Lawrence Berkeley National Laboratory
Recent Work

Title
RESQme Studies for SDC Regional Computing Centers

Permalink
https://escholarship.org/uc/item/8jt69297

Author
Day, C.T.

Publication Date
1993-02-01

eScholarship.org Powered by the California Digital Library
University of California

LBL-33540
UC-405

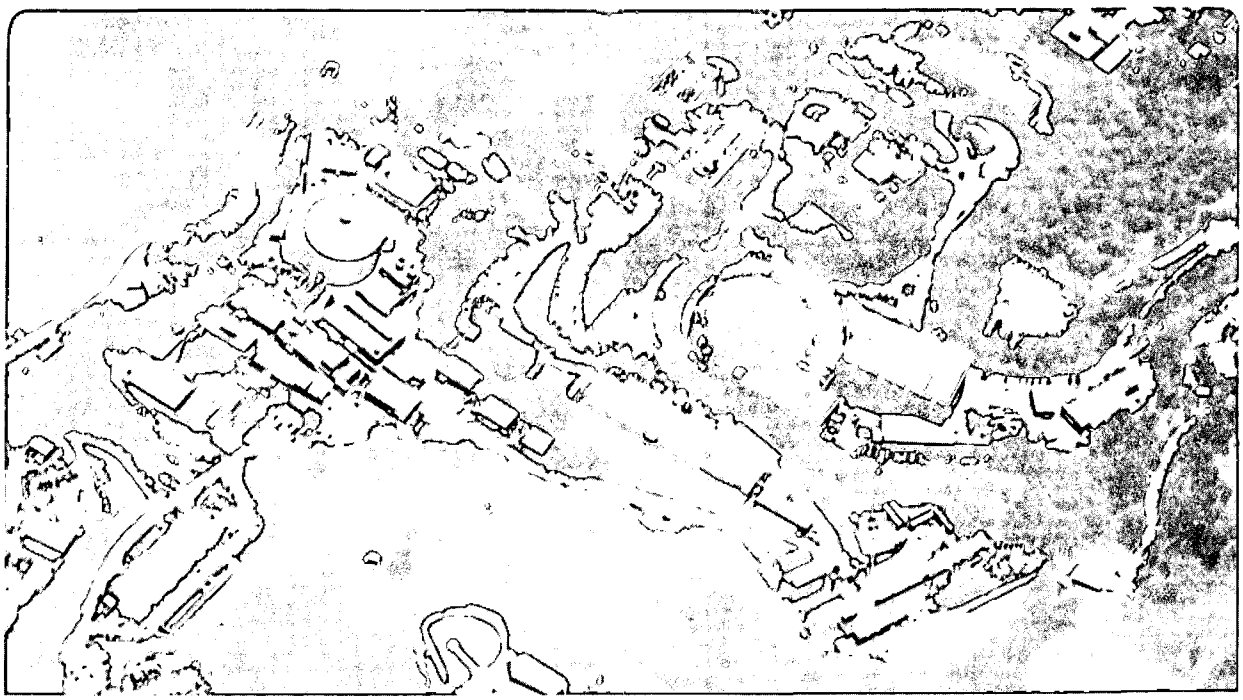
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Information and Computing Sciences Division

RESQme Studies for SDC Regional Computing Centers

C.T. Day

February 1993



LOAN COPY
Circulates
for 4 weeks
Bldg. 50 Library.
Copy 2
LBL-33540

Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. Neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California and shall not be used for advertising or product endorsement purposes.

This report has been reproduced directly
from the best available copy.

Available to DOE and DOE Contractors
from the Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, VA 22161

Lawrence Berkeley Laboratory is an equal opportunity employer.

This publication has been reproduced from the best available copy

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-33540

RESQme Studies for SDC Regional Computing Centers

C.T. Day[1]

Information & Computing Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

February 1993

1 This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

2 "Technical Design Report," SDC-92-201, SSCL-SR-1215, The Solenoidal Detector Collaboration, 1992.

3 "An Introduction to the RESearch Queueing Package for Modeling Computer Systems and Communication Networks," Research Report RA 208 (#75028), 1991, E.A. Macnair and R.F. Gordon, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598.

Printed on recycled paper

SDC Note #92-230

RESQme Studies for SDC Regional Computing Centers

C. T. Day[1]

Information and Computing Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 97420

Introduction—

The Technical Design Report for the SDC[2] proposes a model for offline computing which includes a computing center at the SSCL, containing all of the physics data, together with a number of regional computing centers around the world. These regional centers would contain subsets of the data, and would support the daily work of most physicists. For large or unusual requests, where the data are not held at the regional center, the requests would automatically be forwarded to the SSCL. It is assumed that the "closeness" of the regional centers and their reduced demand from fewer users would result in improved system performance.

Such a system is too complex to model analytically; simulation is the only viable approach. However, Monte Carlo models built from scratch for complicated systems are very difficult to maintain and hard to modify. Fortunately, we have obtained from IBM a modeling framework, *RESQme*[3], explicitly designed for building statistical models of computer systems. This note describes a first pass at modeling the proposed offline system.

Central Model—

For ease of explanation, I will begin by describing a simpler model which deals only with a single, central processing center at the SSCL. However, this center is assumed to have five processing engines, in order to match the processing power that would be available in a configuration with three regional centers in the USA, one in Japan and one in Europe. Figure 1 shows the overall model, while Listing 1 is the corresponding textual input to the simulation engine for the model and all its sub-models. Note that the NUMERIC PARAMETER section of this listing describes all of the quantities which can be varied from one run of the model to the next.

Jobs are introduced into the system from the *long* and *short* sources. The intervals between arrivals of jobs are chosen from exponential distributions with different means for the long and short jobs. The mean values can be set for each simulation run. All times are measured in minutes; the global *clock* variable maintains the simulated time elapsed at the SSCL.

---

[1] This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098.

[2] "Technical Design Report", SDC-92-201, SSCL-SR-1215, The Solenoidal Detector Collaboration, 1992.

[3] "An Introduction to the RESearch Queueing Package for Modeling Computer Systems and Communication Networks," Research Report RA 208 (#75028), 1991, E. A. Macnair and R. F. Gordon, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598.

The short job stream is further modulated by the sub-model *workday*, shown in Figure 2. Short jobs enter the sub-model through the input node *ins*. For tracking purposes, a job variable is set for each job in the *regattrs* node. This variable labels the job randomly as coming from one of the five regional areas. Each job is then sent, based on this label, to node *setxxx* where a second variable is set corresponding to the local time of day that the job was submitted, given the time zone of the originating area. Jobs which would be submitted outside of local working hours are discarded into the *SINK* node; others return to the main model through the *outs* node. The net result of this sub-model is to limit short job submission to normal working hours, but with users distributed around the world. Since long jobs bypass this sub-model, they can be submitted at any time of day, such as from a batch queue at midnight.

Back at the main model, short jobs next enter the *netshort* passive queue by taking a token at the *bgnshort* allocate node. The job then passes through the *sscl* sub-model, releases its token at the *endshort* release node, thereby exiting the passive queue, and leaves the model through the *SINK* node. Long jobs similarly pass through the *netlong* passive queue by way of the *bgnlong* allocate node, the *sscl* sub-model, the *endlong* release node and the *SINK*. The passive queues are useful since RESQme automatically keeps statistics on the lengths of times tokens are held; these statistics then provide overall measures of the system behavior. Enough tokens are supplied to the passive queues that incoming jobs never have to wait for a token.

The heart of the simulation is in the *sscl* sub-model. There are separate pathways for the long and short jobs, but all jobs share the physical resources of tape robots, CPUs and network bandwidth.

Short jobs enter from the *ins* nodes. A parameter sets the probability that the required data is already on disk. A corresponding random fraction of the short jobs are queued for service by the CPUs. The remainder must first wait for the tape robot, thereby modeling the time spent waiting for the data to be fetched. In either case, all short jobs take a token from the *shorttime* passive queue at either the *startprocs* or *starttapes* allocate node, hold it while being serviced and return it at the *stops* release node. Like the main model's *netshort* passive queue, the statistics for *shorttime* summarize the behavior of short job processing. There are an effectively infinite number of tokens for short jobs so that an arbitrary number of them maybe serviced simultaneously, but there are only *cljobs* tokens in the *longtime* queue, limiting the number of concurrently running long jobs. This prevents a large number of long jobs from consuming the CPUs and tape robots and starving out the short jobs.

All jobs in this sub-model are eventually serviced by the *regprocq* queue. Short jobs are directed to the class list *regprocs* and given a fixed work demand, while long jobs go to the class list *regprocl* and are given a different, fixed work demand. The work demand represents the time in minutes that a standard CPU server with nothing else to do would take to process the job. The *regprocq* in fact has *cservers* independent processors which can work in parallel—if there are *cservers* or fewer jobs active, they will all complete at full speed. If there are more than *cservers* jobs active, the processors will be shared and jobs will take proportionately longer to complete. All processors are taken to have the same intrinsic speed.

Jobs which must have their data fetched to disk go first to the *tapeq* queue, short ones to the *tapeprocs* class and long ones to the *tapeprocl* class. Here all short jobs are given a fixed work demand and all long jobs are given a distinct, fixed work demand. These work demands correspond to the time in minutes required to fetch, mount and position the volume with the data and then to transfer the data to disk. There are *ctapesvrs* independent robots, but these cannot be shared by concurrent jobs. Hence, at most *ctapesvrs* tape operations can be ongoing at one time. Once a job's data is on disk, it proceeds to the *regprocq* like any other job.

The final task of this sub-model is the transmission of the results back to the original users. This is modeled by having all jobs pass through the *networkq* queue, short jobs in the *shortnet* class and long jobs in the *longnet* class. The work demands are again fixed for each class and computed as the time for transmission over a standard speed network (typically 100 Mb/sec) divided by the relative speed of the network being modeled. This allows the speed of the network to be varied as an independent parameter.

The network is simulated by a very simple queuing discipline—all jobs start as soon as they arrive and complete in their service time regardless of how many other jobs are being serviced concurrently. As long as the network is so lightly loaded that multiple jobs are only rarely active on the network, this should be an acceptable approximation. If the network becomes heavily loaded, however, the bandwidth used becomes unbounded and this queue will have to be replaced with a more realistic one.

Finally, short jobs exit the sub-model through the *outs* node while long jobs exit through the *outl* node.

Regional Model—

Most of the actual simulations reported on here were conducted with a more flexible model than the Central Model described above. However, this more general Regional Model is very closely related to its simpler predecessor. Basically, it is constructed by making an array of computing centers, each a duplicate of the Central Model described above and dedicated to the jobs submitted within one region, and adding one additional center at the SSCL to handle jobs whose data do not reside at the local center. This special SSCL center always has the data, at least on tape.

The overall form of the Regional Model is shown in Figure 4, and Listing 2 is the complete description of the model and all its sub-models as input to the simulation engine.

Except that most nodes are now promoted to arrays of nodes, the overall structure is very similar to the Central Model. Jobs still originate in the *short* and *long* source nodes, only now there is one such node for each regional center. Each regional center's stream of short jobs is modulated down to normal local working hours by its own *workday* sub-model, shown in Figure 5. Long jobs pass through the *workday* sub-models unaffected; the extra path is introduced purely for technical reasons involving restrictions on how array nodes may be strung together.

Each short or long job picks up a token from its corresponding *bgnshrt* or *bgnlong* allocate node. However, all short jobs share a single pool of tokens as do the long jobs. This means that the summary statistics measure the overall system performance, not that of the individual centers.

Upon exit from the allocate nodes, a fraction of the jobs are diverted to the central *sscdata* sub-model; these represent jobs whose data are not local. Once the data for such a job are fetched, the job rejoins the stream at its *regcntrs* sub-model for possible further processing and for transmittal to the user. The same sub-model is used for all the *regcntrs* nodes and for the *sscdata* node; it is the same sub-model as used as the *sscl* node in the Central Model.

On exit from the regional center sub-models, all jobs return their tokens and exit the model through the *SINK* node.

There is one set of parameters to describe all of the regional centers and a separate set to describe the SSCL data center. By careful adjustment of these parameters, a variety of topologies and

policies can be realized without rebuilding the model. For example, to model a policy where all long jobs are done at the SSCL and all short jobs at the Regional Centers, set *lregprob* to 0.0 and *sregprob* to 1.0; *rlproctime* must also be set to 0.0 to prevent long jobs from being processed a second time at the regional centers. To eliminate the regional centers altogether, set all of *lregprob, sregprob, rlproctime* and *rsproctime* to 0.0; to eliminate the SSCL center, set *lregprob* and *sregprob* to 1.0.

There are several weaknesses of this model topology one should be careful about. If one sends some short jobs to regional centers and some to the SSCL and has processing done at each place, then jobs sent to the SSCL will necessarily have processing done at *both* places. Also, if the SSCL is used only as a data store and no processing is done there, there will be occasions when data come from the SSCL to the regional center and go immediately onto tape before the job can process them.

Simulation Runs—

A series of simulations were run to explore the effects of various distributions of jobs between the regional centers and the SSCL center. All simulations were run until 2000 long jobs have been processed. Three basic topologies were modeled—1) all jobs handled by the SSCL center, 2) all jobs handled at the regional centers, and 3) all short jobs handled by the regional centers while all long jobs are fetched from the SSCL center but processed on the regional centers. Since the tape transfer times were suspected of being a limiting factor, each of these configurations was run with one tape robot per CPU processor and again with two robots per processor.

The overall set of simulations was designed to address the question of how effective the cacheing of data on disk has to be to get acceptable performance from the system. To this end, for each configuration described above, 16 runs were made for all combinations of disk cache hit probabilities given in Table 1. For each such run, many statistics are produced; Figures 7–18 plot the average queuing time in minutes, the maximum queuing time in minutes and the maximum queue length in jobs over the course of the simulation. Each of these quantities is for the system as a whole and is plotted separately for long and short jobs. Other parameters of the models were kept fixed over these runs. These other values are listed in Table 2.

