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj i times Date Notes 00:17:28 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2301.1471557.0 Nodes 16 00:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1437444.0 Nodes 16 00:15:21 Tue Mar 15 19:29:09 2004 hpcb2301.14372464.0 Nodes 16 00:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437296.0 Nodes 16 00:15:44 Mon Mar 15 19:29:09 2004 hpcb2301.1437296.0 Nodes 16 00:15:40 00:14:04 00:17:32 1 Dev : 00:00:52 read : 5% Standard Deviation to Average Ratio bb2501{/home/ops/cog}:80\$</pre>		.	under alle					لم بالم	(d. o. 77)	/	. / . h c			. Inn Indunnu	
times Date Notes 00:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.143917.0 Nodes 16 00:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1438272.0 Nodes 16 00:15:24 Mon Mar 15 13:23:09 2004 hpcb2301.1437444.0 Nodes 16 00:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437444.0 Nodes 16 00:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 00:15:01 00:15:23 00:15:38 im. 70x 00:15:07 00:15:38 im. 95x 00:14:52 00:15:53 Dev : 00:00:15 00:15:38 read : 1% Standard Deviation to Average Ratio bb2501{/home/ops/cog}:53\$ g/dar/ehn0/an/main/xx/an/4dvar/ifstraj : rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj : rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj : rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an.1433272.0 Nodes 16 : no:15:24 Mon Mar 15 16:2004 hpcb2301.1471657.0 Nodes 16 : no:15:24 Mon Mar 15 18:2004 hpcb2301.14337444.0 Nodes 16 : no:15:24 Mon Mar 15 18:41:45 2004 hpcb2301.1437444.0 Node		100:	rux-en	nu_irs	(traj_4)	uvar-/	npc0/	rax_a	nryn)y/uar	-yennu	17 an 7	main/xx	/an/4uvar,	ritstra
00:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.1439117.0 Nodes 16 10:15:24 Tue Mar 16 01:04:31 2004 hpcb2301.1438272.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2401.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:01 00:15:23 00:15:44 10:15:01 00:15:23 00:15:38 10:20:15:01 00:15:23 00:15:53 10:20:00:015 Tead : 1% Standard Deviation to Average Ratio 10:20:15:01 53 10:20:15:01 00:15:23 00:15:53 10:20:15:01 00:15:53 10:20:15:01 00:15:53 10:20:15:01 00:15:53 10:20:15:01 00:15:53 10:20:15:01 00:15:53 10:20:15:01 00:15:53 10:20:15:01 00:15:23 00:4 hpcb2301.1471557.0 Nodes 16 10:15:01 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:15:01 Tue Mar 16 03:42:50 2004 hpcb2301.143917.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1438272.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1438272.0 Nodes 16 10:15:25 Min Avg Max Doi:17:28 10:25:20:25 Min Avg Max Doi:17:28 10:25:20:25 Min Avg Ma	l	Run	times	Date	:				N	otes					
N0:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1438272.0 Nodes 16 N0:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 N0:15:44 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 N0:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 S Min Avg Max 00:15:01 00:15:23 00:15:44 S Min OU:15:07 00:15:38 1 Dev : 00:00:15 Yead : 1% Standard Deviation to Average Ratio b25001{/home/ops/cog}:59\$ g/dar/ehn0/an/main/xx/an/4dvar/ifstraj (times Date Notes 10:15:21 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:15:21 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1437444.0 Nodes 16 10:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1437444.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437296.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:24 Mon Mar 15 16:41 Mor Mar 20:15:48 No:17:28 10:15:40 Note 15:48 No:17:28 10:15:40 No:15:48 No:17:28 10:15:40 No:15:40 No:15:48 10:15:40 No:15:40 No:15:48 10:15:40 No:15:40 No:	ľ	00	:15:01	Tue	Mar 16	03:42	: 50 2	004	h	ocb240	01.143	9117	.O Node	s 16	
00:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437494.0 Modes 16 00:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 		00	:15:21	Tue	Mar 16	01:04	:31 2	004	hj	ocb230	01.143	8272	2.0 Node	s 16	
S Min Aug Max 00:15:01 00:15:23 00:15:44 im.95X 00:14:52 00:15:33 Dev : 00:00:15 00:15:33 read : 1% Standard Deviation to Average Ratio		00	:15:24	Mon	Mar 15 Mar 15	19:29	:09 2 :45 2	004	ni hi	ocb230 ocb240)1.143)1.143	37444 37296	1.0 Node 3.0 Node	s 16 s 16	
Init HVy Max 00:15:01 00:15:23 00:15:44 im.70% 00:14:52 00:15:38 im.95% 00:14:52 00:15:53 I Dev : 00:00:15 00:15:23 00:15:38 im.95% 00:15:97 00:15:53 I Dev : 00:00:15 00:15:23 00:17:53 im.95% 00:15:25 g/dar/ehn0/an/main/xx/an/4dvar/ifstraj ib2501{/home/ops/cog}:58\$ g/dar/ehn0/an/main/xx/an/4dvar/ifstraj itimes Date Notes	-					 Aua									
<pre>im.70% 00:15:07 00:15:38 im.95% 00:14:52 00:15:53 l Dev : 00:00:15 read : 1% Standard Deviation to Average Ratio bb2501{/home/ops/cog}:55\$ g/dar/ehn0/an/main/xx/an/4dvar/ifstraj </pre> : rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj i times Date Notes	2	tuns 1		00:15	5: 01	00:15	: 23	00:	15:4	4					
<pre>Im. 95% 00:14:52 00:15:53 I Dev : 00:00:15 read : 1% Standard Deviation to Average Ratio bb2501{/home/ops/cog}:59\$ g/dar/ehnO/an/main/xx/an/4dvar/ifstraj </pre> <pre> : rdx-ehnO_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehnO/an/main/xx/an/4dvar/ifstraj i times Date Notes</pre>	1	sti	m.70%	00:15	5:07			00:	15:3	3					
<pre>L Dev : 00:00:15 read : 1% Standard Deviation to Average Ratio read : 1% Standard Deviation to Average Ratio rest : rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj rest : rdx-ehn0_ifstraj_4dvar/ehn0/an/main/xx/an/4dvar/ifstraj rest : rdx-ehn0_ifstraj_4dvar/ehn0/an/main/xx/an/4dvar/ifstraj rest : rdx-ehn0_ifstraj_4dvar/ehn0/an/main/xx/an/4dvar/ifstraj rest : rdx-ehn0_ifstraj_rest : rdx-frame/ops/cog}: eos rest : rdx-ehn0_ifstraj_rest : rdx-frame/rest : rdx-frame/rest : rdx-ehn0/rest : rdx-frame/rest : rdx-ehn0/rest : rdx-</pre>	E	sti	m.95%	00:14	1:52			00:	15:5	3					
Yead : 1% Standard Deviation to Average Ratio (b2501{/home/ops/cog}:59\$ g/dar/ehnO/an/main/xx/an/4dvar/ifstraj : rdx-ehnO_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehnO/an/main/xx/an/4dvar/ifstraj :: rdx-ehnO_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehnO/an/main/xx/an/4dvar/ifstraj :: rdx-ehnO_ifstraj_4dvar/ifstraj :: rdx-ehno_ifstraj_2dvar/ifstraj	-	std	Dev : O	0: 00: 1	5										
: rdx-ehnO_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehnO/an/main/xx/an/4dvar/ifstraj i times Date Notes 10:17:28 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:15:01 Tue Mar 16 03:42:50 2004 hpcb2301.1439117.0 Nodes 16 10:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1438272.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 15 Min Avg Max 00:15:01 00:15:48 00:17:28 16.10:16:40 17.32 10.20:52 10.20:52 10.20:52 10.20:52 10.20:52 10.20:52 10.20:52 10.20:52 11.20:52 12.20:52 13.20:52 14.20:52 14.20:52 14.20:52 15	Ě	Spre. Spre.	ad : 1: 2501{/h	% Star ome/or	idard Di Is/cog}	eviati :59\$ a	on to /dar/i	Aver ehnO/	rage ∣ ′an∠m	Ratio ain/xo	(/an/4	lduar	/ifstra	i	<
: rdx-ehn0_ifstraj_4dvar-/hpcb/rdx_dir/log/dar/ehn0/an/main/xx/an/4dvar/ifstraj times Date Notes 10:17:28 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:15:21 Tue Mar 16 03:42:50 2004 hpcb2301.1439117.0 Nodes 16 10:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1439744.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 15 Min Avg Max 00:15:01 00:15:48 00:17:28 16	1	iper.	2001071	0.000, 04					wit7 in-		se ante 1		,		
I times Date Notes 10:17:28 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:17:28 Tue Mar 16 03:42:50 2004 hpcb2301.1439117.0 Nodes 16 10:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1438272.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:45 00:17:28 11:		Toh	rdy-eb	n∩ ifa	trai A	duar-/	hnch/	rdv d	lin/li	na/dau	·/ehnC	Van	main/vv	/an/Aduar	/ifstra
times Date Notes 10:17:28 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 10:17:28 Tue Mar 16 03:42:50 2004 hpcb2301.143917.0 Nodes 16 10:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1439272.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 10:15:24 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 10:15:45 00:17:28 00:14:40 00:17:28 11m. 95% 00:14:04 00:17:32 10 12ev : 00:00:52	ļ	J00.	Tux-en	10_115	rtiaj_4	uva1-/1	прсол	ru.~_u		Jy/uai	7 enne	// all/	109 TH7 AA	,/ all/ 4 uval ,	riistia
00:17:28 Tue Mar 23 14:50:35 2004 hpcb2301.1471557.0 Nodes 16 100:15:01 Tue Mar 16 03:42:50 2004 hpcb2301.1439117.0 Nodes 16 100:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1439272.0 Nodes 16 100:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437444.0 Nodes 16 100:15:44 Mon Mar 15 16:41:45 2004 hpcb2301.1437296.0 Nodes 16 100:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 100:15:01 00:15:48 00:17:28 100:15:01 00:15:48 00:17:28 100:15:00 00:14:56 00:16:40 110:95% 00:14:04 00:17:32 1 Dev : 00:00:52 read : 5% Standard Deviation to Average Ratio 152501{/home/ops/cog}:60\$ ■		Run	times	Date	:				N	otes					
10:15:01 Tue Mar 16 03:42:50 2004 hpcb2401.1439117.0 Nodes 16 10:15:21 Tue Mar 16 01:04:31 2004 hpcb2301.1438272.0 Nodes 16 10:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437244.0 Nodes 16 10:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 11:5:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 11:5:40 00:15:01 00:15:48 00:17:28 11:5:40 00:14:56 00:16:40 11:9:5% 00:14:04 00:17:32 11 Dev : 00:00:52 *ead : 5% Standard Deviation to Average Ratio 15:5:2501{/home/ops/cog}:60\$	l	00	:17:28	Tue	Mar 23	14:50	: 35 2	004	h	ocb230	01.147	1557	.O Node	s 16	
UC:15:21 Tue Mar 15 01:04:31 2004 hpcb2301.14382/2.0 Nodes 16 UC:15:24 Mon Mar 15 19:29:09 2004 hpcb2301.1437244.0 Nodes 16 UC:15:24 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 UC:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 UC:15:01 00:15:48 00:17:28 Tim.70% 00:14:50 00:15:48 00:16:40 Tim.95% 00:14:04 00:17:32 I Dev : 00:00:52 Yead : 5% Standard Deviation to Average Ratio bb2501{/home/ops/cog}:60\$ ■		00	:15:01	Tue	Mar 16	03:42	:50 2	004	hj	ocb240	01.143	9117	.O Node	s 16	
NO:15:44 Mon Mar 15 16:41:45 2004 hpcb2401.1437296.0 Nodes 16 IS Min Avg Max OO:15:01 00:15:48 00:17:28 Tim.70% 00:14:56 00:16:40 Tim.95% 00:14:04 00:17:32 I Dev : 00:00:52 read : 5% Standard Deviation to Average Ratio b2501{/home/ops/cog}:60\$ ■		00	:15:21	Mon	Mar 16 Mar 15	19:29	:31 2 :09 2	004 004	n n hi	ocb230 ocb230	J1.143 01.143	37444	. U Node I. O Node	s 16 s 16	
IS Min Avg Max O0:15:01 O0:15:48 O0:17:28 Tim.70% O0:14:56 O0:16:40 Tim.95% O0:14:04 O0:17:32 I Dev : O0:00:52 Yead : 5% Standard Deviation to Average Ratio Tb2501{/home/ops/cog}:60\$ ■		00	: 15: 44	Mon	Mar 15	16:41	:45 2	004	ĥ	ocb240	01.143	7296	.O Node	s 16	
00:15:01 00:15:48 00:17:28 im.70% 00:14:56 00:16:40 im.95% 00:14:04 00:17:32 I Dev : 00:00:52 read : 5% Standard Deviation to Average Ratio ib2501{/home/ops/cog}:60\$	•	Runs		Min		<u>Aun</u>		Max	,						
:1m,70% 00:14:56 00:16:40 im,95% 00:14:04 00:17:32 Dev : 00:00:52 read : 5% Standard Deviation to Average Ratio :b2501{/home/ops/cog}:60\$ ∎		5		00:15	5: 01	00:15	: 48	00:	17:2	з					
Hmi36% 00:14.04 00:17.52 Dev : OO:00:52 read : 5% Standard Deviation to Average Ratio b2501{/home/ops/cog}:60\$ ■		Esti	m.70%	00:14	1:56			00:	16:4	2					
Dev : 00:00:52 'ead : 5% Standard Deviation to Average Ratio b2501{/home/ops/cog}:60\$ ∎		esen		00.14	1.04			00:	17.3	-					
b2501{/home/ops/cog}: 60\$		Std I Snre	Dev : O ad · E	0:00:5 % star	52 Idard D	euisti	on to	6110 ×	ane l	Ratio					
		hpcb:	2501{/h	ome/op	is/cog}	: 60\$	on co	HVET	agel	acro					

