Management Experiences of a Cray T3D Computer in an Academic Environment

Jean-Michel Chenais, Service Informatique Central Ecole Polytechnique Fédérale de Lausanne

ABSTRACT: *The EPFL has been using Cray Unicos operating systems since the arrival of a Cray 2 in September 1988. Currently, the EPFL computing centre offers high performance services with a YMP/M94 PVP server running Unicos 8.x, and a T3D running the Max 1.2.x operating system.*

This paper will present some of the major administrative tasks needed to support a broad variety of users working with the currently available resources. This text will develop selected characteristics of the system, such as the Y-MP and T3D configurations, some local scheduling mechanisms, production flow control and production optimization. We will conclude with some future trends brought about by the availability of new features provided by Cray Research.

Background

Our first Cray/Unicos system was a Cray-2 machine, installed in September 1988. It offered 43000 hours of service over a period of more than 5 years. This period was devoted to exploration and discoveries of the UNIX world, together with the evaluation of the efficiency of the administration features of the Unicos OS for large Cray Research machines.

As the first massively parallel systems were beginning to come out, EPFL was prospecting for solutions in this field. In September 1993 the EPFL and Cray Research signed a Parallel Application Technology Program (PATP) research cooperation contract.

Consequently, our Y-MP was upgraded to a Y-MP/M94. By the end of the same year, the Cray-2 stopped its production, and most of its load was redirected to the new Y-MP machine.

About 6 months before the arrival of the T3D, the T3D emulator was installed on the new Y-MP, and a number of courses and training periods were organized by the PATP support group to prepare people for massively parallel programming techniques.

We received the T3D machine, with 128 PEs of 2 Mwords each, by the end of May 1994. It started production on June 2nd, and immediately the PATP group was able to run application programs. First results of this work were presented at the fall 1994 CUG, in Tours (France), by a number of PATP members. A few months later, all the hardware boards where changed so that the T3D, in a single cabinet model, was configured with 256 PEs with 8 Megawords each. Since then the distributed memory size (16 Gigabytes) is four times that of the Y-MP (4 Gigabytes).

The Academic environment

The manner in which the Y-MP and the T3D were installed had the following major consequences : a fairly large number of users with different needs, skills and goals, wanted to work independently on the Y-MP only, or on both machines.

The Y-MP acts as a CPU server for its own group of users, and also as a front-end for the T3D users. Users are from the EPFL research centres; however, they may also be from external groups, for example other universities or private companies.

The T3D plays a special role : its primary goal is to provide resources to the PATP projects with highest priority. A development and production environment must be provided to all PATP members. Several EPFL computer science departments, institutes and external universities, have independently expressed their interest for T3D projects. A numbers of various users are thus working with the T3D, making connections from Spain, France, Italy, Germany, Netherlands and Great Britain.

The Y-MP set of users contains about 600 accounts, of which about 100 are active, whereas the T3D has about 130, of which, including the PATP accounts, 30 are accessed daily.

This is to show that the Y-MP and the T3D are heavily loaded, and managing both machines requires specific attention so that services assigned to each one can be fullfilled.

Administrative and Software configuration

The general idea was to create two distinct groups of users : the first contains those people allowed to run programs on the Y-MP server only, the second group contains those users allowed to work on the T3D and the Y-MP.

The hardware configuration was organized to allow this scheme to be easily implemented. Disks, channels and other peripherals are linked to the Y-MP in such a way that most of the pure T3D I/O streams are independant from those of the Y-MP. Each machine has its own set of home directories and scratch areas. This method clarifies administrative tasks, improves throughput and best separates the T3D specific problems from those of the Y-MP.

The NQS system contains 2 sets of queues. The first set allows normal Y-MP work, and the second permits jobs to be launched on T3D. All the queues are "pipe-only", so job assignment to a given queue is determined automatically by the requested resource type. Currently, there is only one common batch queue for the 2 sets. Each set, in NQS terms, is defined as a "complex". Thus, a NQS job will be assigned to the T3D complex as soon as the user has specified a T3D resource. For those users allowed to work with the T3D, the access is therefore transparent.

Standard user profiles

Interactive profiles and machine access are controlled by the udb user data base. Batch profile limits are also defined in the udb, but the access control, which is only needed for public and non public queues, is done by setting the NQS limits and permissions for each queue.

Current standard profiles on the T3D are the following :

- batch mode : up to 256 PEs (the full T3D capacity), with various time limits;
- interactive mode : 16 PEs and 15 minutes. A few selected users are allowed to use 64 PEs.

We had to enlarge the Y-MP limits for the T3D interactive users, as the T3D cross compilers and linkers require about twice the resources the same utilities require for the Y-MP.

Training accounts may use up to 8 processors either in interactive or batch modes.

All the PATP group members are located in a separate EPFL building and have accounts on a local Sparc server of their own. In order to simplify the PATP work, all the users home directories are cross-mounted using NFS between the Y-MP and the local server, and a uniform distributed environment between the 2 machines is provided. So common tasks such as editing or viewing results on the Y-MP may be done remotely.

Y-MP/T3D batch production characteristics

With the emergence of large simulations, we observed that many users are chaining their tasks. In other words, it appears that a single NQS batch run will not solve the researchers problems : a large number of similar runs are necessary for complete results. As a consequence, many users insert a qsub command at the end of their job, and resubmit the same job in the same queue. Some sites do not allow this practice, but we think it is useful and convenient for many users.