Prob. Long job on disk	20%	40%	60%	80%
Prob. Short job on disk	20%	40%	60%	80%

Probability that Job Data Reside on Disk
Table 1

Process time for long job	20 min
Process time for short job	2 min
Tape fetch time for long job	102 min
Tape fetch time for short job	12 min
Mean arrival time for long jobs	90 min
Mean arrival time for short jobs	1.5 min
Total number of CPU servers	5
Number of long jobs per server	1
Network transfer time for short jobs	2 min
Network transfer time for long jobs	20 min
{Central-to-regional transfers 10 times faster}	

Model Parameters Fixed Throughout Runs
Table 2

Results—

Figure 7 shows the results for short jobs in the single center scenario with one tape robot per CPU. The top series of graphs show the average time the system takes to handle a short job [solid line, left scale] for each short job cache hit probability [gray bars, right scale] and for each long job cache hit probability [left-to-right graph series, labeled above]. The decrease in the queuing time in any graph as a function of short hit probability reflects the better performance of the disk relative to the tape. There is a limit to how small this time can be—all jobs require at least two minutes of processing time and two minutes of network transfer time.

For the 20% short hit ratio, the average queuing time decreases as the long hit ratio gets larger. This is because, with a low hit ratio, many short jobs must go to tape; if they get in the queue when all robots are busy with long jobs, they will have to wait a considerable time. (Long job data is assumed to take 120 minutes to transfer from tape at 10 MB/sec.) Compare this with the top series of graphs in Figure 9. These have the same parameters except that the number of tape robots is doubled. Note that the improvement with long job hit ratio has disappeared. This is because the number of long jobs that can be active in the system is less that the number of tape robots, so short jobs are never trapped behind long jobs. (This reasoning suggests that doubling the number of robots is overkill; preventing long jobs from hogging all of them is enough.)

The same argument works for the other two series of graphs in Figures 7 & 9 which plot the maximum, *i.e.*, worst case, short job queuing time and the maximum number of short jobs in the system at any instant. Note that, from the middle series of Figure 9, even with two robots per processor, there are still times when all robots are busy, some of them with long jobs; as the long hit ratio gets larger, this happens less often and the worst case improves.

One should keep in mind the different meanings of the average and maximum queuing times. The average measures the overall throughput of the system and a low value keeps the system administrator happy. The maximum queuing time is how long some user has had to wait for a result. Large values will make that user *very* unhappy. Notice from Figures 7 & 9 that adding tape robots make only a small improvement in the average time once the short hit ratio is at 40% or above. However, the maximum queuing time decreases by a factor of about four by adding robots, regardless of the long job hit ratio. This translates directly into a cost issue, since it take a lot of

disk to keep the long job hit ratio high.

Figures 8 & 10 plot corresponding statistics for long jobs given the same two configurations as Figures 7 & 9. The average queuing time shows little change with the short job hit ratio. This is to be expected since waiting for a short job to release a robot is always a small fraction of the time that the long job will use the robot. The general decrease with rising long job hit ratio is the usual effect of the disk cache. The maximum long job queuing time drops when extra robots are added, especially when the short job hit ratio is low. Getting the flurry of short jobs out of the way helps long jobs, but the effect is not so dramatic. (Be cautious examining worst case graphs. By there nature, they are very sensitive to statistical fluctuations.)

Figures 11–14 show similar graphs for the topology where all jobs are handled at their local regional centers and none at the SSCL. For the first two figures, each center has one tape robot while for the others, each has two. The general trends and arguments still hold, but many of the absolute times are mostly much worse. One might argue that, for the double drive case, long jobs are not too badly effected; this is because of the long job cap. The *average* short job queuing time for the two tape case also looks acceptable, but the *maximum* time is much worse. The demand for the tape robots at each center is too large to be handled well by only two robots. The centralized case performs much better for the same number of total robots.

Finally, Figures 13–18 display similar series for a topology where all short jobs are handled only at the regional centers, and long jobs all have their data fetched from the SSCL, but the processing is is still done in the regional centers. The long job limit is one at each regional center and there are either one or two tape robots at each center. (This makes for either one or two more robots than in the previous cases.) Overall this configuration tends to decouple the two jobs streams. Note that the short job statistics now show almost no change with long job hit ratio. If the short job hit ratio is kept high, then a second robot is unnecessary, however, the cost for lowering the hit ratio is much greater with only one robot.

With all long job data fetches concentrated at the SSCL, having only one robot is a disaster if the long job hit ratio is low. Raising the hit ratio is effective, but requires a lot of disk; adding tape robots may be more effective. Note that a high short job hit ratio with one robot per center and two or more at the SSCL, has the potential for good performance with the fewest drives.

Summary—

This particular model set points out the potential impact of the comparatively long tape transfer times. The good arrangements have robots concentrated so that any given job has as many robots as possible to choose from. However, keeping the two job streams separate and tailoring the hardware for each may well lead to a cheaper system.

Comments for future studies—

Service times in these models are all uniquely determined by whether a job is short or long. Clearly, these times should be given some distribution. It is easy to do this with RESQme and has not been done so far purely to simplify this initial modeling effort.

Since the short and long job streams are so distinct, the interference between them can be made clear in the above models. With fuzzier distributions, the ability to separate the jobs and tune the hardware to each type is probably diminished.

The tape robots appear to be a potential bottle neck. One possible improvement is to use a more sophisticated queuing policy. For example, one could give priority to short jobs and have them be started before any long jobs that may be in the queue; however, one would have to be sure that the long jobs are not starved for service. Or, since the service time for the tape transfers can be accurately estimated, the long transfers could be made interruptible, as long as the restart time is less than the time to complete the long transfer, for example.

It has already been pointed out that the network model has a potentially unbounded bandwidth and should be more accurately portrayed. In a similar vein, there is only a limited backplane bandwidth available for transferring data among the tapes, disks and CPUs; this should also be explicitly modeled.

In the Regional Model, the regional centers are meant to act as caches for the SSCL data center, just as the disks are cache for the tapes and memory is cache for the disks. Caches work when accesses are not truly random and the cache can keep the most useful data in the fastest device. One of the goals of the database computing effort is to reorganize event data and its access to allow for successful caching automatically. It is still an open question if this is possible.

One particularly counter-intuitive result has been the minor rôle played by the network in these models. One reason is that the network bandwidths assumed (10MB/sec to users and 100MB/sec to regional centers) is much larger that current typical disk bandwidths, let alone network capacity. Also, the numbers assumed provide the larger capacity for the longer links, which is not what we are used to. However, it is not at all clear that these bandwidths can be routinely obtained at reasonable cost. Furthermore, the models seem to imply that there is some nexus at the SSCL capable of handling 100MB/sec each from the tape robots, disk arrays, network and memory. This is a very high net throughput and may not be achievable. In this case, more but slower systems at the regional centers, perhaps coupled with more but cheaper tape robots, may provide the best solution. More detailed modeling is needed to examine these questions.

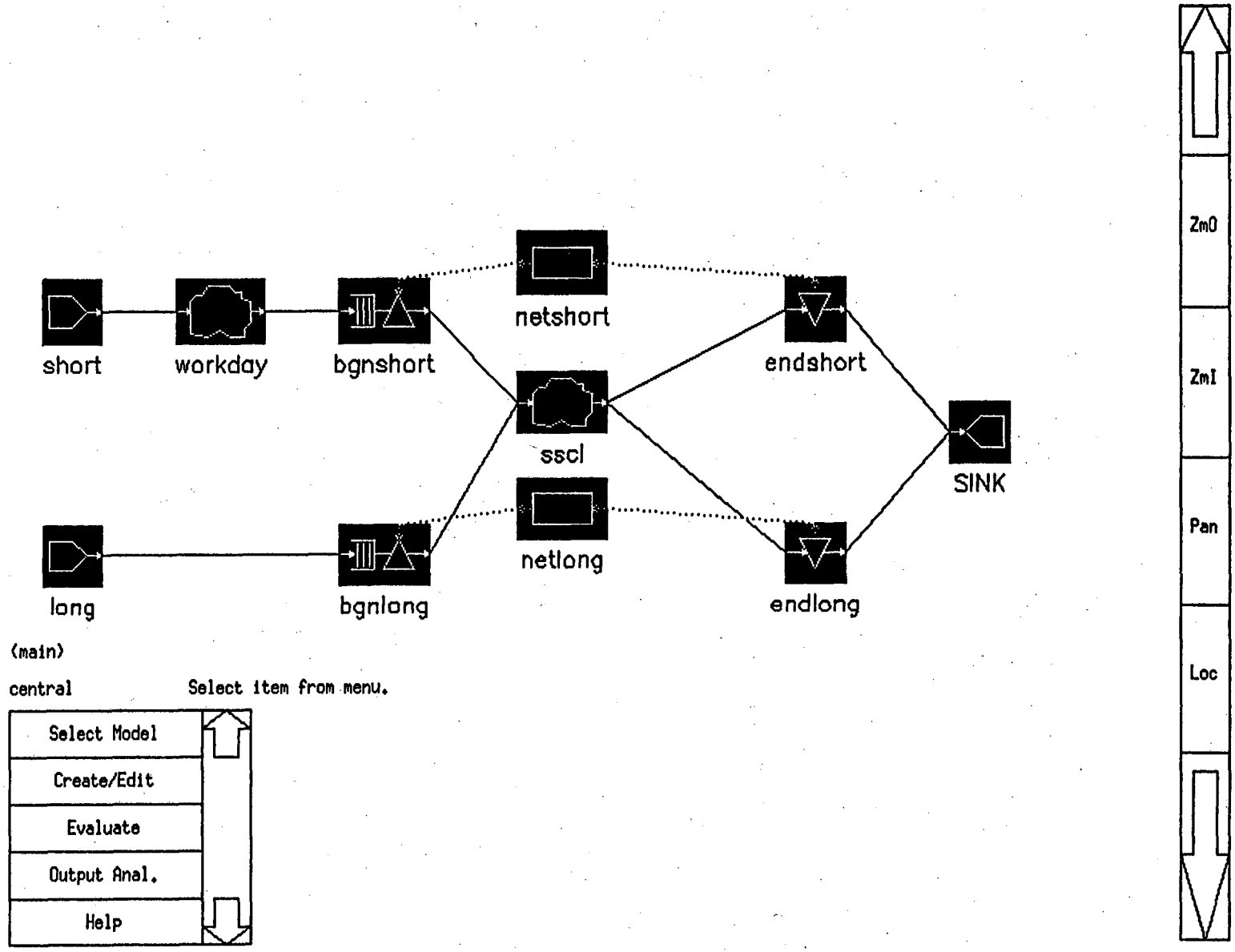
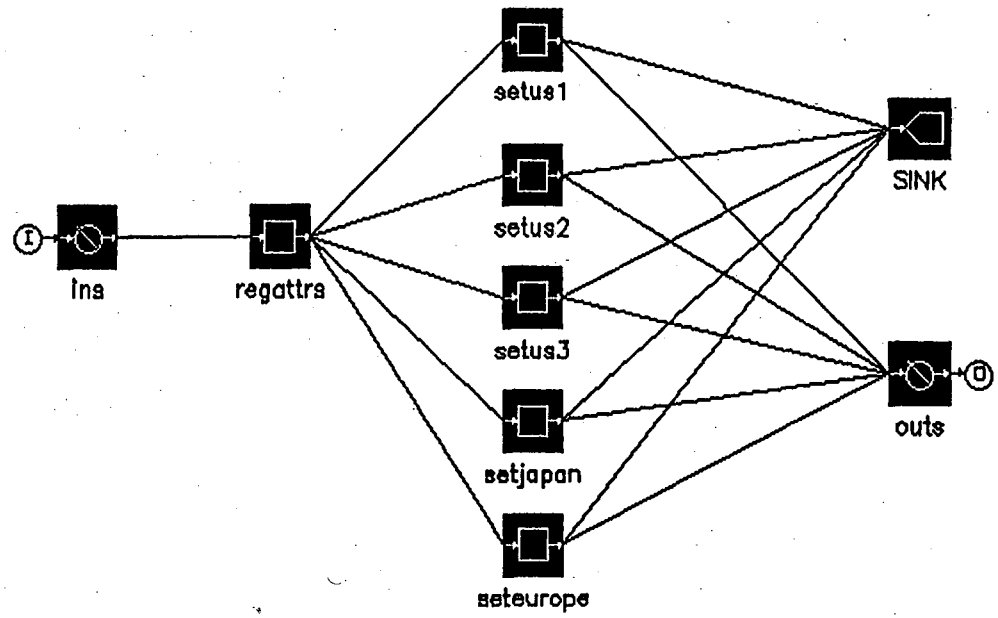
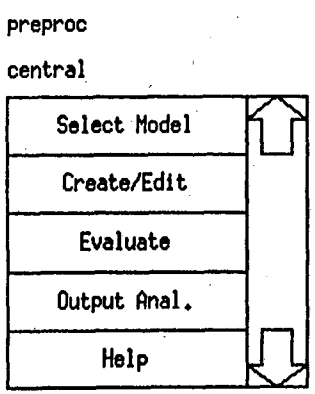

short
workday
bgnshort
netshort
sscl
netlong
endshort
endlong
SINK
long
bgnlong
(main)
central
Select item from menu.
Select Model
Create/Edit
Evaluate
Output Anal.
Help
ZmO
ZmI
Pan
Loc