Figure 7 - Il_jobs in non-operational mode (with no mention of nodes reserved), where Running jobs, Queued jobs and 2 sets of predicted run-times for job ehn0_ifstraj_4dvar are shown.

Figure 7 (the top half) shows ll_jobs in a non-operational period – Running jobs first, below this Queued jobs. No mention is made of nodes being reserved. Running jobs are listed top down in the order that they are expected to complete. Some jobs are highlighted in colour. The job in red has run for longer than predicted. Jobs in green started in the previous 3 minutes. Queued jobs are listed in the order they are expected to be dispatched according to SYSPRIO and UserPriority. Jobs in blue have been waiting for more than 1 hour, jobs in yellow are new and were submitted in the previous 2 minutes.

As there is a job that has run for longer than predicted I have extracted the original 4 run-times from which a predicted run-time of 00:15:53 was calculated and the 5 run-times, from which a new predicted run-time of 00:17:32 is calculated.

17. How many FloatingCpus have been reserved?

As mentioned before during operational periods, the value of FloatingCpus will be less than 936. The scripts fcpus can be used to check the value of FloatingCpus and the relationship with CPUs reserved.

≥ -M hpcb <3>	• • ×
hpcb2501{/home/cos/newops/bin}:15\$ fcpus LoadL_config is a symbolic link to /loadl/LoadL_config.040324-095001 FLOATING_RESOURCES = FloatingCpus(8) FloatingMSCpus(0)	
928 cpus reserved for operations 8 cpus available for general use	
hpcb2501{/home/cos/newops/bin}:16\$	

Figure 8 - a display showing the relationship between the value of FloatingCpus set a CPUs reserved.

When FloatingCpus is set to 8, 928 CPUS (116 nodes) are reserved (ask me later why FloatingCpus is not set to 0), and as the value of FloatingCpus is 8 (<936) ll_jobs will show more information -see Figure 9 – below.