Thus several independant jobs streams may run at the same time on the T3D, asking for the same amount and type of resources (mostly 32 or 64 PEs). This kind of user behaviour has some consequences. Most of the time, the load evolves in a predictable way. This might be usefull for the operator when he leaves his desk at night, as he can set up some NQS specific limits for a given type of load. The negative consequence is that sometimes such jobs will eventually fail, and resubmit themselves at a very high rate, merely because users generally do not check the status of their jobs before sending the next. Thus the NQS system may become overwhelmed by bad requests that may flood the system. Special measures had to be taken, because this situation often cannot be detected soon enough by the operator, and may lead to a disaster when the machine is in unattended mode. A procedure automatically detects such a situation, breaks the chain, mails a warning to the offending user, and closes his batch access until the user has corrected his problem. This procedure has been proved to be very useful, and is now likely to run less often, a few times per month.

T3D configuration

As for the NQS Y-MP queues, it was decided that the T3D NQS queues should just continue the standard T3D interactive users limits. Thus NQS queues have been initially set up for 16, 32, 64, 128 and 256 PEs, for times ranging from 300 up to 3600 seconds. Later, it was decided to also create a few queues with longer times, that would only be opened at night and during the weekend.

For Y-MP work, the PATP members can use the T3D NQS queues by just setting the number of PEs to zero. For the same amount of work, the T3D queues allow to the job to run faster than for the pure Y-MP queues. So the PATP members receive better service.

The T3D is configured with only one pool, for both batch and interactive usage.

In the beginning, the T3D load was controlled by a number of NQS limits applied on each NQS queue. It turned out very quickly that this method could not support many incoming batch requests, and was not giving good response time for short interactive commands, without constantly juggling with the NQS parameters.

Many small partitions (T3D jobs) were blocking larger jobs, that could not enter because of lack of space for a given partition shape. So we informed the users to specify an **express time** in their mppexec commands, whose maximum value was set to 600 seconds. This allowed small and short partitions to pass in front of other bigger and blocking partitions. Nevertheless, when the **maximum wait time** is reached (set to 1800 seconds), partitions are blocked again.

In order to better control job flow, we had to build up some local tools that would display the T3D load and provide some production figures and statistics. Collected information consists of : number of allocated PEs, number of allocated partitions, number of PEs waiting for allocation, and waiting times. This data is systematically recorded in a database. Abstracts from this data base can be collected at any time, and viewed with graphic tools, for the purpose of comparison between separate NQS configurations and load types (cf Fig 1)

In the meantime, with the new T3D OS releases, Cray Research provided some new commands. This really helped us in better understanding the underlying partition allocation process, and allowed us to define other scheduling tuning paradigms.

The T3D partitions and allocator properties

A number of tests were made in order to discover and better understand the T3D partition allocator algorithm. These tests were done by loading the partitions on various physical addresses on the T3D using the **-base** parameter of the mppexec command for some short programs. The **mppview** command showed us where the partitions were allocated on the T3D torus.

All the partitions have power of 2 fixed sizes and shapes.

We observed that each allowed partition geometry, as defined by the configuration driver, was set in such a manner that any shape could be divided on two half-sized partitions. In other words, each time a given allocated partition becomes free, among other possible combinations, exactly two other half-sized partitions can be recursively allocated at the same place for all partition sizes. This could have been not allowed, but the predefined set of shapes, as released by Cray, just have that property.

We observed that sometimes a given geometry could not be found, even if a sufficient numbers of PEs are seemingly available. This occurs because of the partition allocating history. Most of the times this is due to a partition with a small number of PEs (8, 16 or 32) which has not finished its work yet, whereas all surrounding partitions just have. Should this partition remain in the middle of an area usually allocated for larger jobs, then the other jobs may have to wait. This situation leads users to a number of complaints, because they were not all aware of the blocking mechanism.

Another property of the partition allocator is that it tries toexactly fill up holes on the T3D whenever possible, and to leave the largest space available for other bigger partitions. So the allocator seems to make the best decision when allocating new partitions.

Production principles; pseudo-pool concept

We are now experimenting several possible NQS scenarios with 3 goals in mind :

1. have the best interactive access for the T3D during the day. In other words, we garantee immediate interactive access to a fixed number of simultaneous interactive commands.
2. have the greatest throughput figures during the night, without locking out interactive accesses. Attaining this goal during the night will compensate the loss of throughput during the day time-frame because of the interactive service.
3. have the greatest throughput figures during the day, even if priority is given to interactive usage and small batch jobs.

Up to now, we had the following configuration scheme :

- Configure only one pool on the T3D, for batch and interactive, whose access is controlled by group membership set up at the configuration driver level;
- In order to achieve best overall throughput, there must be 2 distinct modes of operation :
  – during the day, allow a maximum number of PEs for batch processing, lower than the physical number of available PEs. This is equivalent to defining 2 virtual separate **pseudo-pools** for batch and interactive partitions, with a flexible limit in between. Batch partitions are allocated within their own global limit. Interactive partitions are first allocated on their own partition, and if needed can overflow into the batch area, should it have enough PEs. Then any new batch partition would have to eventually wait until the interactive partitions allocated in the batch area terminate. Inter-queue priorities are set up so small partitions enter first. This scheme allows interactive usage to have better priority over batch access;
  – during the night, set the maximum number of PEs for batch at the physical number of available PEs. This will likely priviledge batch jobs, and allow minimal possibilities for interactive usage. Inter-queue priority is set up so large jobs start processing before smaller ones (priorities are set in the reverse order).

We have been experimenting this configuration scheme by fixing the **complex mpp_pe_limit** to several values, but 224 seems to be the optimal during the day, and 256 during the night, for our type of load. The NQS limits since then play little role under these conditions.