Figure 1


setus1
setus2
setus3
setjapan
seteurope
SINK
ins
regattrs
outs


preproc
central
Select Model
Create/Edit
Evaluate
Output Anal.
Help

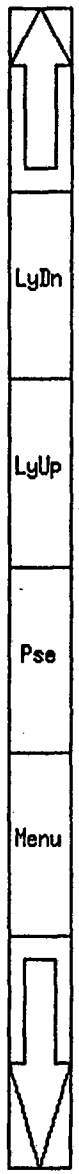

LyDn
LyUp
Pse
Menu

Figure 2

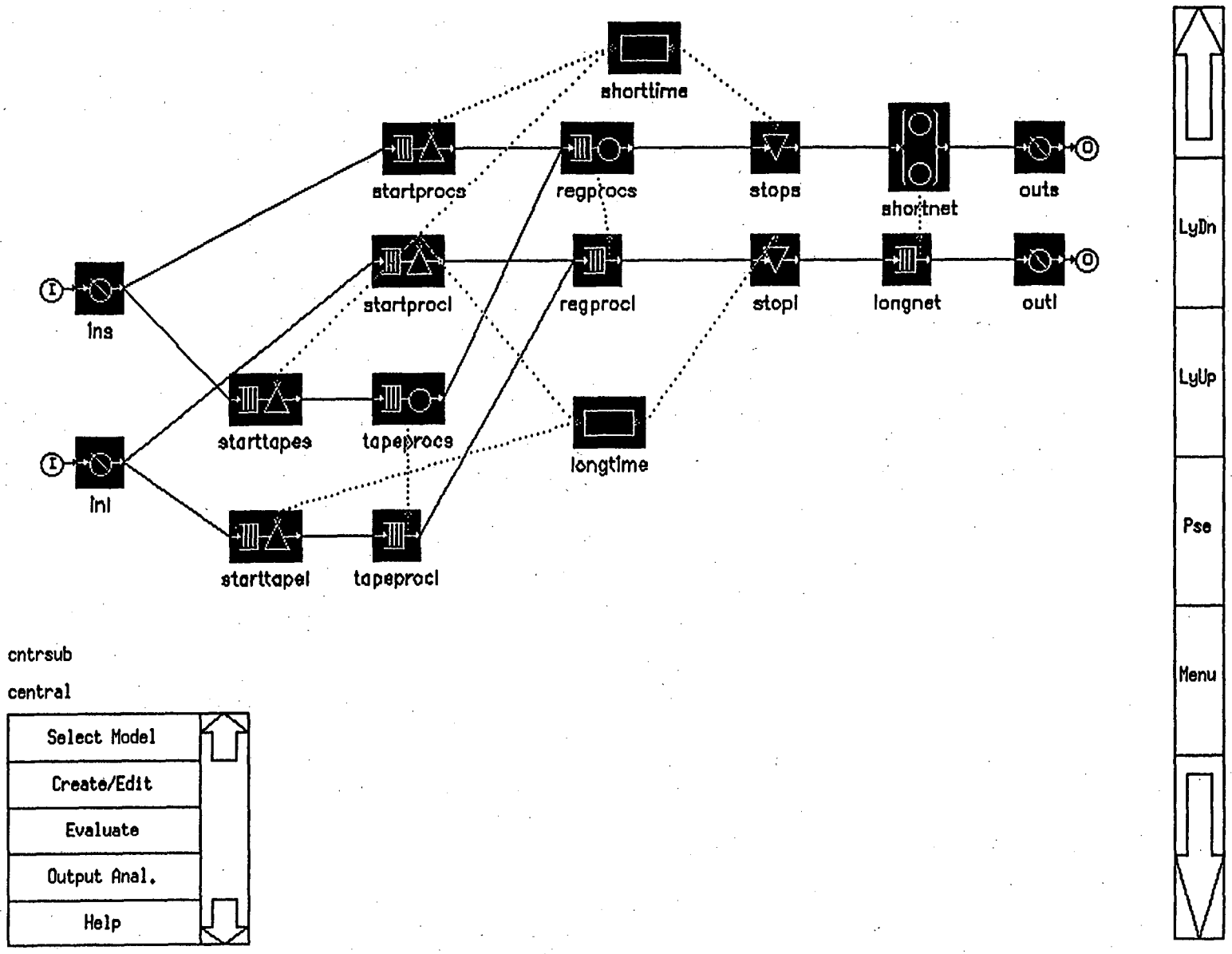

shorttime
startprocs
regprocs
stops
shortnet
outs
ins
startprocl
regprocl
stopl
longnet
outl
starttapes
tapeprocs
longtime
inl
starttapel
tapeprocl
cntrsub
central
Select Model
Create/Edit
Evaluate
Output Anal.
Help
LyDn
LyUp
Pse
Menu

Figure 3

```
/* ************************************************************************** */
MODEL:central  /* Time unit is minute */
   METHOD:SIMULATION
   NUMERIC PARAMETER:clonghit  /* Prob. that long job data is on disk */
   NUMERIC PARAMETER:cshorthit /* Prob. that short job data is on disk */
   NUMERIC PARAMETER:clproctime /* Processing time for long job */
   NUMERIC PARAMETER:csproctime /* Processing time for short job */
   NUMERIC PARAMETER:cltapetime /* Time to fetch tape data for long job */
   NUMERIC PARAMETER:cstapetime /* Time to fetch tape data for short job */
   NUMERIC PARAMETER:clarrive /* Arrival time of long jobs */
   NUMERIC PARAMETER:csarrive /* Arrival time of short jobs */
   NUMERIC PARAMETER:cservers /* Number of process servers in center */
   NUMERIC PARAMETER:cljobs /* Limit on simultaneous long jobs */
   NUMERIC PARAMETER:esltime /* Network xfr time for long job data */
   NUMERIC PARAMETER:esstime /* Network xfr time for short job data */
   NUMERIC PARAMETER:esnetrate /* Network interface relative speed */
   NUMERIC PARAMETER:ctapesvrs /* Number of tape servers in center */
   NUMERIC IDENTIFIER:us1 us2 us3 japan europe
   US1:1
   US2:2
   US3:3
   JAPAN:4
   EUROPE:5
   GLOBAL VARIABLE:clock
   CLOCK:0
QUEUE:netshort
   TYPE: PASSIVE
   TOKENS:100000
   DSPL:FCFS
   ALLOCATE NODE LIST:bgnshort
      NUMBER OF TOKENS TO ALLOCATE: 1
   RELEASE NODE LIST:endshort
QUEUE:netlong
   TYPE: PASSIVE
   TOKENS:100000
   DSPL:FCFS
   ALLOCATE NODE LIST:bgnlong
      NUMBER OF TOKENS TO ALLOCATE: 1
   RELEASE NODE LIST:endlong
/* ************************************************************************** */
SUBMODEL:cntrsub
   NUMERIC PARAMETERS:longhit shorthit
   NUMERIC PARAMETERS:proclong procshort
   NUMERIC PARAMETERS:tapelong tapeshort
   NUMERIC PARAMETERS:tapesvrs procsvrs
   NUMERIC PARAMETERS:xfrlong xfrshort
   NUMERIC PARAMETERS:netspeed
   CHAIN PARAMETERS:chs
   CHAIN PARAMETERS:chl
QUEUE:regprocq
TYPE:ACTIVE
SERVERS:procsvrs
DSPL:PS
CLASS LIST:regprocs
   WORK DEMANDS:constant(procshort)
CLASS LIST:regprocl
   WORK DEMANDS:constant(proclong)
SERVER-
   RATES: 1
   ACCEPTS: all
QUEUE:tapeq
TYPE:ACTIVE
SERVERS:tapesvrs
DSPL:FCFS
CLASS LIST:tapeprocs
   WORK DEMANDS:constant(tapeshort)
```

Listing 1

```
CLASS LIST:tapeprocl
   WORK DEMANDS:constant(tapelong)
SERVER-
   RATES: 1
   ACCEPTS: all
QUEUE:networkq
TYPE: IS
CLASS LIST:shortnet
   WORK DEMANDS:constant(xfrshort/netspeed)
CLASS LIST:longnet
   WORK DEMANDS:constant(xfrlong/netspeed)
QUEUE:shorttime
   TYPE: PASSIVE
   TOKENS:1000000
   DSPL:FCFS
   ALLOCATE NODE LIST:startprocs
      NUMBER OF TOKENS TO ALLOCATE: 1
   ALLOCATE NODE LIST:starttapes
      NUMBER OF TOKENS TO ALLOCATE: 1
   RELEASE NODE LIST:stops
QUEUE:longtime
   TYPE: PASSIVE
   TOKENS:cljobs
   DSPL:FCFS
   ALLOCATE NODE LIST:startprocl
      NUMBER OF TOKENS TO ALLOCATE: 1
   ALLOCATE NODE LIST:starttapel
      NUMBER OF TOKENS TO ALLOCATE: 1
   RELEASE NODE LIST:stopl
DUMMY NODE:ins
DUMMY NODE:outs
DUMMY NODE:inl
DUMMY NODE:outl
CHAIN:chs
   TYPE:EXTERNAL
   INPUT:ins
   OUTPUT:outs
   :ins -> startprocs ;shorthit
   :startprocs -> regprocs ;1.0
   :regprocs -> stops ;1.0
   :ins -> starttapes ;(1-shorthit)
   :starttapes -> tapeprocs ;1.0
   :tapeprocs -> regprocs ;1.0
   :stops -> shortnet ;1.0
   :shortnet -> outs ;1.0
CHAIN:chl
   TYPE:EXTERNAL
   INPUT:inl
   OUTPUT:outl
   :inl -> startprocl ;longhit
   :startprocl -> regprocl ;1.0
   :regprocl -> stopl ;1.0
   :inl -> starttapel ;(1-longhit)
   :starttapel -> tapeprocl ;1.0
   :tapeprocl -> regprocl ;1.0
   :stopl -> longnet ;1.0
   :longnet -> outl ;1.0
END OF SUBMODEL cntrsub
/* ************************************************************************** */
SUBMODEL:preproc
   CHAIN PARAMETERS:ch
SET NODE:regattrs
   ASSIGNMENT LIST:jv(0)=discrete(us1,1/cservers; ++
                                  us2,1/cservers; ++
                                  us3,1/cservers; ++
                                  japan,1/cservers; ++
```

Listing 1 (cont.)

```
                                        europe,1/cservers)
SET NODE:setus1
   ASSIGNMENT LIST:jv(1)=clock mod 1440
SET NODE:setus2
   ASSIGNMENT LIST:jv(1)=(clock-120) mod 1440
SET NODE:setus3
   ASSIGNMENT LIST:jv(1)=(clock+60) mod 1440
SET NODE:setjapan
   ASSIGNMENT LIST:jv(1)=(clock+900) mod 1440
SET NODE:seteurope
   ASSIGNMENT LIST:jv(1)=(clock+360) mod 1440
DUMMY NODE:outs
DUMMY NODE:ins
CHAIN:ch
   TYPE:EXTERNAL
   INPUT:ins
   OUTPUT:outs
   :ins -> regattrs ;1.0
   :regattrs -> setus1 ;if(jv(0)=us1)
   :regattrs -> setus2 ;if(jv(0)=us2)
   :regattrs -> setus3 ;if(jv(0)=us3)
   :regattrs -> setjapan ;if(jv(0)=japan)
   :regattrs -> seteurope ;if(jv(0)=europe)
   :setus1 -> SINK ;if(jv(1)<480 or jv(1)>1080) /*8:00am-6:00pm Central time*/
   :setus2 -> SINK ;if(jv(1)<480 or jv(1)>1080) /*8:00am-6:00pm Pacific time*/
   :setus3 -> SINK ;if(jv(1)<480 or jv(1)>1080) /*8:00am-6:00pm Eastern time*/
   :setjapan -> SINK ;if(jv(1)<480 or jv(1)>1080) /* Japan Time */
   :seteurope -> SINK ;if(jv(1)<480 or jv(1)>1080) /* Europe Time */
   :setus1 -> outs ;if(jv(1)>=480 and jv(1)<=1080)
   :setus2 -> outs ;if(jv(1)>=480 and jv(1)<=1080)
   :setus3 -> outs ;if(jv(1)>=480 and jv(1)<=1080)
   :setjapan -> outs ;if(jv(1)>=480 and jv(1)<=1080)
   :seteurope -> outs ;if(jv(1)>=480 and jv(1)<=1080)
END OF SUBMODEL preproc
INVOCATION:sscl
   TYPE:cntrsub
   LONGHIT:clonghit
   SHORTHIT:cshorthit
   PROCLONG:clproctime
   PROCSHORT:csproctime
   TAPELONG:cltapetime
   TAPESHORT:cstapetime
   TAPESVRS:ctapesvrs
   PROCSVRS:cservers
   XFRLONG:esltime
   XFRSHORT:esstime
   NETSPEED:esnetrate
   CHS:shortjobs
   CHL:longjobs
INVOCATION:workday
   TYPE:preproc
   CH:shortjobs
CHAIN:shortjobs
   TYPE:OPEN
   SOURCE LIST:short
      ARRIVAL TIMES:csarrive
   :short -> workday.ins ;1.0
   :workday.outs -> bgnshort ;1.0
   :bgnshort -> sscl.ins ;1.0
   :sscl.outs -> endshort ;1.0
   :endshort -> SINK ;1.0
CHAIN:longjobs
   TYPE:OPEN
   SOURCE LIST:long
      ARRIVAL TIMES:clarrive
   :long -> bgnlong ;1.0
```

Listing 1 (cont.)

```
   :bgnlong -> sscl.inl ;1.0
   :sscl.outl -> endlong ;1.0
   :endlong -> SINK ;1.0
CONFIDENCE INTERVAL METHOD:NONE
INITIAL STATE DEFINITION-
RUN LIMITS-
   NODES FOR DEPARTURE COUNTS:endlong
      DEPARTURES:2000
LIMIT - CP SECONDS:3600
TRACE:NO
END
```

Listing 1 (cont.)

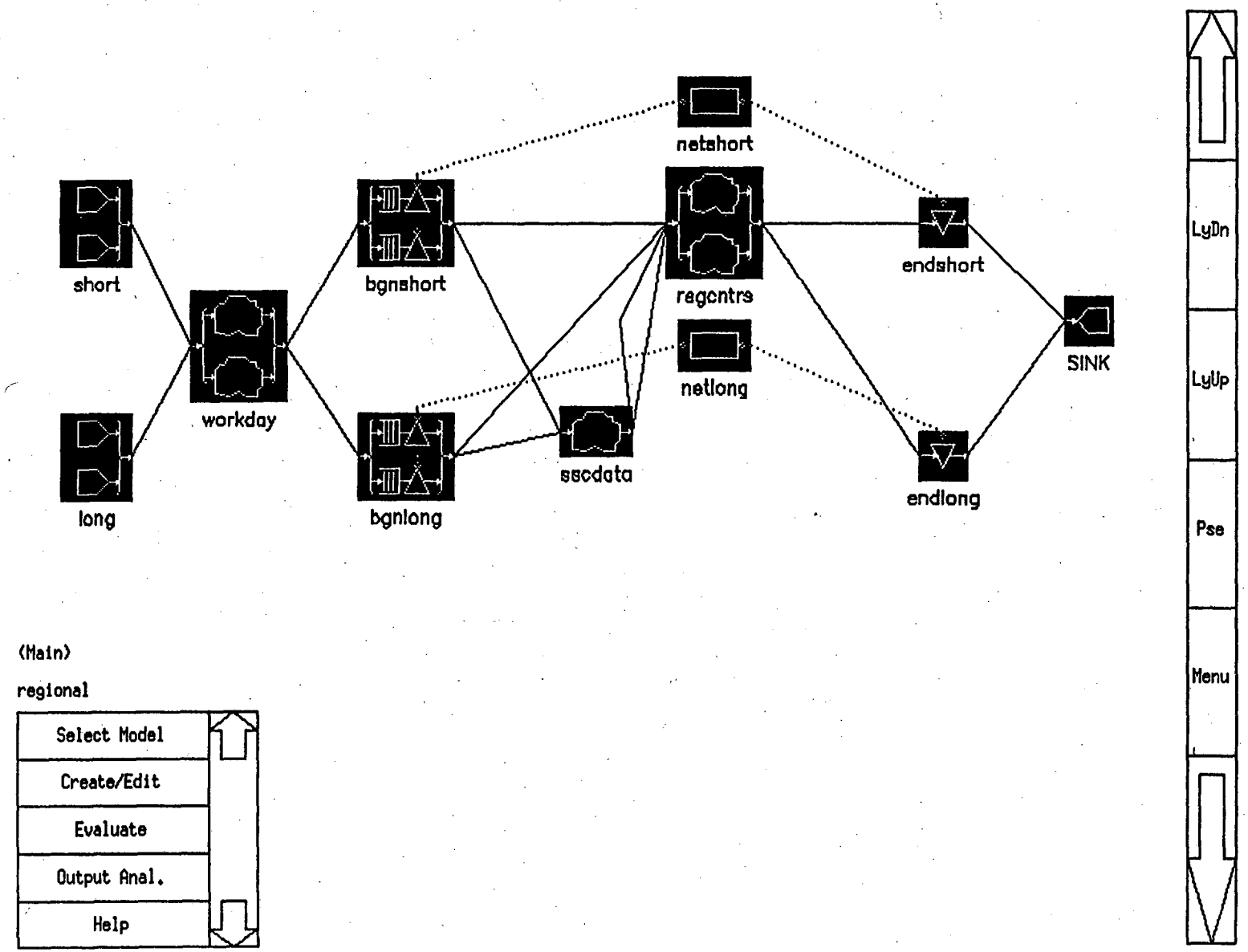

netshort
short
bgnshort
regcntrs
endshort
workday
netlong
SINK
long
bgnlong
sscdata
endlong
(Main)
regional
Select Model
Create/Edit
Evaluate
Output Anal.
Help
LyDn
LyUp
Pse
Menu