Nodes UCIOCK Prior Finish enos prodgen_042 1 hptb2401.1475471.0 op 00.02 36600-90 00:00 enos prodgen_045 1 hptb2401.1475474.0 op 00:02 36600-90 00:02 rdx ehbt_ifstraj_uptraj_0 16 hptb2401.1475474.0 op 00:16 19700-80 00:07 enos 02_pf 4 hptb2301.14775052.0 op 00:26 36600-70 00:21 enos 03_pf 4 hptb2301.1475055.0 op 00:26 36600-70 00:21 enos 03_pf 4 hptb2401.1475455.0 op 00:26 36600-70 00:21 enos 03_pf 4 hptb2401.1475455.0 op 00:26 36600-70 00:21 enos 03_pf 4 hptb2401.1475452.0 op 00:26 36600-70 00:21 enos softf 4 hptb2401.1475452.0 op 00:26 36600-70 00:24 enos softf 4 hptb2401.1475452.0 op 00:26 36600-70<	Nodes Weil Mar 24 Prior Prio Prio Prior	- npc	.D									
Number Prodgen_042 Number Nu	Control Codes Codes <thcods< th=""> Codes Codes <t< th=""><th>lodes</th><th>AVATI OP/N</th><th>0 2/2</th><th>Drained</th><th>0/0 </th><th>praining</th><th>0/0</th><th>Wed Ma</th><th>ar 24 10:</th><th>13:34</th><th>2004</th></t<></thcods<>	lodes	AVATI OP/N	0 2/2	Drained	0/0	praining	0/0	Wed Ma	ar 24 10:	13:34	2004
Description Description <thdescription< th=""> <thdescription< th=""></thdescription<></thdescription<>	<pre>mass product 042</pre>		y prodaon 04:	2	NOUN	es bach2	101 1475	171 0 4	W.CIU	JUK PI	niu F	00:00
Asymptotic Implementation Implementat	<pre>mps productinguptraj_0 16 hpcb2401.1475048.0 np 00014 19700-80 00107 dx ehba_ifstraj_uptraj_0 16 hpcb2401.1475088.0 np 00114 19700-80 00107 mmos 01_pf 4 hpcb2401.1475052.0 op 00126 36600-70 00121 mmos 03_pf 4 hpcb2401.1475054.0 op 00126 36600-70 00121 mmos 03_pf 4 hpcb2401.1475054.0 op 00126 36600-70 00121 mmos 04_pf 4 hpcb2401.1475054.0 op 00126 36600-70 00124 mmos 05_pf 4 hpcb2401.1475451.0 op 00126 36600-70 00124 mmos 05_pf 4 hpcb2401.1475451.0 op 00126 36600-70 00124 mmos 05_pf 4 hpcb2401.1475421.0 op 00126 36600-70 00124 mmos 07_pf 4 hpcb2401.1475421.0 op 00126 36600-70 00124 mmos 07_pf 4 hpcb2401.1475421.0 op 00126 36600-70 00104 mmos 10_pf 4 hpcb2401.1475421.0 op 00126 36600-70 00104 mmos 11_pf 4 hpcb2401.1475452.0 op 00126 36600-70 00104 mmos 12_pf 4 hpcb2401.1475452.0 op 00126 36600-70 00104 mmos 12_pf 4 hpcb2401.1475463.0 op 00126 36600-70 00104 mmos 12_pf 4 hpcb2401.1475463.0 op 00126 36600-70 00104 mmos 12_pf 4 hpcb2401.1475053.0 op 00126 36600-70 00104 mmos 13_pf 4 hpcb2401.1475463.0 op 00126 36600-70 00104 mmos 13_pf 4 hpcb2401.1475463.0 op 00126 36600-70 00104 mmos 13_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00102 mmos 14_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00104 mmos 13_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00102 mmos 14_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00104 mmos 13_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00104 mmos 13_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00104 mmos 13_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00102 mmos 14_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00102 mmos 14_pf 4 hpcb2301.1475053.0 op 00126 36600-70 00104 mmos 13_pf 4 hpcb2301.1475053.0 op 00126 36600-70</pre>	mos	prougen_04	2		hpcb2	401.1473 404 4475	174 0 4	γρ 00:0 γρ 00:0	JZ 3000 10 0000	00-30	00:00
uinc_listraj_uptraj_0 16 hptD2401.1475082.0 op 00:16 13700-80 00:07 mos 02_pf 4 hptD2301.14747348.0 np 00:16 13700-80 00:07 00:26 36600-70 00:21 mos 01_pf 4 hptD2301.1475052.0 op 00:26 36600-70 00:21 mos 02_pf 4 hptD2301.1475053.0 op 00:26 36600-70 00:21 mos 04_pf 4 hptD2301.1475053.0 op 00:26 36600-70 00:21 mos 06_pf 4 hptD2301.1475053.0 op 00:22 36600-70 00:24 mos 06_pf 4 hptD2301.1475053.0 op 00:26 36600-70 00:24 mos 06_pf 4 hptD2301.1475053.0 op 00:26 36600-70 00:24 mos 06_pf 4 hptD2301.1475053.0 op 00:26 36600-70 00:24 mos 07_pf 4 hptD2401.1475452.0 op 00:26 36600-70 00:24 mos 07_pf 4 hptD2401.1475452.0 op 00:26 36600-70 00:04 mos 07_pf 4 hptD2301.1475053.0 op 00:26 36600-70 00:04 </td <td>uinc_lifstraj_uptraj_0 16 hptb2401.1475438.0 np 00116 19700-80 001014 nmos 02_pf 4 hptb2301.14775052.0 op 00126 36600-70 00121 nmos 03_pf 4 hptb2301.1475053.0 op 00126 36600-70 00121 nmos 03_pf 4 hptb2301.1475053.0 op 00126 36600-70 00121 nmos 04_pf 4 hptb2301.1475053.0 op 00126 36600-70 00121 nmos 05_pf 4 hptb2301.1475053.0 op 00126 36600-70 00124 nmos 05_pf 4 hptb2301.1475053.0 op 00126 36600-70 00124 nmos 05_pf 4 hptb2301.1475053.0 op 00126 36600-70 00124 nmos 05_pf 4 hptb2401.1475452.0 op 00121 36600-70 00124 nmos 02_pf 4 hptb2401.1475453.0 op 00121 36600-70 00124 nmos 03_pf 4 hptb2401.1475453.0 op 00126 36600-70 00124 nmos 02_pf 4 hptb2401.1475453.0 op 00126 36600-70 00104 nmos 02_pf</td> <td>dv.</td> <td>prougen_04</td> <td>o si untroj</td> <td>0 40</td> <td>hpcb2</td> <td>401.1475</td> <td>+/4.0 C</td> <td>p 00:0</td> <td>12 3000</td> <td>0-30</td> <td>00:02</td>	uinc_lifstraj_uptraj_0 16 hptb2401.1475438.0 np 00116 19700-80 001014 nmos 02_pf 4 hptb2301.14775052.0 op 00126 36600-70 00121 nmos 03_pf 4 hptb2301.1475053.0 op 00126 36600-70 00121 nmos 03_pf 4 hptb2301.1475053.0 op 00126 36600-70 00121 nmos 04_pf 4 hptb2301.1475053.0 op 00126 36600-70 00121 nmos 05_pf 4 hptb2301.1475053.0 op 00126 36600-70 00124 nmos 05_pf 4 hptb2301.1475053.0 op 00126 36600-70 00124 nmos 05_pf 4 hptb2301.1475053.0 op 00126 36600-70 00124 nmos 05_pf 4 hptb2401.1475452.0 op 00121 36600-70 00124 nmos 02_pf 4 hptb2401.1475453.0 op 00121 36600-70 00124 nmos 03_pf 4 hptb2401.1475453.0 op 00126 36600-70 00124 nmos 02_pf 4 hptb2401.1475453.0 op 00126 36600-70 00104 nmos 02_pf	dv.	prougen_04	o si untroj	0 40	hpcb2	401.1475	+/4.0 C	p 00:0	12 3000	0-30	00:02
unx enda_instrug_uptrag_0 in hptDz201.147474505.0 op 00:26 36600-70 00:21 mmos 02_pf 4 hptDz201.1475052.0 op 00:26 36600-70 00:21 mmos 03_pf 4 hptDz201.1475055.0 op 00:26 36600-70 00:21 mmos 03_pf 4 hptDz201.1475055.0 op 00:26 36600-70 00:21 mmos 04_pf 4 hptDz201.1475055.0 op 00:26 36600-70 00:21 mmos 04_pf 4 hptDz201.1475052.0 op 00:26 36600-70 00:21 mmos 05_pf 4 hptDz201.1475052.0 op 00:22 36600-70 00:21 mmos 05_pf 4 hptDz201.1475052.0 op 01:12 36600-70 00:24 mmos 07_pf 4 hptDz201.147542.0 op 00:25 36600-70 00:04 mmos 07_pf 4 hptDz201.1475452.0 op 00:26 36600-70 00:04 mmos 02_pf 4	unx e100a_115(raj_uptraj_0 Test hptb2201.1475052.0 op 000:26 36600-70 000:21 minos 02_pf 4 hptb2201.1475055.0 op 000:26 36600-70 000:21 minos 03_pf 4 hptb2201.1475055.0 op 000:26 36600-70 000:21 minos 03_pf 4 hptb2201.1475053.0 op 000:26 36600-70 000:21 minos 06_pf 4 hptb2201.1475053.0 op 000:26 36600-70 000:24 minos 06_pf 4 hptb2201.1475053.0 op 000:26 36600-70 000:24 minos 06_pf 4 hptb2201.1475452.0 op 000:26 36600-70 000:24 minos 06_pf 4 hptb2401.1475452.0 op 001:41 36600-70 000:44 minos 07_pf 4 hptb2401.1475453.0 op 000:26 36600-70 000:24 minos 07_pf 4 hptb2401.1475453.0 op 000:26 36600-70 000:04 minos 07_pf 4 hptb2401.1475453.0 op 000:26 36600-70 000:04 <t< td=""><td>ux dv</td><td>ennt_fistr</td><td>aj_uptraj</td><td>_0 16</td><td>hncho</td><td>101,1475</td><td>740 0 F</td><td>ip 00:1</td><td>14 1370</td><td>00-00</td><td>00:04</td></t<>	ux dv	ennt_fistr	aj_uptraj	_0 16	hncho	101,1475	740 0 F	ip 00:1	14 1370	00-00	00:04
minus 01_pf 4 hptb2301.1475053.0 0p 00.26 36800-70 00.21 minus 01_pf 4 hptb2301.1475053.0 0p 00.26 36800-70 00.21 minus 03_pf 4 hptb2301.1475053.0 0p 00.26 36600-70 00.21 minus 04_pf 4 hptb2301.1475053.0 0p 00.26 36600-70 00.28 minus 06_pf 4 hptb2301.1475053.0 0p 00.26 36600-70 00.28 minus 06_pf 4 hptb2301.1475452.0 op 00.26 36600-70 00.28 minus 06_pf 4 hptb2301.1475452.0 op 00.26 36600-70 00.28 minus 06_pf 4 hptb2401.1475462.0 op 00.112 36600-71 00.02 minus 07_pf 4 hptb2401.1475462.0 op 00.26 36600-70 00.02 minus 07_pf 4 hptb2401.1475462.0 op 00.26 36600-70 00.02 minus 07_pf 4 hptb2401.1475462.0 op 00.26 36600-70 00.02 minus 07_pf 4 hptb2401.1475463.0 op 00.26 36600-70 00.02 36600-70 00.02	<pre>mms 02_pf 4 hpcb200.1475450.0 op 00:28 36600-70 00:21 mms 03_pf 4 hpcb200.1475450.0 op 00:28 36600-70 00:21 mms 04_pf 4 hpcb201.1475455.0 op 00:28 36600-70 00:24 mms 04_pf 4 hpcb201.1475455.0 op 00:28 36600-70 00:24 mms model_fc 36 hpcb201.1475450.0 op 00:28 36600-70 00:24 mms model_fc 16 hpcb201.1475450.0 op 00:28 36600-70 00:24 mms model_fc 16 hpcb201.1475450.0 op 00:28 36600-70 00:24 mms model_fc 6 hpcb201.1475450.0 op 00:28 36600-70 00:24 mms 03_pf 4 hpcb201.1475450.0 op 00:28 36600-70 00:24 mms model_fc 6 hpcb201.1475450.0 op 00:28 36600-70 00:24 mms 02_pf 4 hpcb201.1475450.0 op 00:28 36600-70 00:04 mms 03_pf 4 hpcb201.1475450.0 op 00:28 36600-70 00:04 mms 09_pf 4 hpcb201.1475450.0 op 00:28 36600-70 00:04 mms 11_pf 4 hpcb201.1475450.0 op 00:28 36600-70 00:04 mms 12_pf 4 hpcb201.1475450.0 op 00:28 36600-70 00:02 mms 14_pf 4 hpcb201.1475450.0 np 00:20 13700-50 00:02 mms 14_pf 4 hpcb201.1475450.0 np 00:20 13700-50 00:02 mms 14_pf 4 hpcb201.1475</pre>	ux moc	enba_fistro	aj_uptraj	_0 16	hncho	301,1474. 204 44750	748.0 P	10 00:1	00 1370	0-30	00:07