We have been using this scheme for a few months, and our experience gave the following results :
During the day :

- immediate access to the T3D in most cases;
- batch jobs enter only if there are enough PEs available in their pseudo-pool;
- interactive partitions can be allocated without waiting by overflowing the NQS space if this space is not completely used by batch partitions;
- many interactive accesses can be honored per day without waiting;
- much less geometry contention situations

During the night :

- overall good throughput figures

but, during the day :

- nearly no possibility for large PEs jobs, even if they are of very short duration;
- no possibility of control on interactive usage;

- high cost involved because of the interactive service that must be permanently maintened.

and, during the night :

- reduced interactive access;
- need to set up a procedure when switching from day to night schedule;
- medium sized jobs may never get processed.

Nevertheless, analysis of daily statistics and reports showed us that during these last four months we improved our production figures from about 50% to more than 80% on the average. We experienced good batch throughput, and an increasing number of short duration interactive partitions (see fig 4, 5, 6 and 7).

Production scenario; pilot jobs

The idea is to define a set of batch job combinations, so the partitions would fully map onto the pseudo-pools at any time. Such combinations are labeled by a production level. Each level is defined by the number of combinations the largest partition allows for each given partition size. The number of various PEs sizes for each level would be set to 2. For instance, during the night, the production levels for 256, 128 and 64 PEs jobs are :

Level	Combination
1	1*256
2	2*128
3	1*128, 2*64
4	4*64

Going for example from level 1 to level 2 is fairly easy, as a finishing 256 PEs partition can immediately be replaced by two 128 PEs waiting jobs. Going in the reverse order is more difficult, because merely two 128 PEs jobs will not finish at the same time, and the freed space would be allocated to some less demanding PEs job.

This technique is particularly usefull when switching from day to night schedules, simply because 256 PEs and 128 PEs jobs usually cannot run during the day.

Thus we have set up some T3D overbooking techniques, so larger jobs normally not allowed to run at a certain level during the day can be initiated. These streams will then enter the execute queue, and will wait for PEs resources in a first-in first-served basis, until all preceeding jobs needing fewer PEs terminate.

In order to garantee that a unique chain of jobs at a given level will not be interrupted by smaller jobs comming from any another levels, a **pilot-job** is launched, just for the purpose of keeping the level active. This job kills itself when the chain is exhausted or the queue get closed.

This procedure can be initiated manually or automatically at any time, but primarily at schedule switching times. It is called when a different type of load change is detected, implying the need for an automatic production level change.

Future trends and perspectives

Cray Research recently provided the "rollin/rollout" feature. This has some major benefits :

- it is possible to temporarily pull a partition out of the T3D, and then to reallocate it at another location. This will be equivalent of moving a partition from one area to another, so a required geometry would be made available and a higher level job could enter.
- it allows to temporarily "swap out" a given partition, should a very high priority job be present.
- job checkpointing becomes now possible. This will allow us to save user jobs between T3D shutdowns.
- rollin/rollout may be initiated under the control of the user. This will give him program analysis possibilities foroff-line debugging purposes.

Conclusions

The tests we have made on the Unicos/Max OS has proven that the management of the Y-MP/T3D pair at our site is possible, and can give satisfying results in our attempts to provide a good interactive environment, along with some interesting production figures, without overloading the normal and pure YMP production service too much. Nevertheless, managing the T3D with NQS is not a straightforward process. The NQS system is very popular, but offers static scheduling mechanisms only. Many awkward local tools must be set up, which looks as remedies for the NQS deficiencies rather than real administering work on a MPP environment. Some "gang scheduler" mechanisms should be implemented into NQS.

Our tests also showed that the T3D management requires specific methods, due to its architecture with its own set of principles, which are very specific. They differ very much from those applicable for traditional time sharing systems.

For example, in a traditional PVP machine, even if the shared memory is not fully assigned, is is very easy to keep all CPUs active, thus achieving 100% usage. A few low priority, long and small jobs running in the background will just allow this To achieve good productivity with a MPP machine, a maximum number of PEs must be assigned, and some non trivial full mapping mechanisms must be setup. And should a single job requiring a few PEs be present, then the production figures may decrease, due to the geometry blocking mechanism.

Although we succeeded to avoid most of the blocking situations due to unavailable geometries, we expect to improve the overall T3D productivity with the "rollin/rollout" feature. But if we have now the tool, still some "gang scheduling" strategies and mechanisms have to be set up.