Figure 4

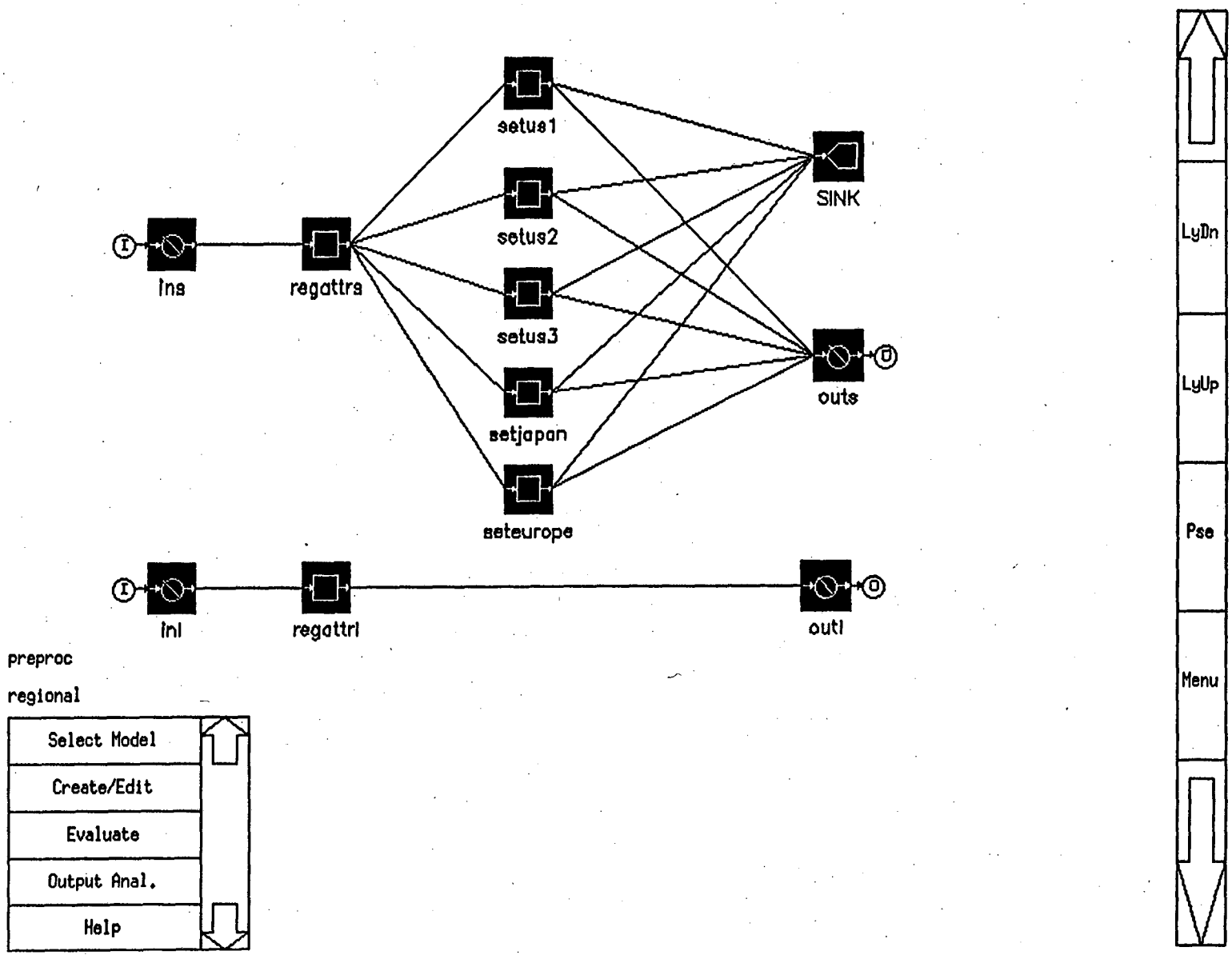

setus1
setus2
setus3
setjapan
seteurope
SINK
ins
regattrs
outs
inl
regattrl
outl
preproc
regional
Select Model
Create/Edit
Evaluate
Output Anal.
Help
LyDn
LyUp
Pse
Menu

Figure 5

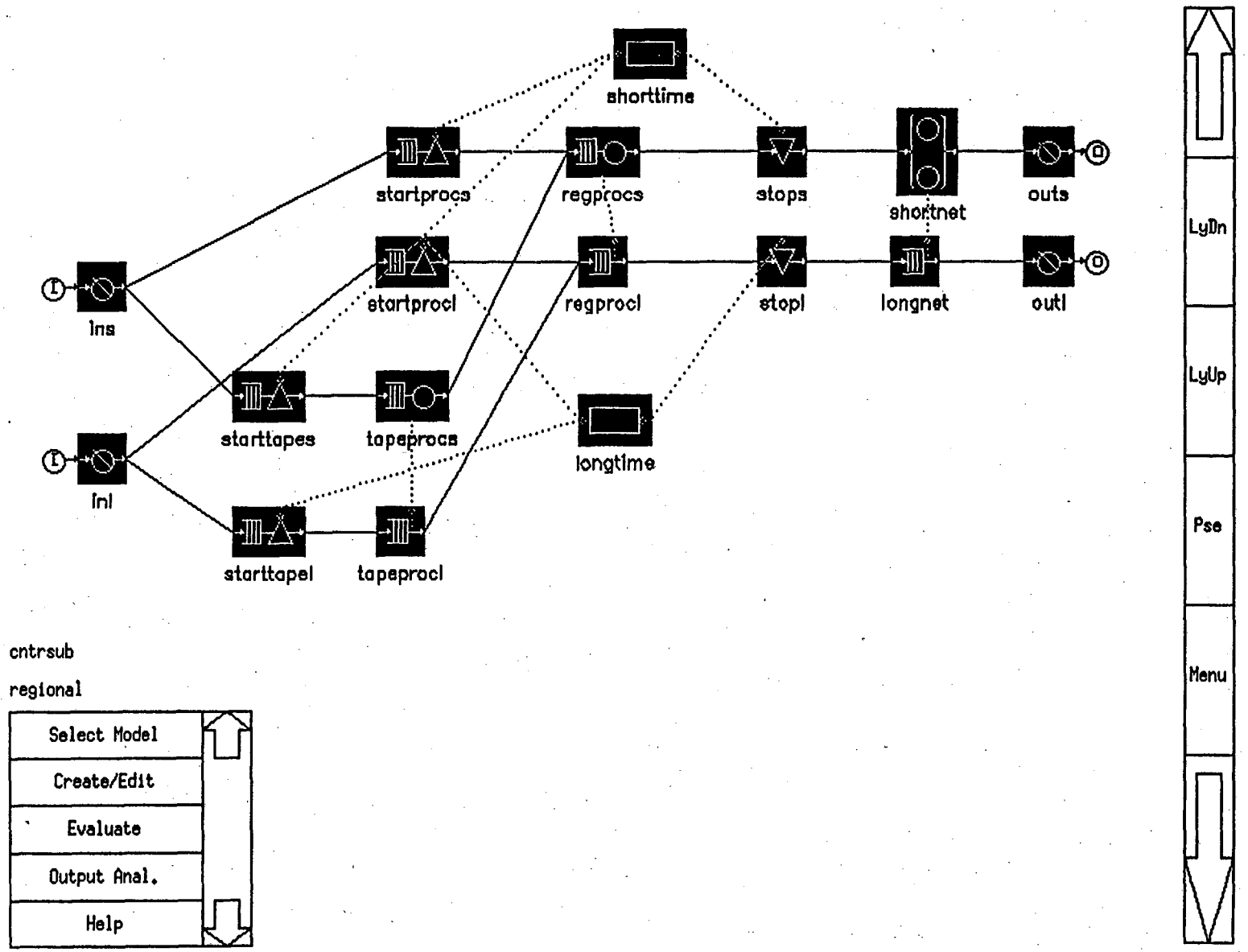

shorttime
startprocs
regprocs
stops
shortnet
outs
ins
startprocl
regprocl
stopl
longnet
outl
starttapes
tapeprocs
longtime
inl
starttapel
tapeprocl
cntrsub
regional
Select Model
Create/Edit
Evaluate
Output Anal.
Help
LyDn
LyUp
Pse
Menu

Figure 6

```
/* ************************************************************************** */
MODEL:regional /* Time unit is minute */
   METHOD:SIMULATION
   NUMERIC PARAMETER:lregprob /* Prob. regional center has long data */
   NUMERIC PARAMETER:sregprob /* Prob. regional center has short data */
   NUMERIC PARAMETER:larrive /* System-wide interarrival time for long jobs */
   NUMERIC PARAMETER:sarrive /* System-wide interarrival time for short jobs */
   NUMERIC PARAMETER:rlonghit  /* Regional prob. that long data is on disk */
   NUMERIC PARAMETER:rshorthit /* Regional prob. that short data is on disk */
   NUMERIC PARAMETER:clonghit  /* SSCL prob. that long data is on disk */
   NUMERIC PARAMETER:cshorthit /* SSCL prob. that short data is on disk */
   NUMERIC PARAMETER:rprocsrvs  /* Regional number of process servers */
   NUMERIC PARAMETER:rlproctime /* Regional process time for long jobs */
   NUMERIC PARAMETER:rsproctime /* Regional process time for short jobs */
   NUMERIC PARAMETER:rljobs     /* Regional limit on simultaneous long jobs */
   NUMERIC PARAMETER:cprocsrvs  /* SSCL number of process servers */
   NUMERIC PARAMETER:clproctime /* SSCL process time for long jobs */
   NUMERIC PARAMETER:csproctime /* SSCL process time for short jobs */
   NUMERIC PARAMETER:cljobs     /* SSCL limit on simultaneous long jobs */
   NUMERIC PARAMETER:rtapesrvs  /* Regional number of tape servers */
   NUMERIC PARAMETER:rltapetime /* Regional tape fetch time for long jobs */
   NUMERIC PARAMETER:rstapetime /* Regional tape fetch time for short jobs */
   NUMERIC PARAMETER:ctapesrvs  /* SSCL number of tape servers */
   NUMERIC PARAMETER:cltapetime /* SSCL tape fetch time for long jobs */
   NUMERIC PARAMETER:cstapetime /* SSCL tape fetch time for short jobs */
   NUMERIC PARAMETER:lxfrtime /* Network transfer time for long job data */
   NUMERIC PARAMETER:sxfrtime /* Network transfer time for short job data */
   NUMERIC PARAMETER:rnetrate /* Regional network interface relative speed */
   NUMERIC PARAMETER:cnetrate /* SSCL network interface relative speed */
   NUMERIC IDENTIFIER:nregcntrs /* Number of regional centers */
   NREGCNTRS:5
   NUMERIC IDENTIFIER:sscl us1 us2 us3 japan europe /* Processing centers */
   SSCL:0
   US1:1
   US2:2
   US3:3
   JAPAN:4
   EUROPE:5
   GLOBAL VARIABLE:clock
   CLOCK:0
   NODE ARRAY:short(nregcntrs)
   NODE ARRAY:long(nregcntrs)
   NODE ARRAY:bgnshort(nregcntrs)
   NODE ARRAY:bgnlong(nregcntrs)
QUEUE:netshort
   TYPE: PASSIVE
   TOKENS: 100000
   DSPL:FCFS
   ALLOCATE NODE LIST:bgnshort(*)
      NUMBER OF TOKENS TO ALLOCATE: 1
   RELEASE NODE LIST:endshort
QUEUE:netlong
   TYPE: PASSIVE
   TOKENS: 100000
   DSPL:FCFS
   ALLOCATE NODE LIST:bgnlong(*)
      NUMBER OF TOKENS TO ALLOCATE: 1
   RELEASE NODE LIST:endlong
/* ************************************************************************** */
SUBMODEL:preproc
   NUMERIC PARAMETERS:reg_cntr
   CHAIN PARAMETERS:chs chl
SET NODE:regattrs
   ASSIGNMENT LIST:jv(0)=reg_cntr
SET NODE:regattrl
   ASSIGNMENT LIST:jv(0)=reg_cntr ++
```