mms 08_pf 4 hptb2301.1475055.0 0p 00:26 36600-70 00:21 mms 08_pf 4 hptb2301.1475055.0 0p 00:26 36600-70 00:21 mms 08_pf 4 hptb2301.1475055.0 0p 00:26 36600-70 00:21 mms 06_pf 4 hptb2301.1475054.0 0p 00:26 36600-70 00:24 mms 06_pf 4 hptb2301.1475054.0 0p 00:26 36600-70 00:24 mms 06_pf 4 hptb2301.1475054.0 0p 00:26 36600-70 00:24 mms 06_pf 4 hptb2401.1475451.0 op 00:26 36600-70 00:24 mms model_fc 16 hptb2401.1475452.0 op 00:26 36600-71 00:04 mms 09_pf 4 hptb2401.1475453.0 op 00:26 36600-70 00:04 mms 02_pf 4 hptb2401.1475453.0 op 00:26 36600-70 00:04 mms 12_pf 4 hptb2401.1475453.0 op 00:26 36600-70 00:04 mms 12_pf 4 hptb2301.1475053.0 op 00:26 36600-70 00:02 36600-70 00:02 <t< td=""><td><pre>imbs 01_pf 4 hptD2401.1475455.0 0p 00:26 36600-70 00:21 mos 03_pf 4 hptD2301.4475055.0 0p 00:26 36600-70 00:23 mos 04_pf 4 hptD2301.4475053.0 0p 00:26 36600-70 00:24 imos 06_pf 4 hptD2301.4475053.0 0p 00:26 36600-70 00:24 imos 05_pf 4 hptD2301.4475053.0 0p 00:26 36600-70 00:24 imos 05_pf 4 hptD2401.1475455.0 0p 00:26 36600-70 00:24 imos 06_pf 4 hptD2401.1475451.0 0p 00:26 36600-70 00:24 imos 07_pf 4 hptD2401.1475452.0 0p 00:26 36600-70 00:04 imos 07_pf 4 hptD2401.1475452.0 0p 00:28 36600-70 00:04 imos 09_pf 4 hptD2401.1475452.0 0p 00:28 36600-70 00:04 imos 09_pf 4 hptD2401.1475450.0 0p 00:28 36600-70 00:04 imos 09_pf 4 hptD2401.1475450.0 0p 00:28 36600-70 00:04 imos 12_pf 4 hptD2401.1475450.0 0p 00:28 36600-70 00:04 imos 12_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:04 imos 13_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:04 imos 14_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:04 imos 14_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:04 imos 14_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:02 imos 15_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:02 imos 14_pf 4 hptD2301.14750</pre></td><td>mos</td><td>02_p1</td><td></td><td>4</td><td>hpcb2</td><td>301,1475</td><td>J52.0 0</td><td>/p 00;2</td><td></td><td>0-70</td><td>00:21</td></t<>	<pre>imbs 01_pf 4 hptD2401.1475455.0 0p 00:26 36600-70 00:21 mos 03_pf 4 hptD2301.4475055.0 0p 00:26 36600-70 00:23 mos 04_pf 4 hptD2301.4475053.0 0p 00:26 36600-70 00:24 imos 06_pf 4 hptD2301.4475053.0 0p 00:26 36600-70 00:24 imos 05_pf 4 hptD2301.4475053.0 0p 00:26 36600-70 00:24 imos 05_pf 4 hptD2401.1475455.0 0p 00:26 36600-70 00:24 imos 06_pf 4 hptD2401.1475451.0 0p 00:26 36600-70 00:24 imos 07_pf 4 hptD2401.1475452.0 0p 00:26 36600-70 00:04 imos 07_pf 4 hptD2401.1475452.0 0p 00:28 36600-70 00:04 imos 09_pf 4 hptD2401.1475452.0 0p 00:28 36600-70 00:04 imos 09_pf 4 hptD2401.1475450.0 0p 00:28 36600-70 00:04 imos 09_pf 4 hptD2401.1475450.0 0p 00:28 36600-70 00:04 imos 12_pf 4 hptD2401.1475450.0 0p 00:28 36600-70 00:04 imos 12_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:04 imos 13_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:04 imos 14_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:04 imos 14_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:04 imos 14_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:02 imos 15_pf 4 hptD2301.1475053.0 0p 00:28 36600-70 00:02 imos 14_pf 4 hptD2301.14750</pre>	mos	02_p1		4	hpcb2	301,1475	J52.0 0	/p 00;2		0-70	00:21
mms 03_pf 4 hptb2301.1475055.0 0p 00:26 36600-70 00:21 mms 04_pf 4 hptb2301.1475053.0 0p 00:26 36600-70 00:24 mms 05_pf 4 hptb2301.1475054.0 0p 00:26 36600-70 00:24 mms 06_pf 4 hptb2301.1475054.0 op 00:26 36600-70 00:24 mms 06_pf 4 hptb2301.1475054.0 op 00:26 36600-70 00:24 mms 06_pf 4 hptb2301.1475058.0 op 00:26 36600-70 00:24 mms 07_pf 4 hptb2401.1475452.0 op 00:26 36600-71 00:04 mms 07_pf 4 hptb2401.1475452.0 op 00:26 36600-71 00:04 mms 07_pf 4 hptb2401.1475452.0 op 00:26 36600-70 00:04 mms 07_pf 4 hptb2401.1475453.0 op 00:26 36600-70 00:04 mms 07_pf 4 hptb2401.1475458.0 op 00:26 36600-70 00:04 mms 12_pf 4 hptb2301.1475053.0 op 00:26 36600-70 00:04 mms 12_pf 4 hptb2	mins 08_pf 4 hptD2301.1475455.0 0p 00:26 36600-70 00:21 mins 04_pf 4 hptD2301.1475455.0 0p 00:26 36600-70 00:24 mins 05_pf 4 hptD2301.1475455.0 0p 00:26 36600-70 00:24 mins 06_pf 4 hptD2301.1475457.0 0p 00:26 36600-70 00:24 mins model_fc 36 hptD2401.1475421.0 0p 01:12 36600-70 00:24 mins 00_pf 4 hptD2401.147542.0 0p 01:12 36600-70 00:24 mins 00_pf 4 hptD2401.147542.0 0p 00:26 36600-70 00:24 mins 00_pf 4 hptD2401.147545.0 0p 00:28 36600-70 00:04 mins 00_pf 4 hptD2401.147545.0 0p 00:28 36600-70 00:04 mins 01_pf 4 hptD2401.1475450.0 p 00:28 36600-70 00:04 mins 12_pf 4 hptD2301.1475058.0 p 00:28 36600-70 00:04 mins 12_pf 4 hptD2301.1475058.0 p 00:28 36600-70 00:04 mins 12_pf 4	mos	on_pr		4	hpcb2	401.14754	454.U U	/p 00;2		JU-70	00:21
minus 03_p1 4 hptp2401.1475455.0 0p 00:26 35600-70 00:21 mos 06_pf 4 hptp2201.1475054.0 0p 00:26 36600-70 00:24 mos 05_pf 4 hptp2201.1475054.0 0p 00:26 36600-70 00:24 mos 04_ft 36600-70 00:24 36600-70 00:24 36600-70 00:24 mos 020_ft 4 hptp2201.1475052.0 op 00:24 36600-70 00:24 mos 020_ft 6 hptp2401.1475042.0 op 00:24 36600-70 00:24 mos 07_pf 4 hptp2401.1475042.0 op 00:24 36600-70 00:04 mos 07_pf 4 hptp2401.1475042.0 op 00:24 36600-70 00:04 mos 02_pf 4 hptp2401.1475042.0 op 00:26 36600-70 00:04 mos 12_pf 4 hptp2401.1475042.0 op 00:26 36600-70 00:04 mos 12_pf 4 hptp2401.1475042.0 op 00:26 36600-70 00:02 mos 12_pf 4	mmos 04_pf 4 hptD2401.1475455.0 0p 00:26 36600-70 00:21 mmos 06_pf 4 hptD2301.475053.0 0p 00:26 36600-70 00:23 mmos 05_pf 4 hptD2301.475053.0 0p 00:26 36600-70 00:24 mmos 05_pf 4 hptD2401.1475456.0 0p 00:21 36600-70 00:24 mmos 06_pf 4 hptD2401.1475457.0 0p 00:21 36600-70 00:24 mmos 07_pf 6 hptD2401.1475457.0 op 00:24 36600-70 00:24 mmos 07_pf 4 hptD2401.1475457.0 op 00:24 36600-70 00:04 mmos 07_pf 4 hptD2401.1475457.0 op 00:26 36600-70 00:04 mmos 12_pf 4 hptD2301.1475057.0 op 00:28 36600-70 00:04 mmos 12_pf 4 hptD2301.1475057.0 op 00:28 36600-70 00:04 mmos 12_pf 4 hptD2301.1475057.0 op 00:28 36600-70 00:04 mos 12_pf 4 hptD2301.1475057.0 op 00:28 36600-70 00:02 mos 12_pf 4	mos	08_pi		4	hpc02	301,1475	J55.0 (/p 00;2		07-00	00:21
Instrume OH Intro Solution One and the probability of	mmos 06_pf 4 hptb2301.1475054.00 00.26 36600-70 00.24 mmos 06_pf 4 hptb2301.1475054.00 00.0126 36600-70 00.24 mmos 06_pf 4 hptb2301.1475054.00 00.0126 36600-70 00.24 mos model_fc 36 hptb2401.1475451.00 00.0111 36600-70 00.24 mos model_fc 36 hptb2401.1475452.00 00.0141 36600-71 00.02 muss 07_pf 4 hptb2401.1475452.00 00.026 36600-70 00.04 muss 09_pf 4 hptb2401.1475452.00 00.026 36600-70 00.04 muss 12_pf 4 hptb2301.1475458.00 00.026 36600-70 00.04 muss 12_pf 4 hptb2301.1475058.00 00.026 36600-70 00.026 muss 13_pf 4 hptb2301.1475058.00 00.026 36600-70 00.026 muss 14_pf 4 hptb2301.1475058.00 00.026 36600-70 00.026 muss 12_pf 4 hptb2301.1475057.00 00.026 36600-70 00.026 muss 12_pf 4	mos	os_pr		4	hpcb2	101.1475	+55.U U	p 00;2		0-70	00:21
And Solution And Decomining of the probability of	mos 06_pf 4 hptb2001.1475486.0 0p 00126 36600-70 00124 mos model_fc 36 hptb2001.1475481.0 0p 00112 36600-70 00124 usuad Nodes w.Clock Prio Wait mos control_cf 6 hptb2401.1475482.0 op 00124 36600-70 00104 mos control_cf 6 hptb2401.1475482.0 op 00126 36600-70 00104 mos control_cf 6 hptb2401.1475482.0 op 00126 36600-70 00104 mos 07_pf 4 hptb2401.1475482.0 op 00126 36600-70 00104 mos 12_pf 4 hptb2301.1475058.0 op 00126 36600-70 00104 mos 12_pf 4 hptb2301.1475058.0 op 00126 36600-70 00102 mos 13_pf 4 hptb2301.1475058.0 op 00126 36600-70 00102 mos 14_pf 4 hptb2301.1475058.0 op 00126 36600-70 00102 mos 15_pf 4 hptb2301.1475058.0 op <td>mos</td> <td>oe_pi</td> <td></td> <td>4</td> <td>hpcb2</td> <td>201.14750</td> <td>JS3.0 0 DE4 0 4</td> <td>n 00:2</td> <td></td> <td>0-70</td> <td>00:23</td>	mos	oe_pi		4	hpcb2	201.14750	JS3.0 0 DE4 0 4	n 00:2		0-70	00:23