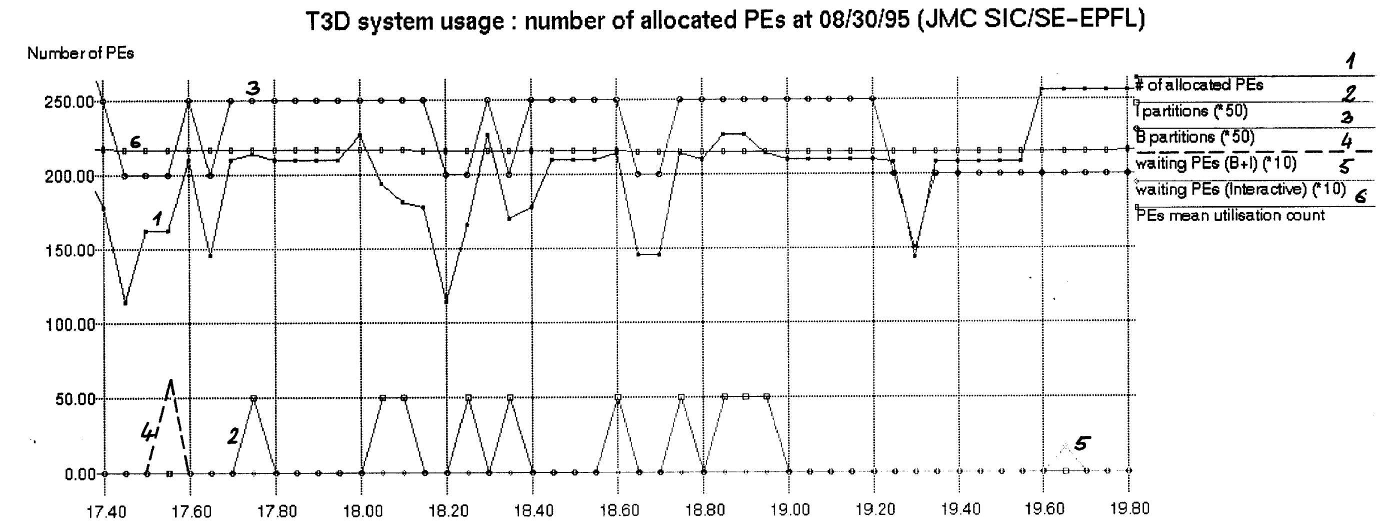
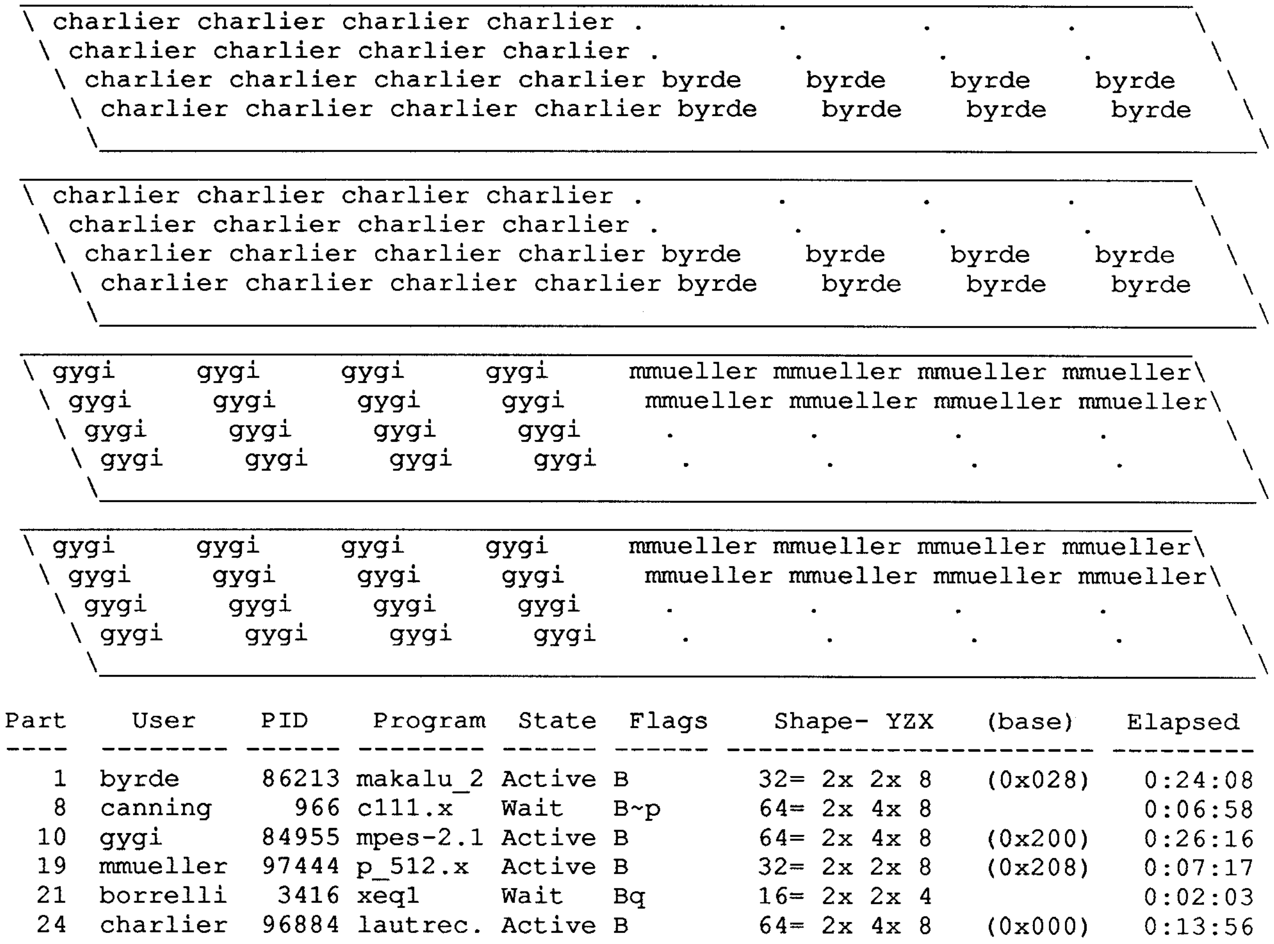


Figure 1.

```
 _______________________________________________________________________
\ .       .       .       .       .       .       .       .       \
 \ .       .       .       .       .       .       .       .       \
  \ .       .       .       .    borrelli borrelli .       .       \
   \ .       .       .       .    borrelli borrelli .       .       \
    \_______________________________________________________________\

 _______________________________________________________________________
\ .       .       .       .       .       .       .       .       \
 \ .       .       .       .       .       .       .       .       \
  \ .       .       .       .    borrelli borrelli .       .       \
   \ .       .       .       .    borrelli borrelli .       .       \
    \_______________________________________________________________\

 _______________________________________________________________________
\ devita   devita   devita   devita   .       .       .       .       \
 \ devita   devita   devita   devita   .       .       .       .       \
  \ devita   devita   devita   devita   .       .       .       .       \
   \ devita   devita   devita   devita   .       .       .       .       \
    \_______________________________________________________________\

 _______________________________________________________________________
\ devita   devita   devita   devita   .       .       .       .       \
 \ devita   devita   devita   devita   .       .       .       .       \
  \ devita   devita   devita   devita   .       .       .       .       \
   \ devita   devita   devita   devita   .       .       .       .       \
    \_______________________________________________________________\

Part   User     PID    Program  State  Flags   Shape- YZX            (base)   Elapsed
----  -------- ------ -------- ------ ------ ---------------------- ---------  ---------
  17  devita    42375 ccp.x    Active B        64= 2x 4x 8         (0x200)    0:22:08
  23  borrelli  47686 xeql     Active B        16= 2x 2x 4         (0x028)    0:07:53
```