Listing 2

```
                    jv(1)=0
SET NODE:setus1
   ASSIGNMENT LIST:jv(1)=clock mod 1440
SET NODE:setus2
   ASSIGNMENT LIST:jv(1)=(clock-120) mod 1440
SET NODE:setus3
   ASSIGNMENT LIST:jv(1)=(clock+60) mod 1440
SET NODE:setjapan
   ASSIGNMENT LIST:jv(1)=(clock+900) mod 1440
SET NODE:seteurope
   ASSIGNMENT LIST:jv(1)=(clock+360) mod 1440
DUMMY NODE:outs
DUMMY NODE:outl
DUMMY NODE:ins
DUMMY NODE:inl
CHAIN:chs
   TYPE:EXTERNAL
   INPUT:ins
   OUTPUT:outs
   :ins -> regattrs ;1.0
   :regattrs -> setus1 ;if(jv(0)=us1)
   :regattrs -> setus2 ;if(jv(0)=us2)
   :regattrs -> setus3 ;if(jv(0)=us3)
   :regattrs -> setjapan ;if(jv(0)=japan)
   :regattrs -> seteurope ;if(jv(0)=europe)
   :setus1 -> SINK ;if(jv(1)<480 or jv(1)>1080) /*8:00am-6:00pm Central time*/
   :setus2 -> SINK ;if(jv(1)<480 or jv(1)>1080) /*8:00am-6:00pm Pacific time*/
   :setus3 -> SINK ;if(jv(1)<480 or jv(1)>1080) /*8:00am-6:00pm Eastern time*/
   :setjapan -> SINK ;if(jv(1)<480 or jv(1)>1080) /* Japan Time */
   :seteurope -> SINK ;if(jv(1)<480 or jv(1)>1080) /* Europe Time */
   :setus1 -> outs ;if(jv(1)>=480 and jv(1)<=1080)
   :setus2 -> outs ;if(jv(1)>=480 and jv(1)<=1080)
   :setus3 -> outs ;if(jv(1)>=480 and jv(1)<=1080)
   :setjapan -> outs ;if(jv(1)>=480 and jv(1)<=1080)
   :seteurope -> outs ;if(jv(1)>=480 and jv(1)<=1080)
CHAIN:chl
   TYPE:EXTERNAL
   INPUT:inl
   OUTPUT:outl
   :inl -> regattrl ;1.0
   :regattrl -> outl ;1.0
END OF SUBMODEL preproc
/* ************************************************************************** */
SUBMODEL:cntrsub
   NUMERIC PARAMETERS:longhit shorthit
   NUMERIC PARAMETERS:proclong procshort
   NUMERIC PARAMETERS:tapelong tapeshort
   NUMERIC PARAMETERS:tapesvrs procsvrs
   NUMERIC PARAMETERS:xfrlong xfrshort
   NUMERIC PARAMETERS:netspeed
   CHAIN PARAMETERS:chs
   CHAIN PARAMETERS:chl
QUEUE:regprocq
TYPE:ACTIVE
SERVERS:procsvrs
DSPL:PS
CLASS LIST:regprocs
   WORK DEMANDS:constant(procshort)
CLASS LIST:regprocl
   WORK DEMANDS:constant(proclong)
SERVER-
   RATES: 1
   ACCEPTS: all
QUEUE:tapeq
TYPE:ACTIVE
SERVERS:tapesvrs
```

Listing 2 (cont.)

```
DSPL:FCFS
CLASS LIST:tapeprocs
   WORK DEMANDS:constant(tapeshort)
CLASS LIST:tapeprocl
   WORK DEMANDS:constant(tapelong)
SERVER-
   RATES: 1
   ACCEPTS: all
QUEUE:networkq
TYPE: IS
CLASS LIST:shortnet
   WORK DEMANDS:constant(xfrshort/netspeed)
CLASS LIST:longnet
   WORK DEMANDS:constant(xfrlong/netspeed)
QUEUE:shorttime
   TYPE: PASSIVE
   TOKENS:1000000
   DSPL:FCFS
   ALLOCATE NODE LIST:startprocs
      NUMBER OF TOKENS TO ALLOCATE: 1
   ALLOCATE NODE LIST:starttapes
      NUMBER OF TOKENS TO ALLOCATE: 1
   RELEASE NODE LIST:stops
QUEUE:longtime
   TYPE: PASSIVE
   TOKENS:cljobs
   DSPL:FCFS
   ALLOCATE NODE LIST:startprocl
      NUMBER OF TOKENS TO ALLOCATE: 1
   ALLOCATE NODE LIST:starttapel
      NUMBER OF TOKENS TO ALLOCATE: 1
   RELEASE NODE LIST:stopl
DUMMY NODE:ins
DUMMY NODE:outs
DUMMY NODE:inl
DUMMY NODE:outl
CHAIN:chs
   TYPE:EXTERNAL
   INPUT:ins
   OUTPUT:outs
   :ins -> startprocs ;shorthit
   :startprocs -> regprocs ;1.0
   :regprocs -> stops ;1.0
   :ins -> starttapes ;(1-shorthit)
   :starttapes -> tapeprocs ;1.0
   :tapeprocs -> regprocs ;1.0
   :stops -> shortnet ;1.0
   :shortnet -> outs ;1.0
CHAIN:chl
   TYPE:EXTERNAL
   INPUT:inl
   OUTPUT:outl
   :inl -> startprocl ;longhit
   :startprocl -> regprocl ;1.0
   :regprocl -> stopl ;1.0
   :inl -> starttapel ;(1-longhit)
   :starttapel -> tapeprocl ;1.0
   :tapeprocl -> regprocl ;1.0
   :stopl -> longnet ;1.0
   :longnet -> outl ;1.0
END OF SUBMODEL cntrsub
INVOCATION:sscdata
   TYPE:cntrsub
   LONGHIT:clonghit
   SHORTHIT:cshorthit
   PROCLONG:clproctime
```

Listing 2 (cont.)

```
   PROCSHORT:csproctime
   TAPELONG:cltapetime
   TAPESHORT:cstapetime
   TAPESVRS:ctapesrvs
   PROCSVRS:cprocsrvs
   XFRLONG:lxfrtime
   XFRSHORT:sxfrtime
   NETSPEED:cnetrate
   CHS:shortjobs
   CHL:longjobs
INVOCATION:regcntrs(nregcntrs)
   TYPE:cntrsub
   LONGHIT:rlonghit
   SHORTHIT:rshorthit
   PROCLONG:rlproctime
   PROCSHORT:rsproctime
   TAPELONG:rltapetime
   TAPESHORT:rstapetime
   TAPESVRS:rtapesrvs
   PROCSVRS:rprocsrvs
   XFRLONG:lxfrtime
   XFRSHORT:sxfrtime
   NETSPEED:rnetrate
   CHS:shortjobs
   CHL:longjobs
INVOCATION:workday(nregcntrs)
   TYPE:preproc
   REG_CNTR:(do i=1 to nregcntrs by 1) i
   CHS:shortjobs
   CHL:longjobs
CHAIN:shortjobs
   TYPE:OPEN
   SOURCE LIST:short
      ARRIVAL TIMES:sarrive*nregcntrs
   :short(*) -> workday(*).ins ;1.0
   :workday(*).outs -> bgnshort(*) ;1.0
   :bgnshort(*) -> regcntrs(*).ins ;sregprob
   :bgnshort(*) -> sscdata.ins ;(1.0-sregprob)
   :sscdata.outs -> regcntrs(us1).ins ;if(jv(0)=us1)
   :sscdata.outs -> regcntrs(us2).ins ;if(jv(0)=us2)
   :sscdata.outs -> regcntrs(us3).ins ;if(jv(0)=us3)
   :sscdata.outs -> regcntrs(japan).ins ;if(jv(0)=japan)
   :sscdata.outs -> regcntrs(europe).ins ;if(jv(0)=europe)
   :regcntrs(*).outs -> endshort ;1.0
   :endshort -> SINK ;1.0
CHAIN:longjobs
   TYPE:OPEN
   SOURCE LIST:long
      ARRIVAL TIMES:larrive*nregcntrs
   :long(*) -> workday(*).inl ;1.0
   :workday(*).outl -> bgnlong(*) ;1.0
   :bgnlong(*) -> regcntrs(*).inl ;lregprob
   :bgnlong(*) -> sscdata.inl ;(1.0-lregprob)
   :sscdata.outl -> regcntrs(us1).inl ;if(jv(0)=us1)
   :sscdata.outl -> regcntrs(us2).inl ;if(jv(0)=us2)
   :sscdata.outl -> regcntrs(us3).inl ;if(jv(0)=us3)
   :sscdata.outl -> regcntrs(japan).inl ;if(jv(0)=japan)
   :sscdata.outl -> regcntrs(europe).inl ;if(jv(0)=europe)
   :regcntrs(*).outl -> endlong ;1.0
   :endlong -> SINK ;1.0
CONFIDENCE INTERVAL METHOD:NONE
INITIAL STATE DEFINITION-
RUN LIMITS-
   NODES FOR DEPARTURE COUNTS:endlong
      DEPARTURES:2000
LIMIT - CP SECONDS:3600
```

Listing 2 (cont.)

```
TRACE:NO
END
```

Listing 2 (cont.)