moss model_fc 36 mpcb2401.1475443.0 00.128 36000700 00:24 var ehf9_model_fc 36 hpcb2401.1475443.0 np *1-day 19700-50 22:52 var ehf9_model_fc 6 hpcb2401.1475462.0 np *1-day 19700-50 22:52 var control_cf 6 hpcb2401.1475457.0 np *1-day 19700-50 22:52 var 03.05 07.07 4 hpcb2401.1475457.0 np *1-day 19700-50 22:52 var 03.05 00.126 36600-70 00:04 36600-70 00:04 imos 03.05 03.05 04 npcb2401.1475463.0 09 00:26 36600-70 00:04 imos 10.07 4 hpcb2301.1475053.0 09 00:26 36600-70 00:02 imos 13.07 4 hpcb2301.1475067.0 09 00:26 36600-70 00:02 imos 14.pf 4 hpcb2301.1475017.0 00 00:21 13700-50	mos uod_lfc 14 hpbb2401.1475421.0 up 001.12 36600-70 001.44 dx ehf9_model_fc 16 hpcb2401.1475421.0 up *1-day 19700-50 22:52 upued Nodes W.Clock Prio Wait mos control_cf 6 hpcb2401.1475482.0 up 001.26 36600-70 001.04 mos control_cf 6 hpcb2401.1475482.0 up 001.26 36600-70 001.04 mos 03_pf 4 hpcb2401.1475482.0 up 001.26 36600-70 001.04 mos 12_pf 4 hpcb2401.1475482.0 up 001.26 36600-70 001.04 mos 12_pf 4 hpcb2401.1475482.0 up 001.26 36600-70 001.04 mos 12_pf 4 hpcb2301.1475058.0 up 001.26 36600-70 001.04 mos 12_pf 4 hpcb2301.1475058.0 up 001.26 36600-70 001.02 mos 15_pf 4 hpcb2301.1475058.0 up 001.26 36600-70 001.02 mos 15_pf 4 hpcb	mos	OF of		4	hpcb2	301,14750 404 44750	J54.0 (150 0 0	n 00:2		0-70	00124
model Sist NpCD2401.1474974.0 np ×1-day Sobol-70 00.40 Wait model Nodes W.Clock Prio Wait moss control_cf 6 hpcb2401.1475452.0 op 00.41 36600-70 00:04 moss 07_pf 4 hpcb2401.1475452.0 op 00:26 36600-70 00:04 moss 10_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 moss 12_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 moss 13_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:04 moss 15_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 moss 15_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 moss 14_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 moss 15_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 moss 14_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 moss 15_pf 4 hpcb2301.1475017.0 np 00:21 13700-50 03:43 moss 14_pf 1 hpcb2301.1475017.0 np 00:17	model_fc 16 hpcb2401.1474974.0 np *1-day 19700-50 22:52 vector 16 hpcb2401.1474974.0 np *1-day 19700-50 22:52 vector 6 hpcb2401.1475452.0 op 00:41 38600-70 00:04 mos 07_pf 4 hpcb2401.1475452.0 op 00:26 38600-70 00:04 mos 07_pf 4 hpcb2401.1475452.0 op 00:26 38600-70 00:04 mos 07_pf 4 hpcb2401.1475452.0 op 00:26 38600-70 00:04 mos 11_pf 4 hpcb2401.1475450.0 op 00:26 38600-70 00:04 mos 12_pf 4 hpcb2301.1475057.0 op 00:26 38600-70 00:02 mos 13_pf 4 hpcb2301.1475057.0 op 00:26 38600-70 00:02 mos 13_pf 4 hpcb2301.1475057.0 op 00:26 38600-70 00:02 mos 14_pf 4 hpcb2301.1475057.0 op 00:26 38600-70 00:02 mos 14_pf 4 hpcb2301.1475057.0 op 00:26 38600-70 00:02 mos 14_pf 4 hpcb2301.1475057.0 op 00:27 19700-50 0	mos	wodel fr			hpcb2	101.1475	124 0 0	n 04:4	12 2000	0-70	00.24
Viewed Nodes Viewed Viewe Viewe Viewe	undet Nodes W.Qlock Prio Wait imos 07_pf 6 hpcb2401.1475452.0 op 00:44 36600-70 00:04 imos 07_pf 4 hpcb2401.1475452.0 op 00:26 36600-70 00:04 imos 03_pf 4 hpcb2401.1475453.0 op 00:26 36600-70 00:04 imos 12_pf 4 hpcb2201.1475053.0 op 00:26 36600-70 00:04 imos 12_pf 4 hpcb2301.1475053.0 op 00:26 36600-70 00:04 imos 12_pf 4 hpcb2301.1475053.0 op 00:26 36600-70 00:02 imos 12_pf 4 hpcb2301.1475053.0 op 00:26 36600-70 00:02 imos 15_pf 4 hpcb2301.147507.0 op 00:26 36600-70 00:02 imos 14_pf 4 hpcb2301.147507.0 op 00:26 36600-70 00:02 imos 14_pf 4 hpcb2301.147507.0 np 00:20 19700-50 03:45 idx ehms_ifstraj_uptraj_0 16 hpcb2301.1474748.0 n	:mus idv	mouer_rc	f.	30	hncho	401.1475	+21.0 C	n xa da	12 3000		00:44
vueed Nodes w.Clock Prio Wait mos control_cf 6 hpcb2401.1475452.0 op 00:41 36600-71 00:04 imos 03_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 imos 03_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 imos 11_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 imos 12_pf 4 hpcb2301.1475059.0 op 00:26 36600-70 00:02 imos 13_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 imos 15_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 imos 17_pf 4 hpcb2401.1475487.0 op 00:26 36600-70 00:02 imos 14_pf 4 hpcb2301.14745487.0 op 00:26 36600-70 00:02 <t< td=""><td>nueued Nodes w.Clock Prio Wait moss control_cf 6 hpcb2401.1475482.0 op 00:41 36600-71 00:04 moss 07_pf 4 hpcb2401.1475452.0 op 00:26 36600-70 00:04 moss 09_pf 4 hpcb2401.1475480.0 op 00:26 36600-70 00:04 moss 11_pf 4 hpcb2401.1475480.0 op 00:26 36600-70 00:04 moss 10_pf 4 hpcb2301.1475059.0 op 00:26 36600-70 00:02 moss 13_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 moss 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 moss 14_pf 4 hpcb2301.147508.0 op 00:26 36600-70 00:02 moss 14_pf 4 hpcb2301.147508.0 op 00:21 19700-50 00:34 moss 14_pf 4 hpcb2301.147508.0 op 00:21 19700-50</td><td>ux</td><td>eni a_mouer_</td><td>_1 C</td><td>16</td><td>npcbz</td><td>+01,1474;</td><td>5/4.0 1</td><td>ip n-ua</td><td>iy 1970</td><td>JU-5U</td><td>22:52</td></t<>	nueued Nodes w.Clock Prio Wait moss control_cf 6 hpcb2401.1475482.0 op 00:41 36600-71 00:04 moss 07_pf 4 hpcb2401.1475452.0 op 00:26 36600-70 00:04 moss 09_pf 4 hpcb2401.1475480.0 op 00:26 36600-70 00:04 moss 11_pf 4 hpcb2401.1475480.0 op 00:26 36600-70 00:04 moss 10_pf 4 hpcb2301.1475059.0 op 00:26 36600-70 00:02 moss 13_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 moss 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 moss 14_pf 4 hpcb2301.147508.0 op 00:26 36600-70 00:02 moss 14_pf 4 hpcb2301.147508.0 op 00:21 19700-50 00:34 moss 14_pf 4 hpcb2301.147508.0 op 00:21 19700-50	ux	eni a_mouer_	_1 C	16	npcbz	+01,1474;	5/4.0 1	ip n-ua	iy 1970	JU-5U	22:52
mos control_cf 6 hpcb2401.1475462.0 op 00:41 36600-71 00:04 mos 07_pf 4 hpcb2401.1475457.0 op 00:26 36600-70 00:04 mos 09_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 mos 11_pf 4 hpcb2401.1475450.0 op 00:26 36600-70 00:04 mos 12_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:04 mos 13_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2401.1475451.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.14754541.0 op 00:26 36600-70 00:27 mos 14_pf 4 hpcb2301.14754541.0 op 00:26 36600-70 00:27 dx ehq4_ifsmin_uptraj_0 16 hpcb2301	<pre>mos control_cf 6 hpcb2401.1475462.0 op 00.41 36600-71 00:04 mos 07_pf 4 hpcb2401.1475457.0 op 00:26 36600-70 00:04 mos 03_pf 4 hpcb2401.1475483.0 op 00:26 36600-70 00:04 mos 11_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 mos 12_pf 4 hpcb2301.1475053.0 op 00:26 36600-70 00:02 mos 13_pf 4 hpcb2301.1475053.0 op 00:26 36600-70 00:02 mos 13_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475468.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475468.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475468.0 op 00:28 36600-70 00:02 mos 17_pf 4 hpcb2301.1475467.0 op 00:28 36600-70 00:02 mos 17_pf 4 hpcb2301.1475467.0 op 00:28 36600-70 00:02 mos 17_pf 5 10 16 hpcb2301.14745487.0 np 00:21 19700-50 03:45 dx ehg2_ifstraj_uptraj_0 16 hpcb2301.147487.0 np 00:21 19700-50 03:45 dx ehg8_ifstraj_uptraj_0 16 hpcb2301.147487.0 np 00:21 19700-50 03:45 dx ehg8_ifstraj_uptraj_0 16 hpcb2301.147487.0 np 00:21 19700-50 03:45 dx ehg8_ifstraj_uptraj_0 16 hpcb2301.1474950.0 np 00:31 19700-50 03:45 dx ehg8_ifstraj_uptraj_0 16 hpcb2301.1474789.0 np 01:17 19700-50 00:28 dx ehf8_model_fc 16 hpcb2401.1475489.0 np 02:13 19700-50 00:28 dx ehf8_model_fc 16 hpcb2401.1475489.0 np 02:13 19700-50 00:28 dx ehhx_model_fc 16 hpcb2401.1475489.0 np 02:13 19700-50 00:28 dx ehhx_model_fc 16 hpcb2401.1475089.0 nf 02:13 19700-50 00:28 dx ehhx_model_fc 16 hpcb2401.1475089.0 nf 02:13 19700-50 00:28 dx ehhx_model_fc 16 hpcb2401.14775089.0 16 16 10:17 ehba_ifstraj_uptraj_0 hpcb2301.1474748.0 16 0 10:21 Chpcb2501{/home/cos/newops/bin}:935</pre>	ueued			Node	25			W.Clo	ock Pr	rio	Wait
mos 07_pf 4 hpcb2401.1475457.0 op 00:26 38600-70 00:04 mos 09_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 mos 12_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 mos 10_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 mos 13_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 mos 13_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2401.1475070.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475070.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2301.14750761.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475071.0 np 00:21 19700-50 03:45 dx ehg_inde_ifc 16 hpcb2301.14747850.0 np 00:17 19700-50 03:30 dx ehg_inde_ifc	<pre>mos 07_pf 4 hpcb2401.1475457.0 op 00:26 36600-70 00:04 mos 09_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 mos 11_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 mos 12_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.147507.0 np 00:21 19700-70 00:27 mos 14_pf 4 hpcb2301.1475048.0 op 00:28 36600-69 00:04 dx ehq4_ifsmin_uptraj_0 16 hpcb2301.1474487.0 np 00:21 19700-50 03:42 dx ehg2_ifstraj_uptraj_0 16 hpcb2301.1474487.0 np 00:21 19700-50 03:30 dx ehj8_model_fc 16 hpcb2301.1474488.0 np 01:55 19700-50 03:13 dx ehj8_model_fc 16 hpcb2301.1474789.0 np 01:55 19700-50 03:13 dx ehh8_model_fc 16 hpcb2301.1474789.0 np 02:13 19700-50 03:13 dx ehh8_model_fc 16 hpcb2401.1475483.0 np 02:13 19700-50 00:28 dx ehhx_model_fc 16 hpcb2401.1474748.0 16 0 10:21 dx ehpc2501{/home/cos/newops/bin}:935</pre>	mos	control cf		6	hpcb2	401.1475	1 62.0 d	p 00:4	11 3660	00-71	00:04