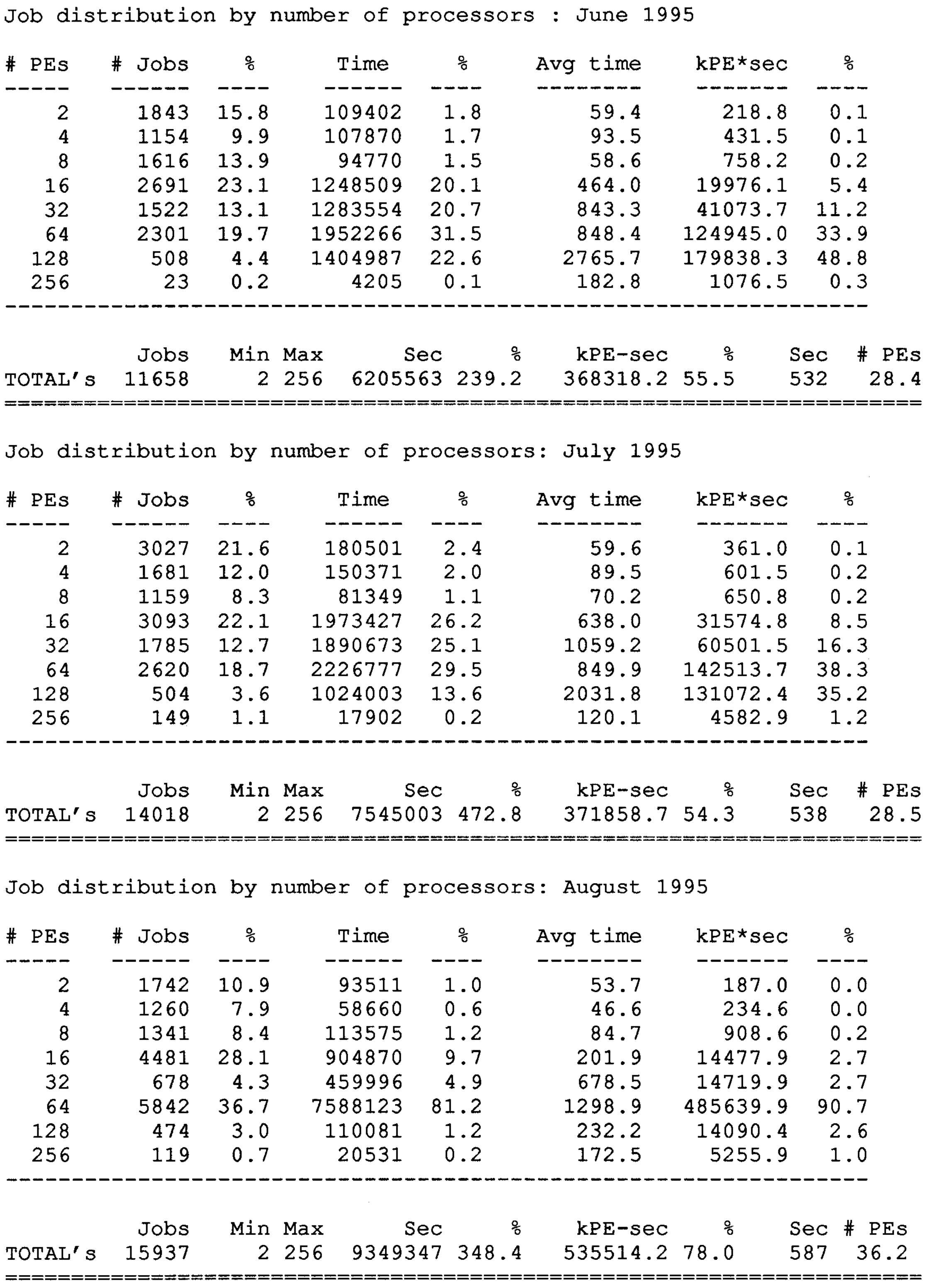
Figure 2: This slide shows how a small partition can prevent a 128 PEs job to enter. In all such cases, specially if the small job has to run for a long time, there is potentially a great waste of resources. This is why such jobs normally are not allowed to run by night. Nevertheless, reallocating this partition with the "rollin/rollout" feature to another base address will allow 128 PEs job to enter.



```
 ________________________________________________________________________
\ charlier charlier charlier charlier .        .        .        .        \
 \ charlier charlier charlier charlier .        .        .        .        \
  \ charlier charlier charlier charlier byrde    byrde    byrde    byrde    \
   \ charlier charlier charlier charlier byrde    byrde    byrde    byrde    \
    \________________________________________________________________________\
 ________________________________________________________________________
\ charlier charlier charlier charlier .        .        .        .        \
 \ charlier charlier charlier charlier .        .        .        .        \
  \ charlier charlier charlier charlier byrde    byrde    byrde    byrde    \
   \ charlier charlier charlier charlier byrde    byrde    byrde    byrde    \
    \________________________________________________________________________\
 ________________________________________________________________________
\ gygi     gygi     gygi     gygi     mmueller mmueller mmueller mmueller\
 \ gygi     gygi     gygi     gygi     mmueller mmueller mmueller mmueller\
  \ gygi     gygi     gygi     gygi     .        .        .        .        \
   \ gygi     gygi     gygi     gygi     .        .        .        .        \
    \________________________________________________________________________\
 ________________________________________________________________________
\ gygi     gygi     gygi     gygi     mmueller mmueller mmueller mmueller\
 \ gygi     gygi     gygi     gygi     mmueller mmueller mmueller mmueller\
  \ gygi     gygi     gygi     gygi     .        .        .        .        \
   \ gygi     gygi     gygi     gygi     .        .        .        .        \
    \________________________________________________________________________\

Part   User     PID    Program  State  Flags    Shape- YZX    (base)   Elapsed
---- -------- ------ -------- ------ ------ ----------------------- ---------
   1 byrde     86213 makalu_2 Active B        32= 2x 2x 8   (0x028)   0:24:08
   8 canning     966 c111.x   Wait   B~p      64= 2x 4x 8             0:06:58
  10 gygi      84955 mpes-2.1 Active B        64= 2x 4x 8   (0x200)   0:26:16
  19 mmueller  97444 p_512.x  Active B        32= 2x 2x 8   (0x208)   0:07:17
  21 borrelli   3416 xeql     Wait   Bq       16= 2x 2x 4             0:02:03
  24 charlier  96884 lautrec. Active B        64= 2x 4x 8   (0x000)   0:13:56
```