All Jobs at Center
Single Tape per Processor

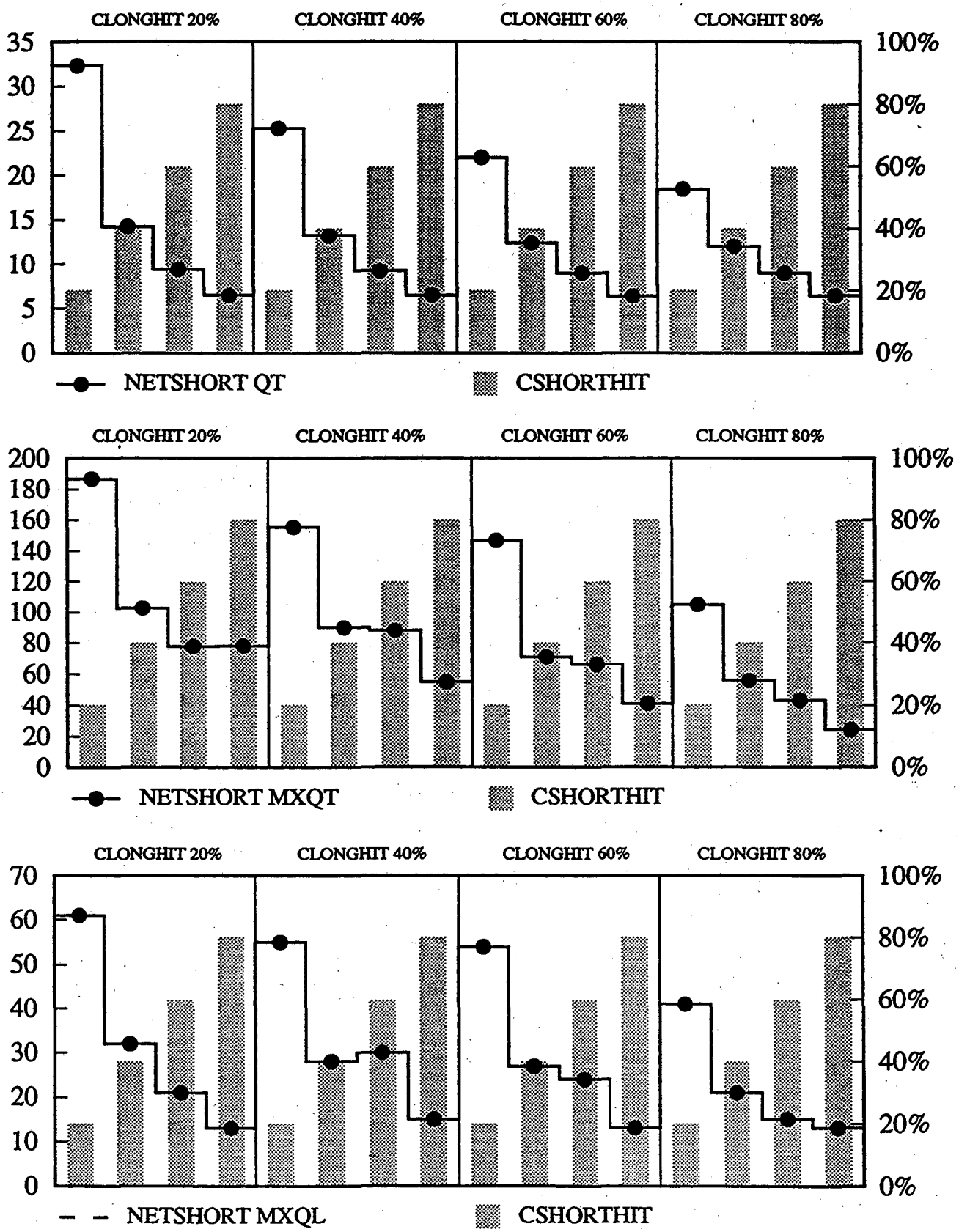


Figure 7

All Jobs at Center
Single Tape per Processor

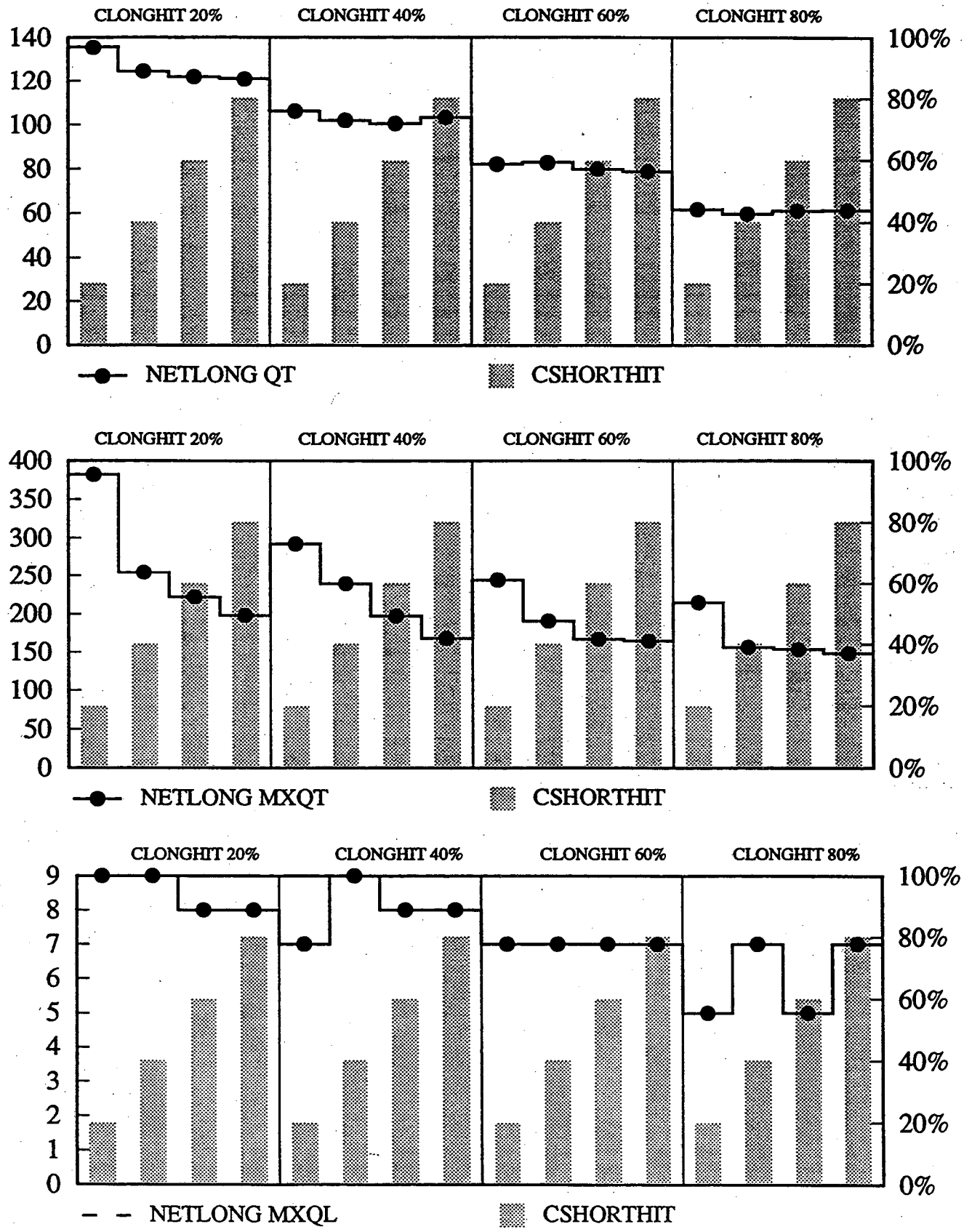


Figure 8

All Jobs at Center
Two Tapes per Processor

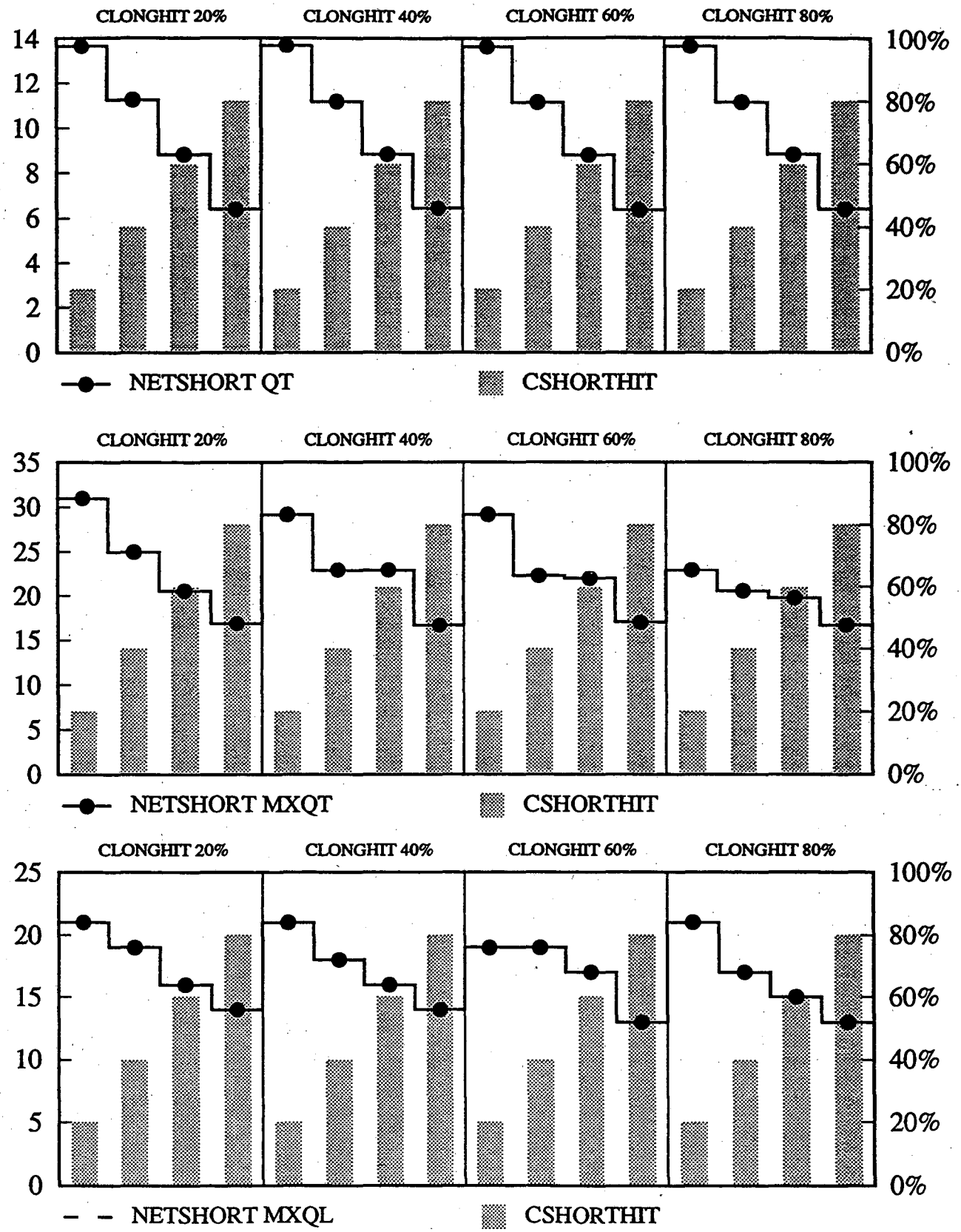


Figure 9

All Jobs at Center
Two Tapes per Processor

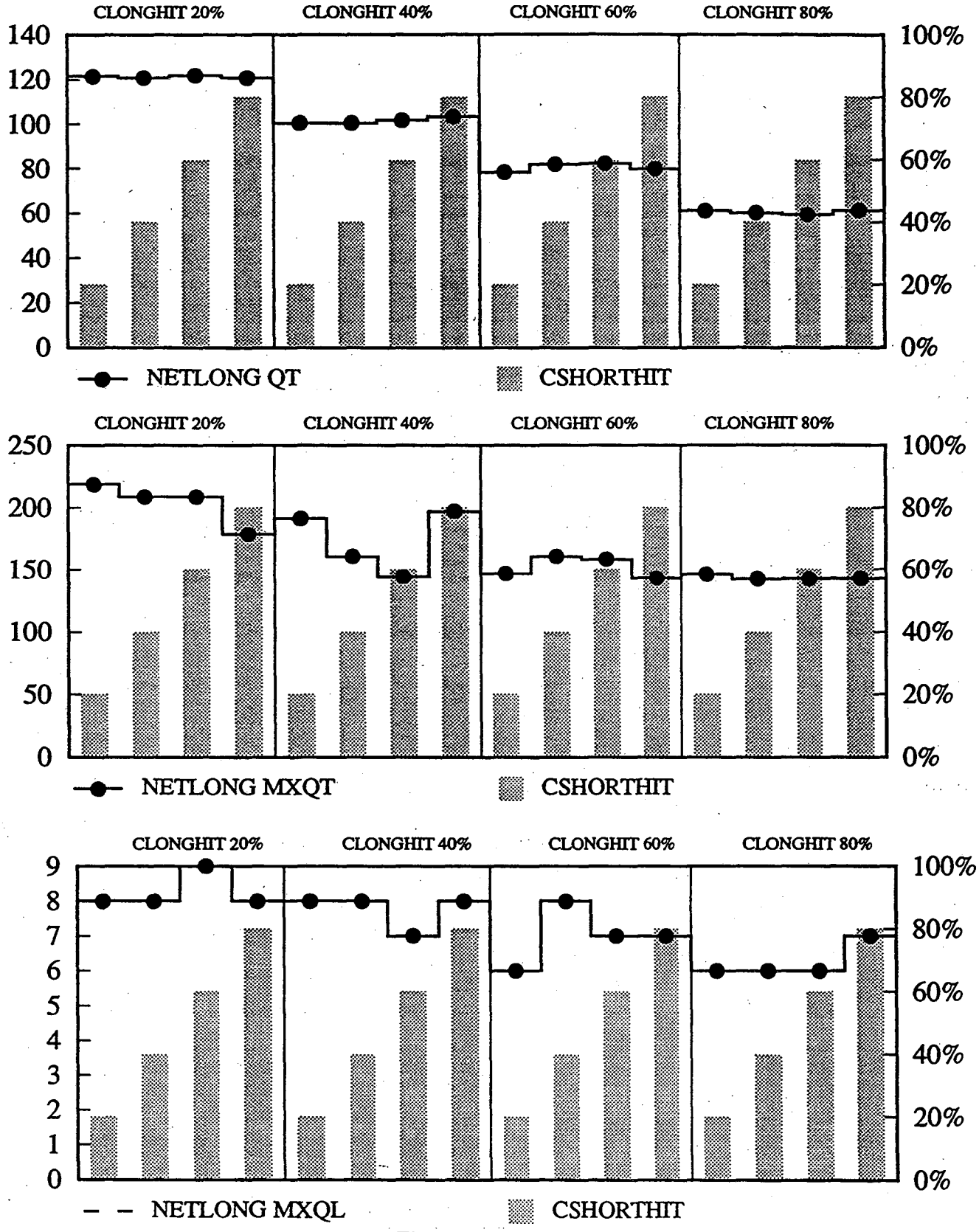


Figure 10

All Jobs at Regional
Single Tape per Processor