mos 09_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 mos 11_pf 4 hpcb2401.1475450.0 op 00:26 36600-70 00:04 mos 10_pf 4 hpcb2301.1475053.0 op 00:26 36600-70 00:02 mos 10_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 mos 13_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475067.0 np 00:21 19700-70 00:27 dx ehg2_ifstraj_uptraj_0 16 hpcb2301.14747468.0 np 00:17 19700-50 03:45 dx ehg2_model_fc 16 hpcb2301.1474748.0 np 00:03 19700-50 03:13 dx ehg	mos 09_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 mos 11_pf 4 hpcb2401.1475458.0 op 00:26 36600-70 00:04 mos 12_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 mos 12_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 mos 13_pf 4 hpcb2301.1475059.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475068.0 op 00:21 19700-70 00:21 dx ehq4_ifsmin_uptraj_0 16 hpcb2301.1475087.0 np 00:21 19700-70 00:23 dx ehg4_ifstraj_uptraj_0 16 hpcb2301.147478.0 np 00:21 19700-70 00:23 14700-70 03:13 dx ehgk_ifstraj_uptraj_0 12 hpcb2301.1474789.0 np 00:03 19700-50 03:13 dx ehgk_ifstraj_uptraj_0 12 hp	mos	07_pf		4	hpcb2	401.1475	457.0 d	p 00:2	26 3660	00-70	00:04
mos 11_pf 4 hpcb2401.1475460.0 op 00:26 36600-70 00:04 mos 12_pf 4 hpcb2301.1475053.0 op 00:26 36600-70 00:04 mos 13_pf 4 hpcb2301.1475057.0 op 00:26 36600-70 00:02 mos 13_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475067.0 op 00:21 13700-70 00:21 mos 18_pf 16 hpcb2301.1475461.0 op 00:21 13700-50 03:30 mox ehq4_ifsmin_uptraj_0 16 hpcb2301.1474788.0 onp 01:55 19700-50 03:30 mox ehg4_ifstraj_uptraj_0 12 hpcb2301.14747893.0 np 01:55 19700-50 00:02 mox ehf9_mo	<pre>mos 11_pf 4 hpcb2401.1475460.0 op 00:26 36600-70 00:04 mos 12_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 mos 13_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2401.1475461.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475461.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475461.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475461.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1477468.0 op 00:27 19700-50 03:45 mos 17_pf 5 19700-50 03:45 mos 17_pf 5 19700-50 03:45 mos 17_pf 1 16 hpcb2301.1474748.0 np 00:17 19700-50 03:30 mos 17_pf 1 16 hpcb2301.1474748.0 np 00:17 19700-50 03:45 mos 14x ehgs_model_fc 16 hpcb2301.1474748.0 np 00:03 19700-50 03:13 mos 14x ehf9_model_fc 16 hpcb2401.1475469.0 np 02:03 19700-50 02:23 mos 14x ehhs_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 mos 16 hpcb2501{/hom/cos/newops/bin}:935 mos 16 hpcb2301.1474748.0 16 0 10:21 mos 16 hpcb2501{/hom/cos/newops/bin}:935 mos 16 hpcb2301.14774748.0 16 0 10:21 mos 16 hpcb2501{/hom/cos/newops/bin}:935 mos 16 hpcb2301.1475469.0 16 hpcb2301.1475469.0 16 hpcb2301</pre>	mos	09 pf		4	hbcb2	401.1475	458.O d	00:2	26 3660	00-70	00:04
mos 12_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 mos 13_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:02 mos 13_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2401.1475068.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475017.0 np 00:21 19700-70 00:27 mos 17_pf 4 hpcb2301.1474687.0 np 00:20 19700-50 03:45 mdx ehg4_ifsmin_uptraj_0 16 hpcb2301.14747488.0 np 00:17 19700-50 03:45 mdx ehgk_ifstraj_uptraj_0 16 hpcb2301.1474748.0 np 01:17 19700-50 03:45 mdx ehf9_model_fc 16 hpcb2401.1474985.0 np 00:03 19700-50 00:02 mdx	<pre>mos 12_pf 4 hpcb2301.1475058.0 op 00:26 36600-70 00:04 mos 10_pf 4 hpcb2301.1475057.0 op 00:28 36600-70 00:02 mos 13_pf 4 hpcb2301.1475057.0 op 00:28 36600-70 00:02 mos 14_pf 4 hpcb2301.1475088.0 op 00:28 36600-70 00:02 mos 14_pf 4 hpcb2301.1475088.0 op 00:28 36600-70 00:02 mos 17_pf 4 hpcb2301.1475087.0 op 00:28 36600-70 00:02 mos 17_pf 4 hpcb2301.1475087.0 np 00:21 19700-70 00:27 dx ehq4_ifsmin_uptraj_0 16 hpcb2301.1475070.0 np 00:21 19700-50 03:45 dx ehg4_ifstraj_uptraj_0 16 hpcb2301.1474887.0 np 00:20 19700-50 03:30 dx ehpk_ifstraj_uptraj_0 16 hpcb2301.1474789.0 np 00:17 19700-50 03:30 dx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474789.0 np 00:03 19700-50 03:19 dx ehf9_model_fc 16 hpcb2401.1475449.0 np 01:55 19700-50 03:13 dx ehpk_ifstraj_uptraj_0 12 hpcb2301.147489.0 np 02:13 19700-50 00:28 dx ehhx_model_fc 16 hpcb2401.1475469.0 np</pre>	mos	11_pf		4	hpcb2	401.1475	460.0 d	p 00;2	26 3660	00-70	00:04
mos 10_pf 4 hpcb2301.1475059.0 op 00:26 36600-70 00:04 mos 13_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475070.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475070.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475070.0 op 00:21 19700-70 00:24 dx ehq4_ifsmin_uptraj_0 16 hpcb2301.14747807.0 np 00:20 19700-50 03:45 dx ehg2_ifstraj_uptraj_0 16 hpcb2301.1474780.0 np 01:17 19700-50 03:43 dx ehg2_ifstraj_uptraj_0 12 hpcb2301.1474789.0 np 01:15 19700-50 03:13 dx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474798.0 np 01:03 19700-50 00:28 dx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474798.0 np 01:03 19700-50 00:02<	imos 10_pf 4 hpcb2301.1475059.0 op 00:26 36600-70 00:04 imos 13_pf 4 hpcb2301.1475070.0 op 00:26 36600-70 00:02 imos 14_pf 4 hpcb2301.1475070.0 op 00:26 36600-70 00:02 imos 14_pf 4 hpcb2301.1475070.0 op 00:26 36600-70 00:02 imos 17_pf 4 hpcb2301.1475070.0 op 00:26 36600-70 00:02 imos 17_pf 4 hpcb2301.14747867.0 op 00:21 19700-70 00:02 idx ehq4_ifsmin_uptraj_0 16 hpcb2301.14747867.0 np 00:21 19700-50 03:45 idx ehg2_model_fc 16 hpcb2301.14747867.0 np 00:02 19700-50 03:45 idx ehg2_model_fc 16 hpcb2301.147474867.0 np 00:03 19700-50 03:13 idx ehf3_model_fc 16 hpcb2401.1475419.0 np 11.44y 19700-50 00:02 idx ehf3_model_fc 16 hpcb2401.1475463.0 np 02:13 19700-50 00:02	mos	12_pf		4	hpcb2	301.1475	058.0 d	p 00:2	26 3660	00-70	00:04
mos 13_pf 4 hpcb2301.1475067.0 op 00:26 36600-70 00:02 mos 15_pf 4 hpcb2401.1475470.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2301.1475048.0 op 00:26 36600-70 00:02 rdx ehq4_ifsmin_uptraj_0 16 hpcb2301.14750481.0 op 00:21 19700-70 00:27 rdx ehq4_ifsmin_uptraj_0 16 hpcb2301.14775087.0 np 00:20 19700-50 03:45 rdx ehg8_ifstraj_uptraj_0 16 hpcb2301.14774687.0 np 00:17 19700-50 03:30 rdx ehg8_ifstraj_uptraj_0 16 hpcb2301.1477498.0 np 01:15 19700-50 03:13 rdx ehf8_model_fc 16 hpcb2401.1475481.0 np 01:15 19700-50 01:15 rdx ehf9_model_fc 16 hpcb2401.1475498.0 np 02:13 19700-50 00:02 rdx ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50	<pre>prevations 116 nodes reserved at 10:05 for use at 10:09 Free Need the hpcb2401.1475463.0 np 00:26 36600-70 00:02 36600-70 00:02 36700-50 00:02 36700-50 00:02 36700-50 00:03 37700-50 00:03 37700-50 00:02 36700-50 00:02 367000-50 00:02 36700-50 00:02 367000-50 00:02 36700-50 00:02 37000-50 00:02 37000</pre>	mos	10_pf		4	hpcb2:	301.1475)59.O d)p 00:2	26 3660	00-70	00:04
mos 15_pf 4 hpcb2401.1475470.0 op 00:26 36600-70 00:02 mos 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2401.1475068.0 op 00:26 36600-69 00:04 imos 17_pf 4 hpcb2301.1475017.0 np 00:21 19700-50 03:45 idx ehq4_ifstraj_uptraj_0 16 hpcb2301.1474687.0 np 00:02 19700-50 03:45 idx eh8z_ifstraj_uptraj_0 16 hpcb2301.1474789.0 np 00:17 19700-50 03:45 idx ehg8_model_fc 16 hpcb2301.1474789.0 np 00:03 19700-50 03:13 idx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474789.0 np 01:05 19700-50 03:13 idx ehpk_ifstraj_uptraj_0 12 hpcb2301.1475481.0 np 01:05 19700-50 00:02 idx ehpk_ifstraj_uptraj_0 16 hpcb2301.1474789.0 np 02:13 19700-50 00:02 idx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-	<pre>mos 15_pf</pre>	mos	13_pf		4	hpcb2	301.1475	067.0 d	b 00:2	26 3660	00-70	00:02
mos 14_pf 4 hpcb2301.1475068.0 op 00:26 36600-70 00:02 mos 17_pf 4 hpcb2401.1475047.0 op 00:26 36600-69 00:04 rdx ehq4_ifsmin_uptraj_0 16 hpcb2301.1475017.0 np 00:21 19700-70 00:27 rdx ehmv_ifstraj_uptraj_0 16 hpcb2301.14747607.0 np 00:17 19700-50 03:45 rdx ehg2_ifstraj_uptraj_0 16 hpcb2301.1474746.0 np 00:17 19700-50 03:45 rdx ehj8_model_fc 18 hpcb2301.1474746.0 np 00:17 19700-50 03:13 rdx ehj8_model_fc 18 hpcb2301.1474789.0 np 01:55 19700-50 03:13 rdx ehf9_model_fc 16 hpcb2401.1474789.0 np *1-day 19700-50 00:28 rdx ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 rdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02	<pre>mos 14_pf</pre>	mos	15_pf		4	hpcb2	401.14754	470.0 d)b 00:2	26 3660	00-70	00:02