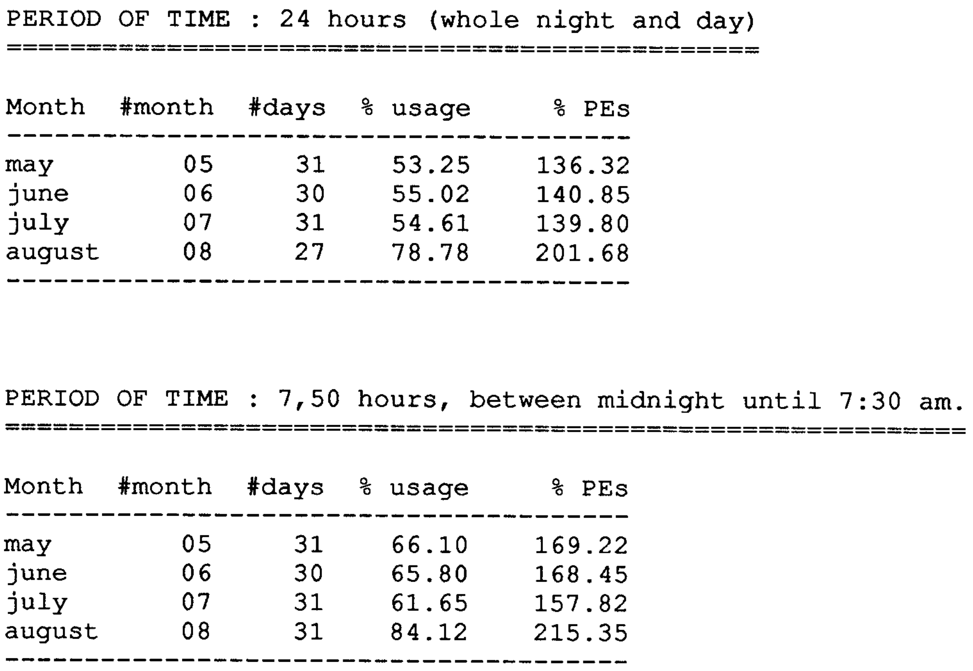
Figure 3: This slide illustrates the geometry contention problem. If the "byrde" and "mmueller" partitions could be located on the same 2 planes, then the "canning" partition would be able to enter. Even if there are enought PEs available, the "borrelli" partition is queued and has to wait, because it is blocked by the "canning" partition.

The "rollin/rollout" would resolve the contention problem.



```
Job distribution by number of processors : June 1995

# PEs    # Jobs    %      Time     %      Avg time     kPE*sec    %
-----    ------   ----   ------   ----    --------     -------   ----
    2      1843   15.8   109402    1.8        59.4       218.8    0.1
    4      1154    9.9   107870    1.7        93.5       431.5    0.1
    8      1616   13.9    94770    1.5        58.6       758.2    0.2
   16      2691   23.1  1248509   20.1       464.0     19976.1    5.4
   32      1522   13.1  1283554   20.7       843.3     41073.7   11.2
   64      2301   19.7  1952266   31.5       848.4    124945.0   33.9
  128       508    4.4  1404987   22.6      2765.7    179838.3   48.8
  256        23    0.2     4205    0.1       182.8      1076.5    0.3
-----------------------------------------------------------------

          Jobs    Min Max      Sec      %     kPE-sec     %     Sec   # PEs
TOTAL's  11658      2 256  6205563 239.2   368318.2 55.5     532    28.4
=====================================================================

Job distribution by number of processors: July 1995

# PEs    # Jobs    %      Time     %      Avg time     kPE*sec    %
-----    ------   ----   ------   ----    --------     -------   ----
    2      3027   21.6   180501    2.4        59.6       361.0    0.1
    4      1681   12.0   150371    2.0        89.5       601.5    0.2
    8      1159    8.3    81349    1.1        70.2       650.8    0.2
   16      3093   22.1  1973427   26.2       638.0     31574.8    8.5
   32      1785   12.7  1890673   25.1      1059.2     60501.5   16.3
   64      2620   18.7  2226777   29.5       849.9    142513.7   38.3
  128       504    3.6  1024003   13.6      2031.8    131072.4   35.2
  256       149    1.1    17902    0.2       120.1      4582.9    1.2
-----------------------------------------------------------------

          Jobs    Min Max      Sec      %     kPE-sec     %     Sec   # PEs
TOTAL's  14018      2 256  7545003 472.8   371858.7 54.3     538    28.5
=====================================================================

Job distribution by number of processors: August 1995

# PEs    # Jobs    %      Time     %      Avg time     kPE*sec    %
-----    ------   ----   ------   ----    --------     -------   ----
    2      1742   10.9    93511    1.0        53.7       187.0    0.0
    4      1260    7.9    58660    0.6        46.6       234.6    0.0
    8      1341    8.4   113575    1.2        84.7       908.6    0.2
   16      4481   28.1   904870    9.7       201.9     14477.9    2.7
   32       678    4.3   459996    4.9       678.5     14719.9    2.7
   64      5842   36.7  7588123   81.2      1298.9    485639.9   90.7
  128       474    3.0   110081    1.2       232.2     14090.4    2.6
  256       119    0.7    20531    0.2       172.5      5255.9    1.0
-----------------------------------------------------------------

          Jobs    Min Max      Sec      %     kPE-sec     %     Sec  # PEs
TOTAL's  15937      2 256  9349347 348.4   535514.2 78.0     587   36.2
=====================================================================
```