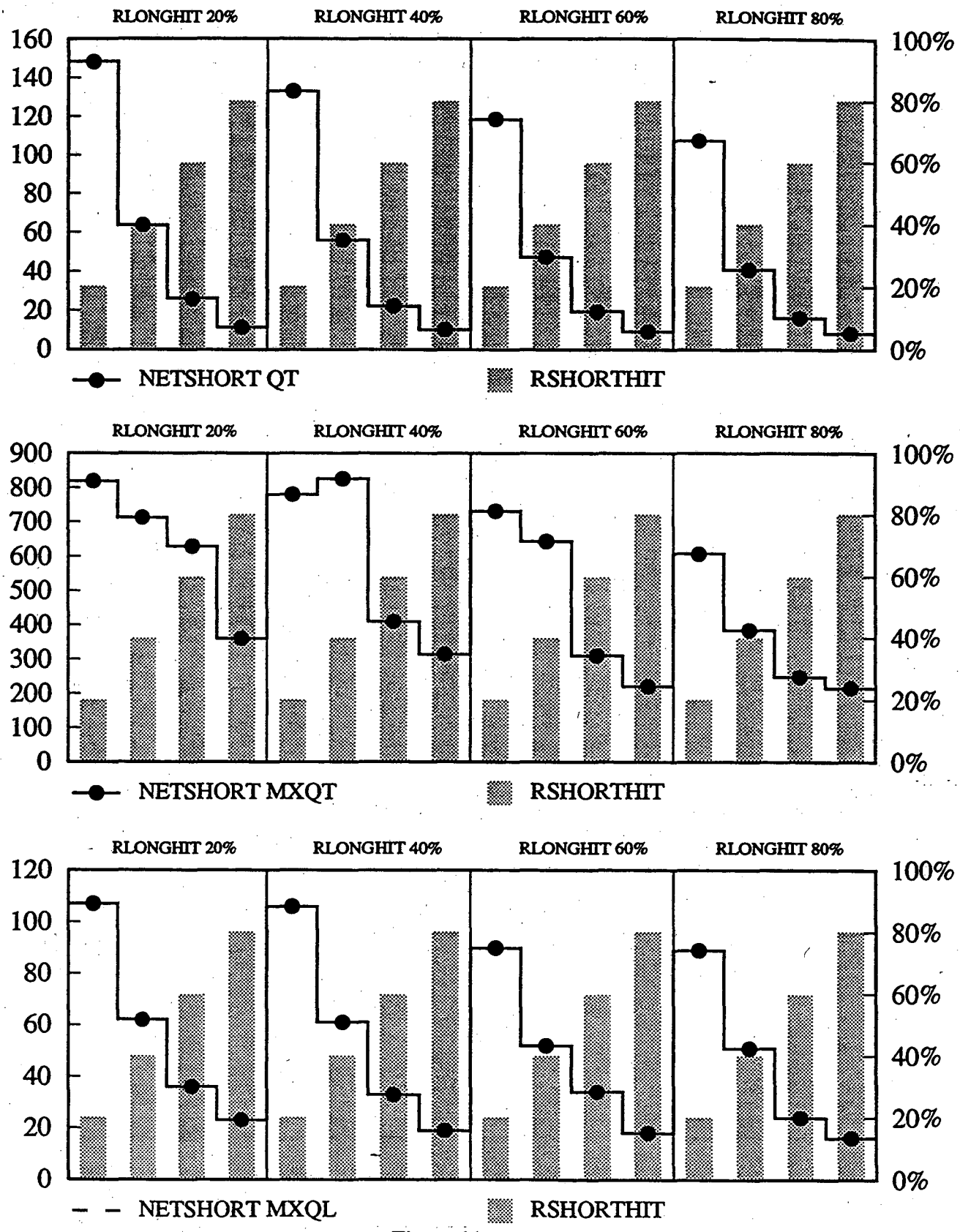


Figure 11

All Jobs at Regional
Single Tape per Processor

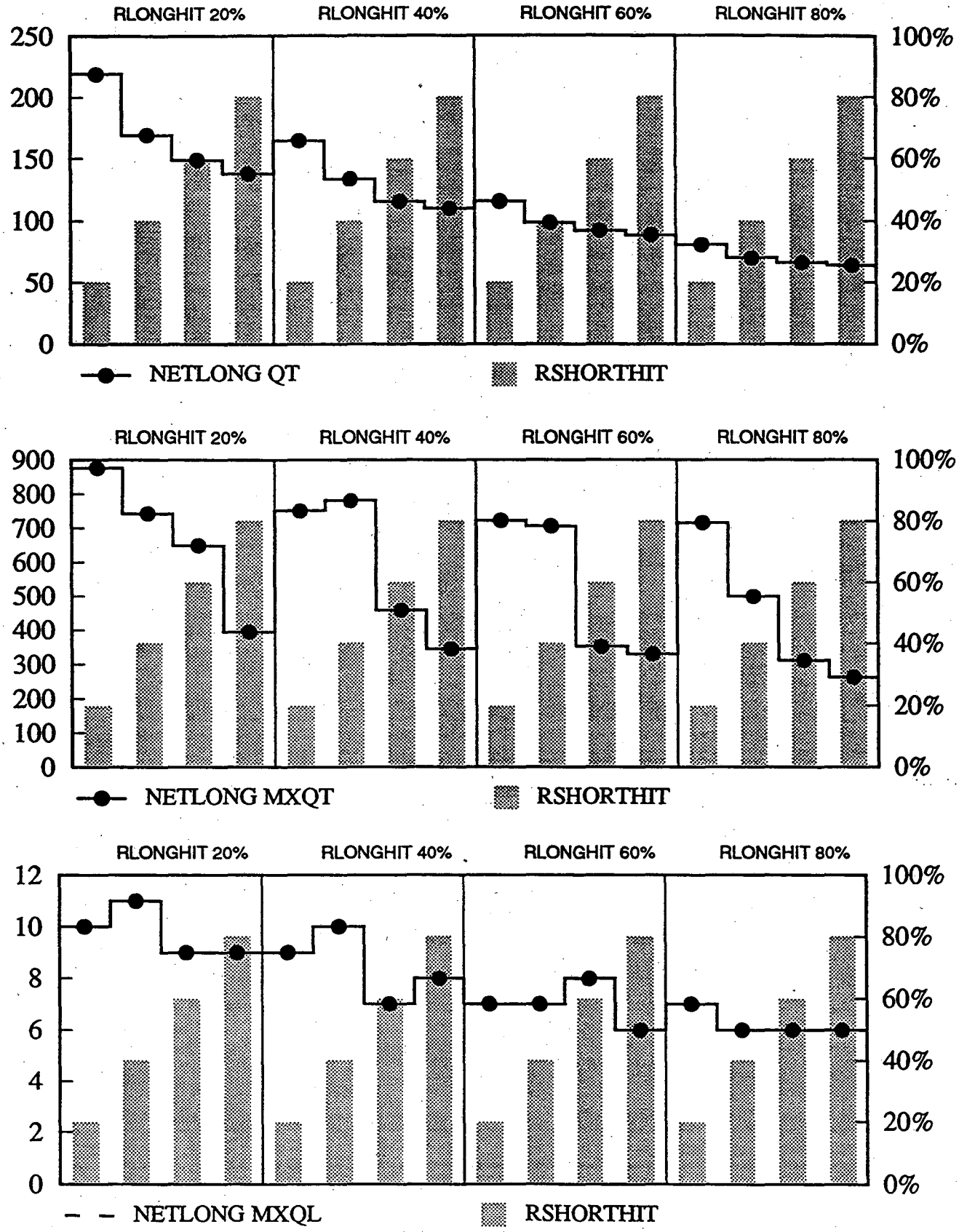


Figure 12

All Jobs at Regional
Two Tapes per Processor

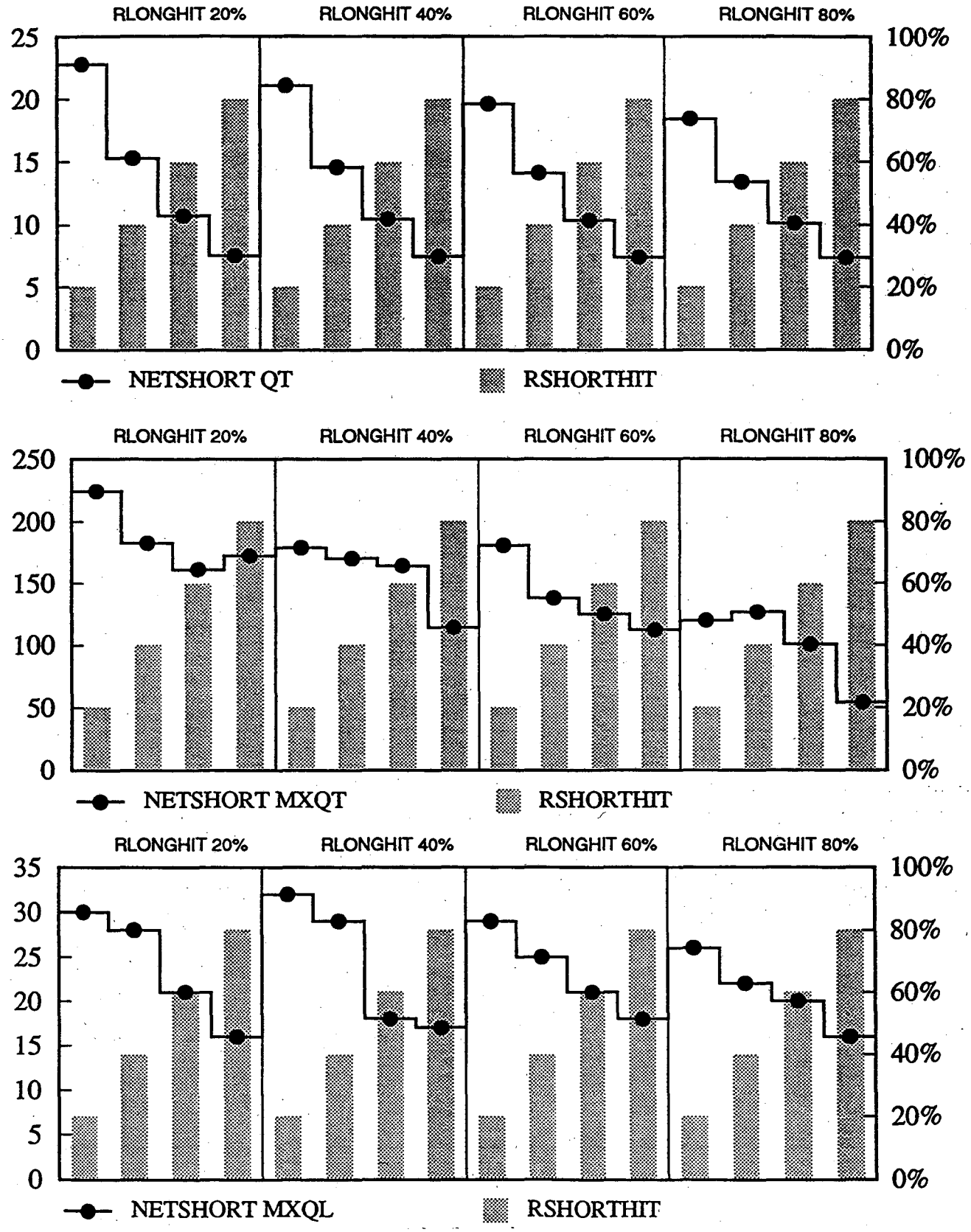


Figure 13

All Jobs at Regional
Two Tapes per Processor

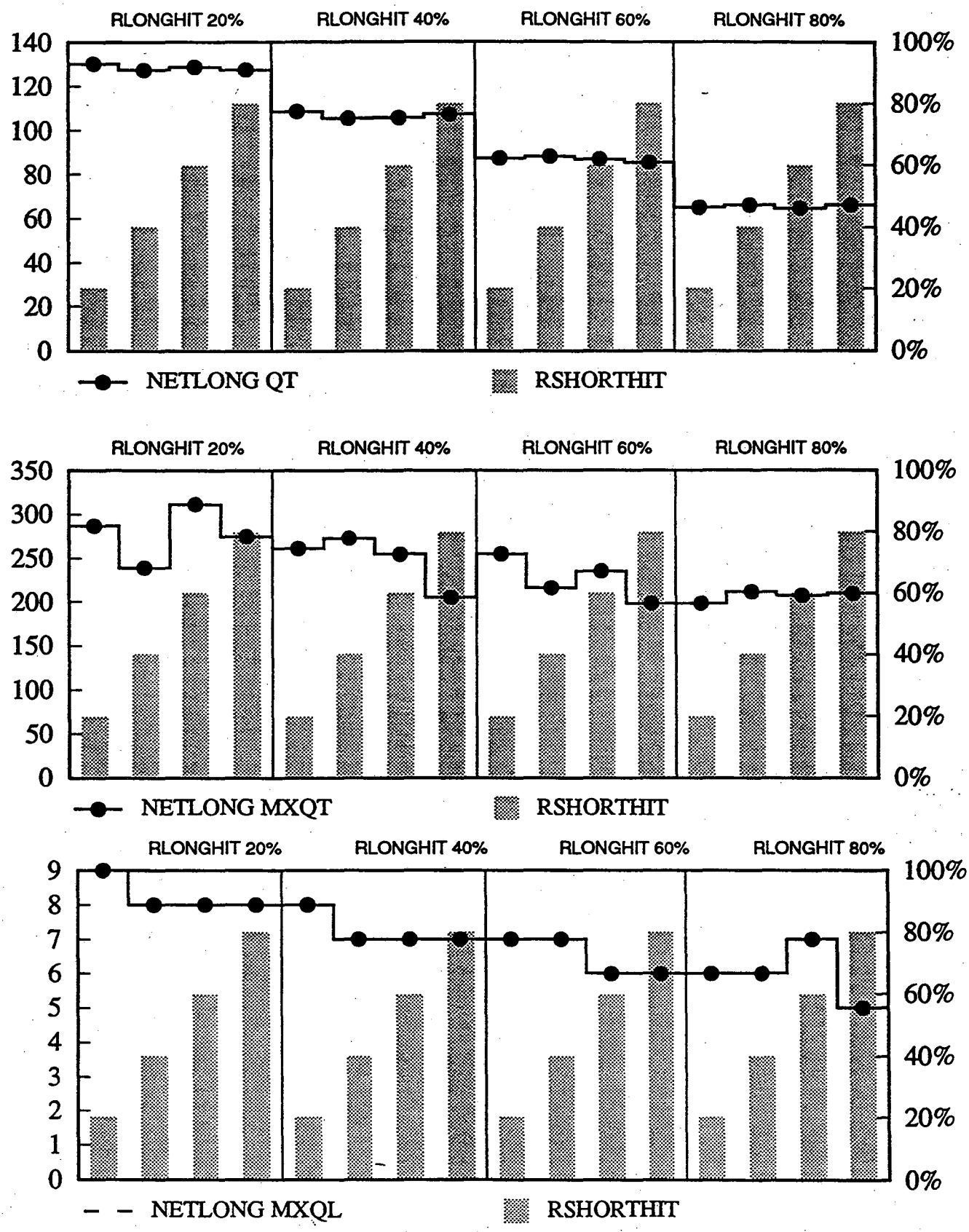
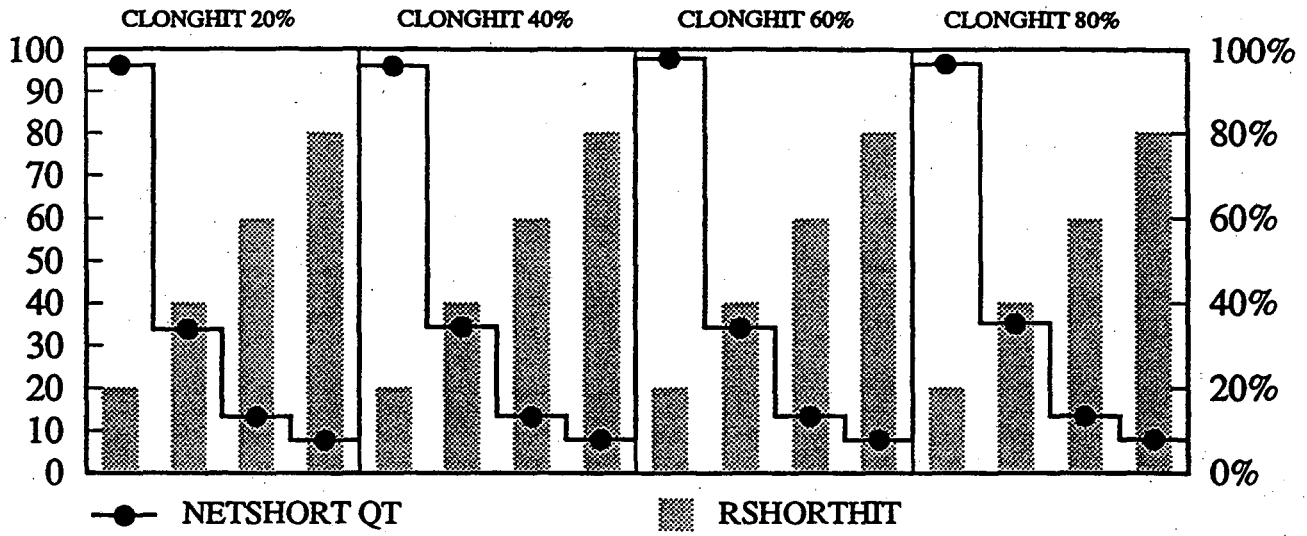
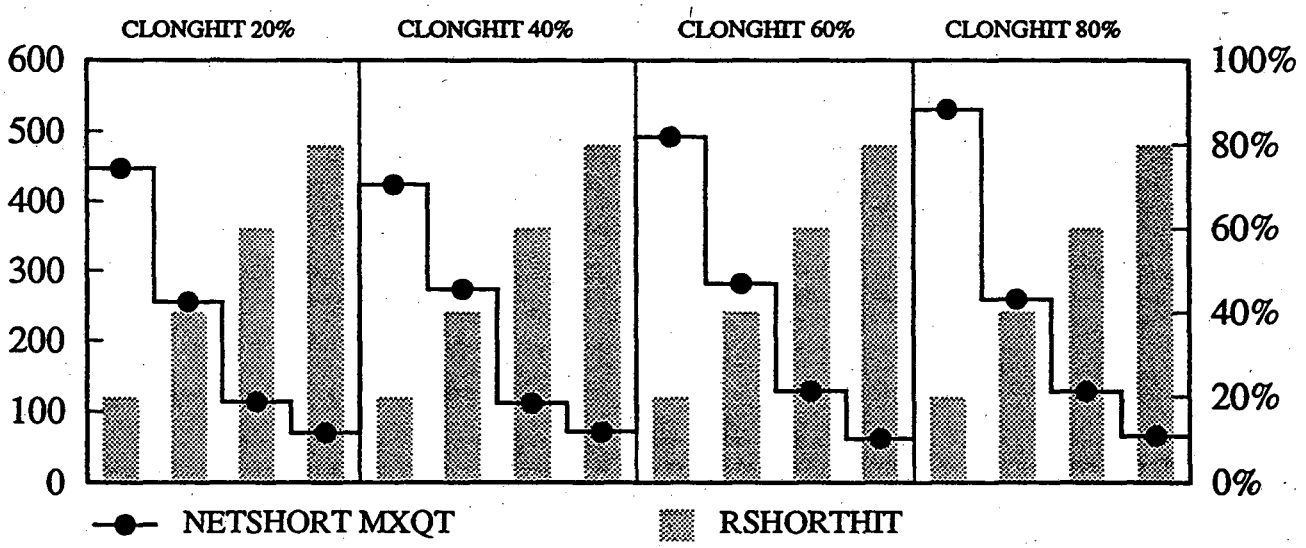