pmos 17_pf 4 hpcb2401.1475461.0 op 00:26 36600-69 00:04 dx ehq4_ifsmin_uptraj_0 16 hpcb2301.1475017.0 np 00:20 19700-70 00:27 dx ehmv_ifstraj_uptraj_0 16 hpcb2301.1474687.0 np 00:20 19700-50 03:45 dx ehg2_ifstraj_uptraj_0 16 hpcb2301.1474687.0 np 00:17 19700-50 03:45 dx ehg8_ifstraj_uptraj_0 16 hpcb2301.1474789.0 np 01:15 19700-50 03:45 dx ehj8_model_fc 16 hpcb2301.1474789.0 np 01:15 19700-50 03:45 dx ehf9_model_fc 16 hpcb2301.1474789.0 np 01:55 19700-50 03:43 dx ehf9_model_fc 16 hpcb2301.1474995.0 np 00:03 19700-50 00:28 dx ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehhx_model_fc 16 hpcb2401.14774974.0 16 2	<pre>imos 17_pf</pre>	mos	14_pf		4	hpcb2	301.1475(068.O d	jp 00:2	26 3660	00-70	00:02
dx ehq4_ifsmin_uptraj_0 16 hpcb2301.1475017.0 np 00:21 19700-70 00:27 dx ehmw_ifstraj_uptraj_0 16 hpcb2301.1474887.0 np 00:20 19700-50 03:45 dx eh8z_ifstraj_uptraj_0 16 hpcb2301.1474746.0 np 00:17 19700-50 03:45 dx eh8z_ifstraj_uptraj_0 16 hpcb2301.1474746.0 np 00:17 19700-50 03:30 dx ehge_ifstraj_uptraj_0 12 hpcb2301.1474746.0 np 00:03 19700-50 03:30 dx ehpk_ifstraj_uptraj_0 12 hpcb2301.14747499.0 np 00:03 19700-50 03:30 dx ehf9_model_fc 16 hpcb2301.14747499.0 np 01:55 19700-50 00:28 dx ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'dx ehhx_model_fc 16 hpcb2401.1475469.0 np	<pre>rdx ehq4_ifsmin_uptraj_0 16 hpcb2301.1475017.0 np 00:21 19700-70 00:27 rdx ehmu_ifstraj_uptraj_0 16 hpcb2301.1474687.0 np 00:20 19700-50 03:45 rdx ehg2_ifstraj_uptraj_0 16 hpcb2301.1474687.0 np 00:17 19700-50 03:30 rdx ehj8_model_fc 16 hpcb2301.147498.0 np 01:55 19700-50 03:13 rdx ehf9_model_fc 16 hpcb2301.147499.0 np 01:55 19700-50 01:15 rdx ehf9_model_fc 16 hpcb2301.147499.0 np 01:03 19700-50 01:15 rdx ehf9_model_fc 16 hpcb2401.1475419.0 np *1-day 19700-50 00:28 rdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 rdx ehhy_model_fc 16 rdx rdx rdx rdx rdx rdx rdx rdx rdx rdx</pre>	mos	17_pf		4	hpcb2	401.14754	461.O C	p 00:2	26 3660	00-69	00:04
dx ehmv_ifstraj_uptraj_0 16 hpcb2301.1474687.0 np 00:20 19700-50 03:45 dx ehsz_ifstraj_uptraj_0 16 hpcb2301.1474789.0 np 00:17 19700-50 03:30 dx ehj8_model_fc 16 hpcb2301.1474789.0 np 01:55 19700-50 03:13 dx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474789.0 np 00:03 19700-50 01:15 dx ehf9_model_fc 16 hpcb2301.1474789.0 np 01:55 19700-50 01:15 dx ehf9_model_fc 16 hpcb2401.1474595.0 np 00:03 19700-50 00:02 dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 dx ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 dx ehf9_model_fc 16 hpcb2401.147474749.0 16 32	dx ehmv_ifstraj_uptraj_0 16 hpcb2301.1474687.0 np 00:20 19700-50 03:45 dx ehsz_ifstraj_uptraj_0 16 hpcb2301.1474746.0 np 00:17 19700-50 03:30 dx ehg_model_fc 16 hpcb2301.1474798.0 np 01:55 19700-50 03:13 dx ehf9_model_fc 16 hpcb2301.1474998.0 np 01:03 19700-50 01:15 dx ehf9_model_fc 16 hpcb2301.1474998.0 np 01:03 19700-50 01:15 dx ehf9_model_fc 16 hpcb2401.1475419.0 np *1-day 19700-50 00:28 dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'hxpst ehf9_model_fc hpcb2401.14774974.0 16 32 10:	•dx	ehq4_ifsmi	n_uptraj_	0 16	hpcb2	301.1475	017.O r	np 00:2	21 1970	00-70	00:27
dx eh8z_ifstraj_uptraj_0 16 hpcb2301.1474746.0 np 00:17 19700-50 03:30 dx ehj8_model_fc 16 hpcb2301.1474748.0 np 01:55 19700-50 03:13 dx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474748.0 np 00:03 19700-50 01:15 dx ehf9_model_fc 16 hpcb2301.1474995.0 np 00:03 19700-50 01:15 dx ehf9_model_fc 16 hpcb2401.1475419.0 np *1-day 19700-50 00:02 var ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 var ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 var ehf9_model_fc hpcb2401.14774974.0 16 32 10:0	dx eh8z_ifstraj_uptraj_0 16 hpcb2301.1474746.0 np 00:17 19700-50 03:30 dx ehj8_model_fc 16 hpcb2301.1474789.0 np 01:55 19700-50 03:13 dx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474995.0 np 00:03 19700-50 01:15 dx ehf9_model_fc 16 hpcb2301.1474995.0 np 00:03 19700-50 00:28 dx ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 var ehhx_model_fc hpcb2401.1475469.0 np 02:13 19700-50 00:02 var ehf9_model_fc hpcb2401.1475469.0 np 2 48 var ehf9_model_fc hpcb2401.1474974.0 16 32 10:06 var ehf9_model_fc	'dx	ehmv_ifstr	aj_uptraj	_0 16	hpcb2	301.14746	387.0 r	ip 00:2	20 1970	00-50	03:45
dx ehj8_model_fc 16 hpcb2301.1474789.0 np 01:55 19700-50 03:13 vdx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474995.0 np 00:03 19700-50 01:15 vdx ehf9_model_fc 16 hpcb2301.1474995.0 np 01:03 19700-50 00:28 vdx ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehf9_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehfy_model_fc hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehfy_model_fc hpcb2401.14747974.0 np 16 32 10:06 vdx ehfy_model_fc hpcb2401.14749748.0 np 16 10:17 vdx ehfy_mode	<pre>vdx ehj8_model_fc 16 hpcb2301.1474783.0 np 01:55 19700-50 03:13 vdx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474935.0 np 00:03 19700-50 01:15 vdx ehf9_model_fc 16 hpcb2401.1475413.0 np *1-day 19700-50 00:28 vdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehhx_model_fc 16 hpcb2401.1474974.0 16 12 11 vdx ehf9_model_fc 16 hpcb2401.1474974.0 16 10:17 ehba_ifstraj_uptraj_0 hpcb2301.1474748.0 16 0 10:21 vdpcb2501{/home/cos/newops/bin}:99\$ </pre>	'dx 👘	eh8z_ifstr	aj_uptraj	_0 16	hpcb2	301.1474)	746.0 r	np 00:1	17 1970	00-50	03:30
vdx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474395.0 np 00:03 19700-50 01:15 vdx ehf9_model_fc 16 hpcb2401.1475449.0 np *1-day 19700-50 00:28 vdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehhx_model_fc hpcb2401.1475469.0 np 02:13 19700-50 00:02 vdx ehhx_model_fc 10:02 48 10:02 48 vdx ehf9_model_fc hpcb2401.1474974.0 16 10:105 vdx ehf1_ifstraj_uptraj_0 hpcb2201.14749748.0 16 10:17 ehbx_ifstrai_uptrai_0 hpcb2201.1474748.0 16	<pre>vdx ehpk_ifstraj_uptraj_0 12 hpcb2301.1474995.0 np 00:03 19700-50 01:15 'dx ehf9_model_fc 16 hpcb2401.1475449.0 np *1-day 19700-50 00:28 'dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 'no p use 66 2 48 'hkpt ehf9_model_fc hpcb2401.1474974.0 16 32 10:06 'ou decide ehht_ifstraj_uptraj_0 hpcb2401.1475089.0 16 16 10:17 ehba_ifstraj_uptraj_0 hpcb2301.1474748.0 16 0 10:21 'Chpcb2501{/home/cos/newops/bin}:99\$ ■</pre>	'dx 👘	ehj8_model.	fc	16	hpcb2	301.1474)	789.0 r	np 01:5	55 1970	00-50	03:13
rdx ehf9_model_fc 16 hpcb2401.1475419.0 np *1-day 19700-50 00:28 rdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 perations 116 nodes reserved at 10:05 for use at 10:09 Free Need in op use 66 2 48 hkpt ehf9_model_fc hpcb2401.1474974.0 16 32 10:06 rou decide ehht_ifstraj_uptraj_0 hpcb201.1474974.0 16 0 10:17 ehba ifstraj uptraj 0 hpcb201.1474974.0 16 0 10:21	<pre>rdx ehf9_model_fc 16 hpcb2401.1475419.0 np *1-day 19700-50 00:28 rdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 rdx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 rdx ehhx_model_fc 16 10:05 for use at 10:09 Free Need</pre>	"dx 👘	ehpk_ifstr	aj_uptraj	_0 12	hpcb2	301.1474	995.O r	np 00:0	03 1970	00-50	01:15
<pre>vperations 116 nodes reserved at 10:05 for use at 10:09 Free Need in op use 66 2 48 :hkpt ehf9_model_fc hpcb2401.1474974.0 16 32 10:06 :ou decide ehht_ifstraj_uptraj_0 hpcb2401.14749748.0 16 0 10:21</pre>	<pre>>>dx ehhx_model_fc 16 hpcb2401.1475469.0 np 02:13 19700-50 00:02 >>>perations 116 nodes reserved at 10:05 for use at 10:09 Free Need in op use 66 2 48 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>	"dx	ehf9_model.	_fc	16	hpcb2	401.14754	419.O r	np ×1−da	ay 1970	00-50	00:28
operations 116 nodes reserved at 10:05 for use at 10:09 Free Need in op use 66 2 48 hkpt ehf9_model_fc hpcb2401.1474974.0 16 32 10:06 You decide ehht_ifstraj_uptraj_0 hpcb2401.1475089.0 16 16 10:17 ehba ifstrai uptraj 0 hpcb2301.1474748.0 16 0 10:21	<pre>>perations 116 nodes reserved at 10:05 for use at 10:09 Free Need n op use 66 2 48 hkpt ehf9_model_fc hpcb2401.1474974.0 16 32 10:06 ou decide ehht_ifstraj_uptraj_0 hpcb2401.1475089.0 16 16 10:17 ehba_ifstraj_uptraj_0 hpcb2301.1474748.0 16 0 10:21 <pre>>chpcb2501{/home/cos/newops/bin}:99\$ ■</pre></pre>	"dx 👘	ehhx_model.	_fc	16	hpcb2	401.1475/	469.0 r	np 02:1	13 1970	00-50	00:02
<pre>vchpcb2501{/home/cos/newops/bin}:99\$ </pre>)perat In op Lhkpt /ou de VChpcb	ions 116 use 66 ehf9_1 cide ehht_ 2501{/home,	nodes res model_fc ifstraj_u ifstraj_u /cos/newo	erved at ^ ptraj_0 ptraj_0 ps/bin}:98	10:05 fr hpcb: hpcb: hpcb: 3≸ ∎	or use at 2401.1474 2401.1479 2301.1479	t 10:09 1974.0 5089.0 1748.0	9 Fre 1 1 1	e Need 2 48 16 32 16 16 16 0	10:06 10:17 10:21	3 7