Figure 4: This slide shows the increased throughput during August (78%), relatively June (55%) and July (54%). This was accomplished by reducing T3D wait times with the NQS global mpp_pe_limit set to 224 by day and 256 by night.

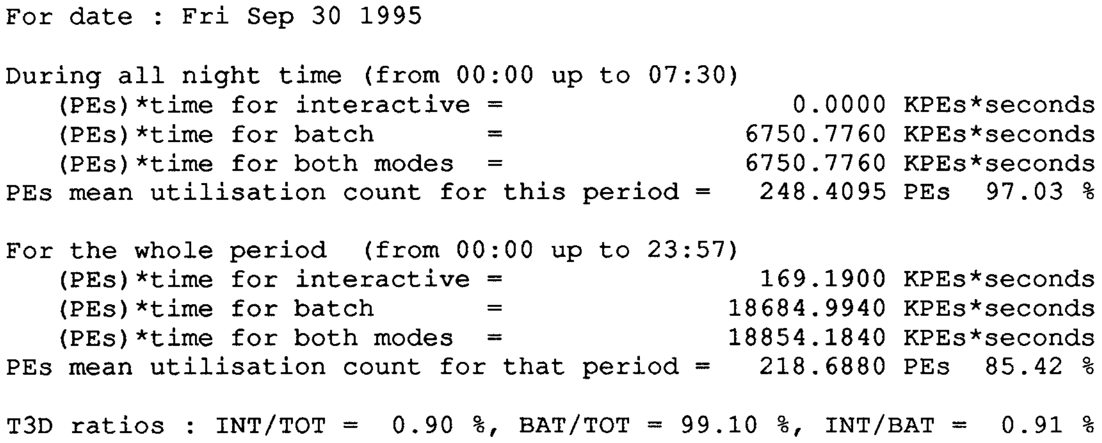


```
PERIOD OF TIME : 24 hours (whole night and day)
===============================================

Month   #month  #days  % usage     % PEs
---------------------------------------
may        05     31    53.25    136.32
june       06     30    55.02    140.85
july       07     31    54.61    139.80
august     08     27    78.78    201.68
---------------------------------------


PERIOD OF TIME : 7,50 hours, between midnight until 7:30 am.
============================================================

Month   #month  #days  % usage     % PEs
---------------------------------------
may        05     31    66.10    169.22
june       06     30    65.80    168.45
july       07     31    61.65    157.82
august     08     31    84.12    215.35
---------------------------------------
```

Figure 5:

This figures shows:

1) how production figures vary between night and day (about 15% PEs). This is due to the T3D PE space we have to reserve for the interactive service we have to provide by day and the higher priority for small NQS jobs.
2) the increase of throughput during the 4 last months.