Figure 14

Short Jobs at Regional, Long at Center
Single Tape per Processor





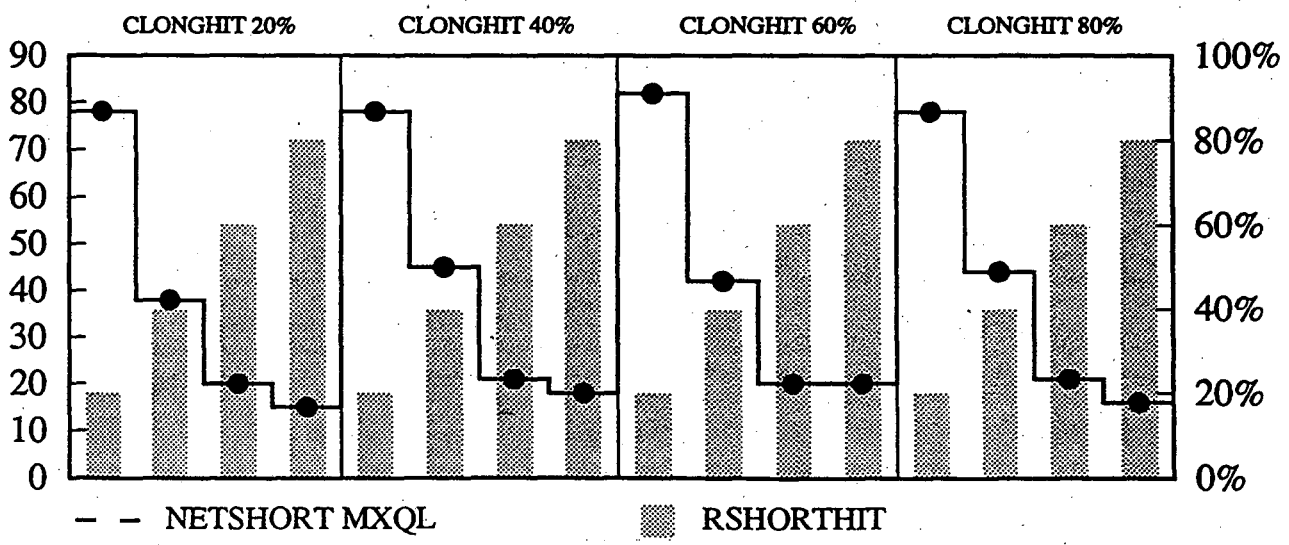


Figure 15

Short Jobs at Regional, Long at Center
Single Tape per Processor

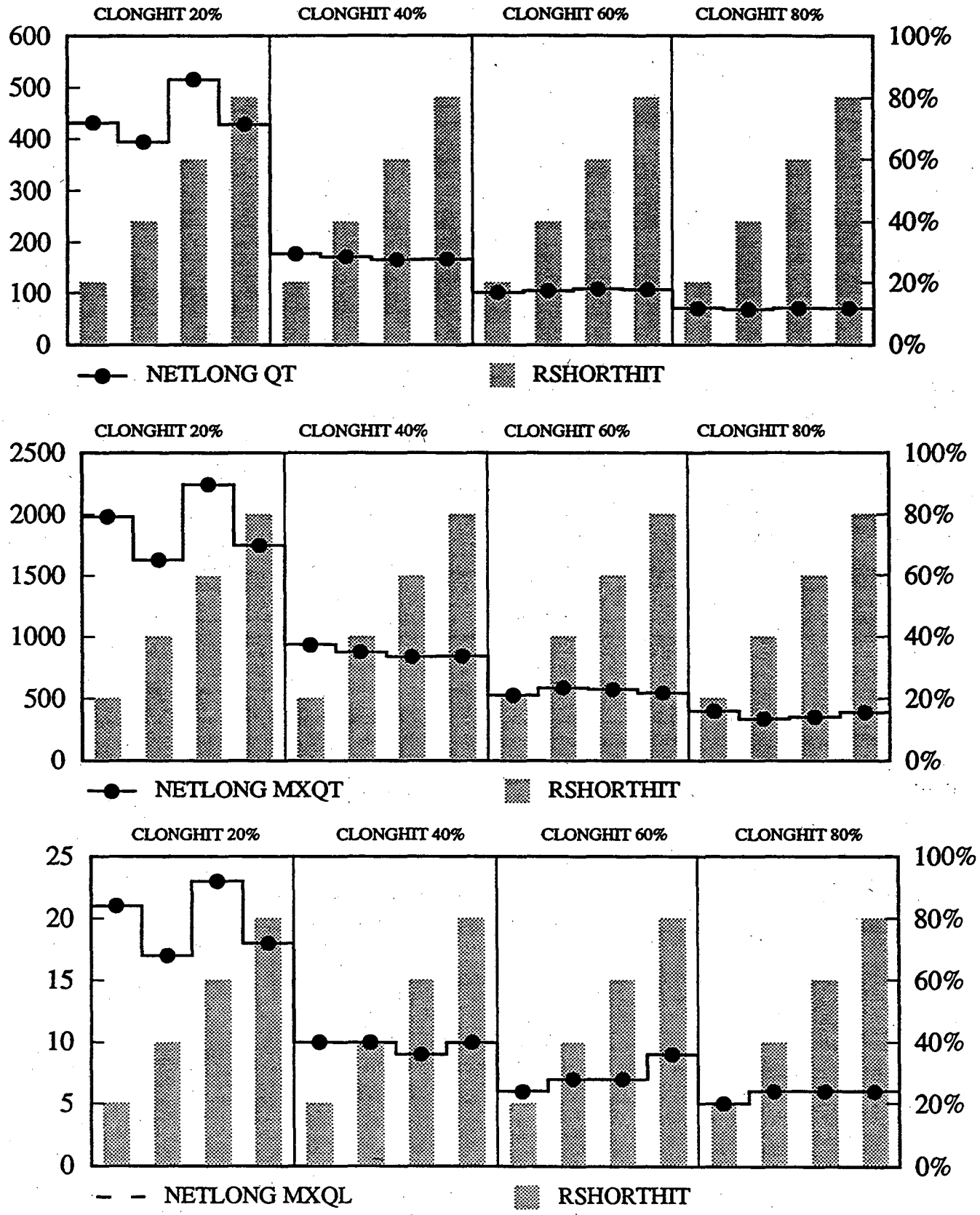


Figure 16

Short Jobs at Regional, Long at Center
Two Tapes per Processor

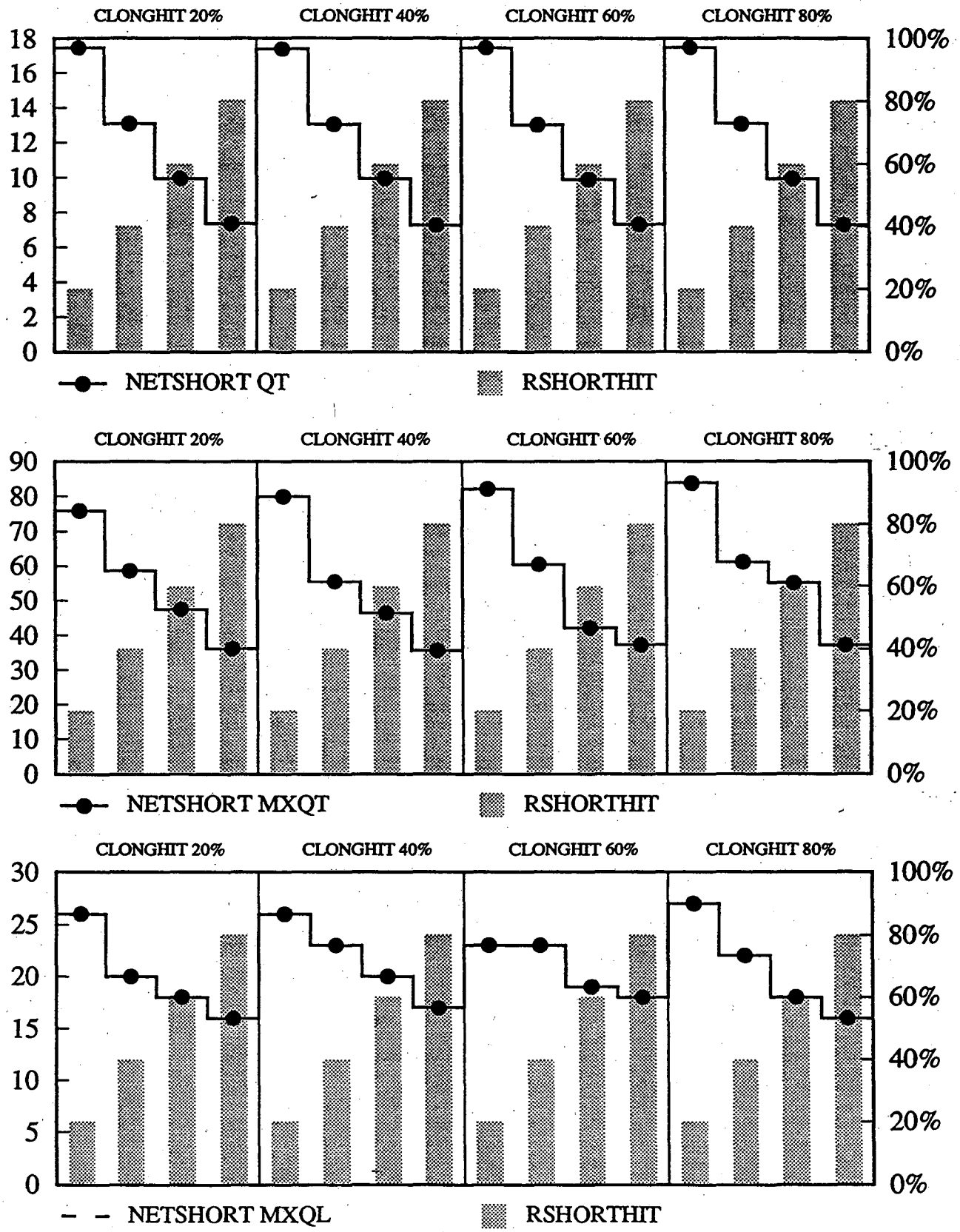


Figure 17

Short Jobs at Regional, Long at Center
Two Tapes per Processor

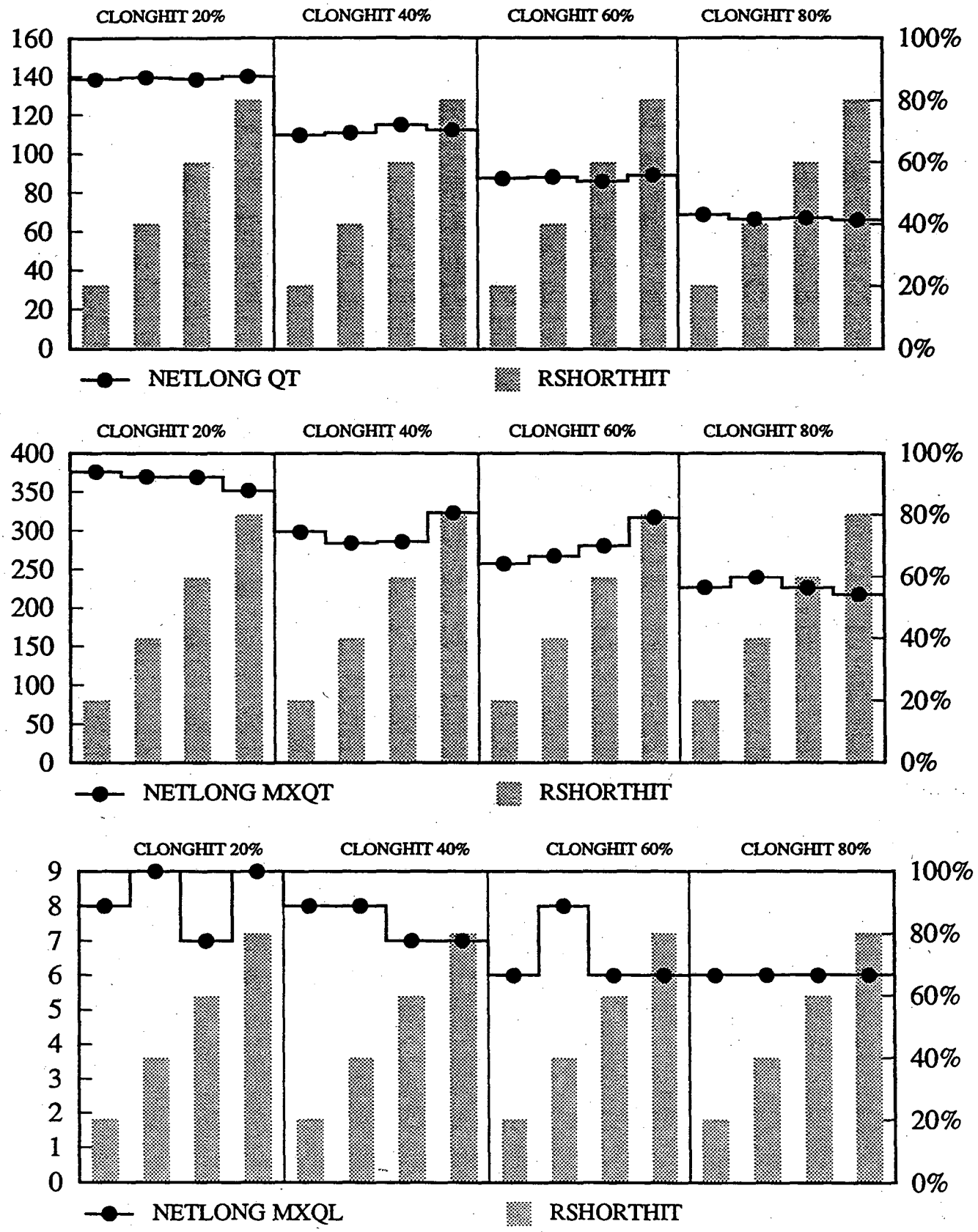


Figure 18

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720