Figure 9 - Il_jobs during an operational period. FloatingCpus = 8, so additional information is provided particularly as resources will not be released at the time specified.

Please note the situation shown in Figure 9 was created artificially. Resource Reservation at 09:50 was delayed until 10:05. Il_jobs shows the time is 10:13:34, 116 nodes reserved, where 10:09 is the time the nodes were needed operationally. Unusually 48 nodes remain in use by non-operational jobs. One decision is easy. Job hpcb2401.1474974.0 (ehf9_model_fc) can be check-pointed and rerun. The other 2 jobs will complete soon, one at 10:17 (in 4 minutes time) and the other at 10:21 (in 8 minutes time), so waiting a few minutes for these 2 jobs to end will not cause a problem, both being well within the 15-minute flexibility.

18. What has been achieved so far?

Has the variability in the start-times and end-times of the operational runs been reduced? Do the daily runs of the suite complete within 15 minutes of the optimum time?

The variability of the start-time of the operational runs, before FloatingCpus were reserved, is shown in Figure 10 and the variability of the start-time of the operational run after FloatingCpus were reserved is shown in Figure 11.



Figure 10 - the variability in the start-time of operational runs, before Resource Reservation was introduced.



Figure 11 - the variability in the start-time of operational runs before Resource Reservation was introduced.

Comparing Figures 10 and 11 you can see the reservation scheme has significantly reduced the variability in the scheduling of the first operational job. Only once in May 2004 was more than 5 minutes of the flexibility used.

In Figures 12 and 13 you can see the variability of the end-times of the operational runs before FloatingCpus were reserved and after FloatingCpus were reserved.



Figure 12 - the variability in the end-times of operational jobs, before Resource Reservation was introduced.



Figure 13 - the variability in the end-time of operational jobs, after Resource Reservation was introduced.

Comparing figures 12 and 13 you can see the variability in the end-times of all operational jobs have been reduced too.

Note – since October 2003 significant improvements have been made to the efficiency of operational jobs and they run faster than before. As a result, the optimal end time of 23:00 achieved in October 2003 has been replaced by an end-time of 22:24 in May 2004. This means the need to keep operational jobs to a 15-minute variation is less acute at present, but not for long. Plans for higher quality forecasts (with greater computational needs) will soon push the end-time back to 23:00.

So clearly the variation exceeds 15 minutes and more work is needed to limit the variation to 15 minutes. 2 solutions are already planned, to optimise the pre-selection of jobs and to ensure when human intervention is needed that action is taken quickly.

Finally Figure 14 shows the allocation of CPUs to all jobs on a typical day since Resource Reservation was introduced.



Figure 14 - the allocation of CPU's to all jobs on a typical day since Resource Reservation was introduced, plotted over 12-minute intervals, where the red line (936 CPUs) is the maximum number that parallel jobs can use.

Comparing Figure 14 with Figure 3 it is clear the allocation of CPU's to all jobs since Resource Reservation was introduced has reduced by a little over 1%. This is greater than was hoped for. But as mentioned earlier there are plans to optimise the selection of jobs before operational periods, which is when the greatest waste occurs. So importantly not only will optimising job selection minimise runtime variation it can be made to increase node utilisation too.

19 The conclusion.

The reservation scheme has been very effective in keeping the start-time to a tighter schedule but less so with the end-time. In addition there is a reduction to the overall system utilisation of just over 1% as seen when comparing Figure 3 and Figure 14. But ways have been described to both minimise the variation and maximise the allocation of nodes and work will continue to achieve this.

20 Main points.

It is essential that parallel jobs can be scheduled independent of other systems, that the jobs are designed well (suit the system) and system performance (GPFS, I/O nodes, Loadleveler, network, paging) does not vary. Thus (using historical data) predicted run-times will be accurate.

The Loadleveler feature FloatingCpus enables resources to be controlled globally, dynamically and logically, independent of job class where jobs that request FloatingCpus are members of one logical class and jobs that do not request FloatingCpus are members of another logical class. Managing the resources to suit these 2 logical classes could not be easier.

Setting values of FloatingCpus a fixed time in advance based on averages is very straightforward and reasonably effective but the variation in runtimes and better system utilisation could be obtained by optimising job selection prior to operational periods.

Reducing the value of FloatingCpus without turning classes off means that the underlying scheduling of nonoperational jobs continues as normal in the reduced set of nodes, and there is complete flexibility as to which nodes are used by operational jobs and which nodes are used by non-operational jobs. Having the system set wall_clock_limit with offset 1+ overcomes the kill on wall_clock_limit exceeded. Operational jobs have their wall_clock_limit set automatically too which means operators need not remember how long the multitude of operational jobs should take to run and any over-run is flagged immediately.

It is essential that monitoring tools (in our case operator displays) are created so that it is possible to confirm both backfill and scheduling are performing as expected or when events do not go to plan a mechanism (again operator displays) is in place to make operators, administrators and analysts aware of the problem(s).

There is no need to have lots of user batch classes (we had many NQS queues on previous systems) for jobs that are short, long, slow, fast, big, small etc so that in pre-operational periods the right jobs can be selected by draining classes, reducing job-limits per class, per user etc.

21 Other thoughts.

The ability to schedule jobs as described stems from the understanding that the system used is a High Performance Cluster (HPC) System managed as a Super-Computer and is not a configuration of loosely coupled Server systems. All those involved in providing elements of the service on the HPC system; analysts, support staff, users, administrators, managers, have a fundamental desire for the system to be configured optimally and for jobs to run efficiently. I cannot stress enough that jobs must not use (need) swap space. The processes in each node of a multi-node, high performance, well parallelised, cpu-bound job, must not exceed physical memory as it only takes 1 process to use too much memory and all nodes will perform badly as the rogue process swaps.

Ideally jobs will parallelise well, scale reasonable well and use a whole node or multiples of nodes. Memory sharing is only permitted if job performance and known-runtimes are not altered. If users have CPU-bound jobs that use less than 1 node, particularly serial jobs with 1 process, the users are advised to run the jobs on Server systems.

Jobs that perform badly must be detected and users must know to call support staff for help (or eventually get caught) if they realise new jobs (for which there is no known run-time data) perform badly. Users and more importantly application programmers, who make it possible for suites of jobs (in the form of an experiment) to be submitted automatically, must be willing to help sort out job related problems and identify solutions.

As the release of all nodes for use by operational jobs occurs twice a day, all user jobs have to finish within 1 hour. Jobs that run for more than 1 hour should either create restart files so that they can be killed without substantial losses, or should have the ability to trap a signal and then create restart files before killing themselves. In Figure 9 such a job was labelled Chkpt. This job will accept and trap such a signal, write a restart file and kill itself.

There is no room for sloppy philosophy. The idea that it is OK for a job to be inefficient and run for a long time as long as the user pays for it, is just not acceptable.