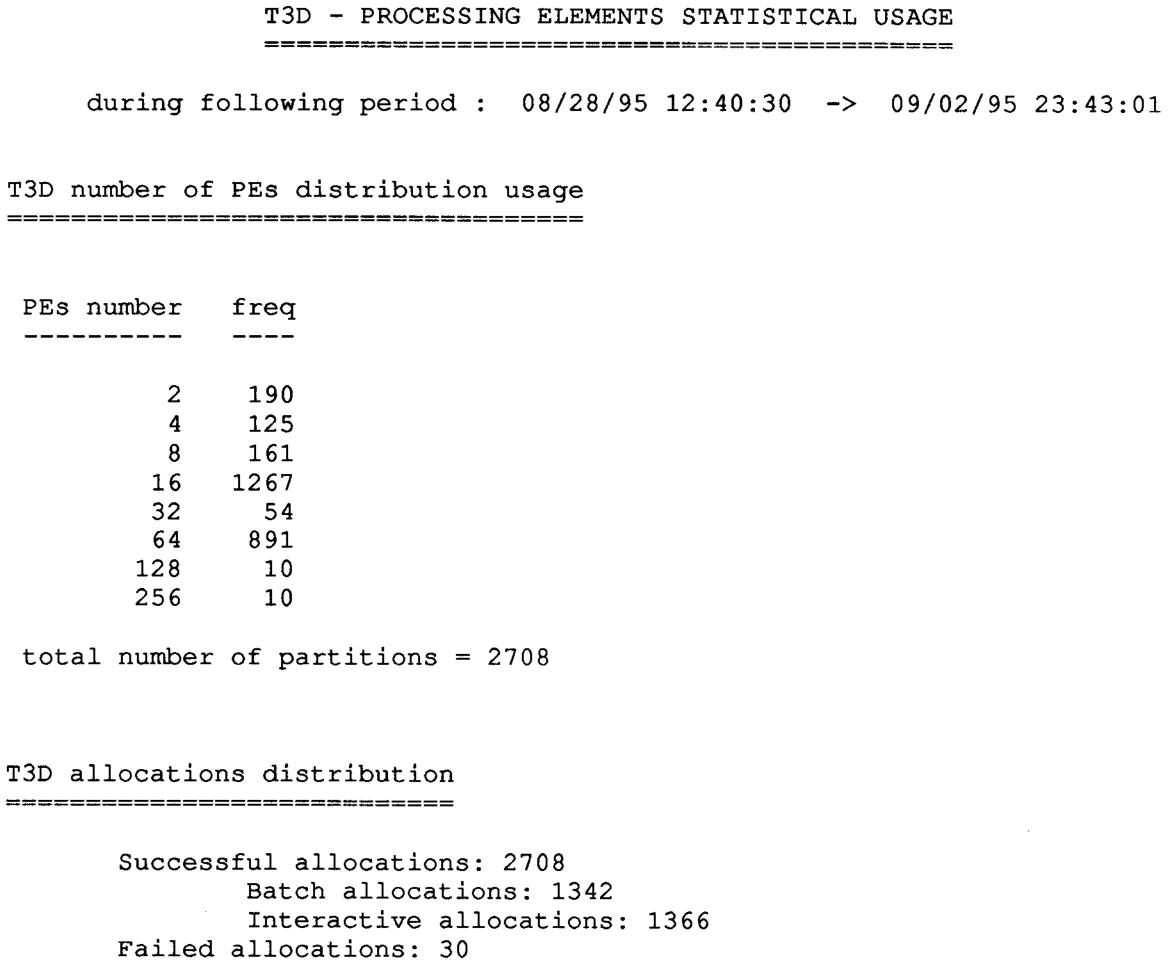


```
For date : Fri Sep 30 1995

During all night time (from 00:00 up to 07:30)
   (PEs)*time for interactive =                 0.0000 KPEs*seconds
   (PEs)*time for batch       =              6750.7760 KPEs*seconds
   (PEs)*time for both modes  =              6750.7760 KPEs*seconds
PEs mean utilisation count for this period =   248.4095 PEs  97.03 %

For the whole period  (from 00:00 up to 23:57)
   (PEs)*time for interactive =               169.1900 KPEs*seconds
   (PEs)*time for batch       =             18684.9940 KPEs*seconds
   (PEs)*time for both modes  =             18854.1840 KPEs*seconds
PEs mean utilisation count for that period =   218.6880 PEs  85.42 %

T3D ratios : INT/TOT =  0.90 %, BAT/TOT = 99.10 %, INT/BAT =  0.91 %
```

Figure 6:

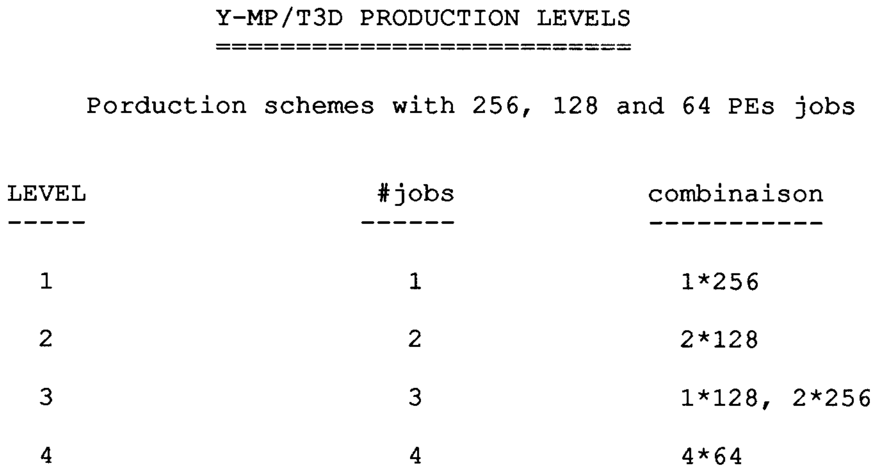
These figures show:

1) good throughput during night (97%)
2) during day, even if there are many interactive allocations (see Fig 5), the low ratio between interactive and batch resources



```
                 T3D - PROCESSING ELEMENTS STATISTICAL USAGE
                 ===========================================

           during following period :  08/28/95 12:40:30  ->  09/02/95 23:43:01


  T3D number of PEs distribution usage
  ====================================


   PEs number   freq
   ----------   ----

            2    190
            4    125
            8    161
           16   1267
           32     54
           64    891
          128     10
          256     10

   total number of partitions = 2708


  T3D allocations distribution
  ============================

         Successful allocations: 2708
                 Batch allocations: 1342
                 Interactive allocations: 1366
         Failed allocations: 30
```

Figure 7: This figure show the relatively high number of interactive partitions relatively to the batch partitions: there are about as many interactive partitions as batch partitions; but interactive partitions count for about only 1% of the total production (see Fig 6).



```
                         Y-MP/T3D PRODUCTION LEVELS
                         ==========================

                 Porduction schemes with 256, 128 and 64 PEs jobs


           LEVEL                    #jobs              combinaison
           -----                   ------              -----------

             1                        1                  1*256

             2                        2                  2*128

             3                        3                  1*128, 2*256

             4                        4                  4*64
```

Figure 8: This shows some possible scenario production levels for large jobs. Going from level 1 to any other level is a straigthforward process. Goiong into reverse way requires some automatic or manual action, that will free up enough PEs