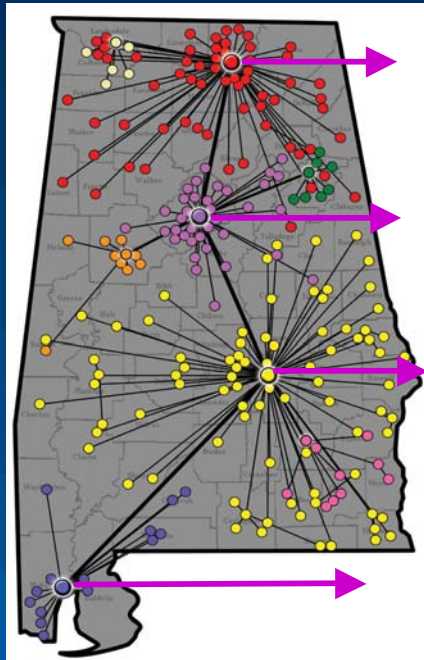
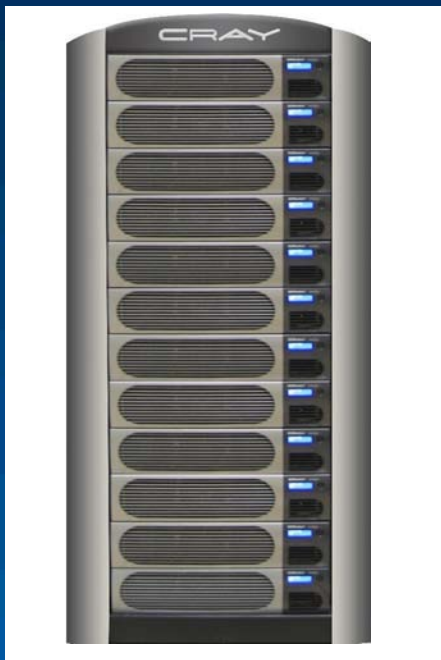


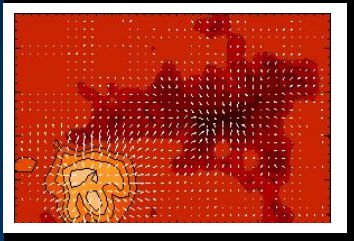


Alabama Supercomputer Center Alabama Research and Education Network

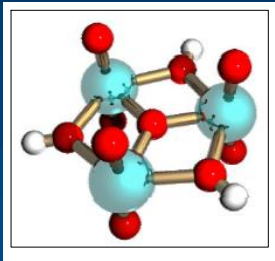




High Performance Computing



University of Alabama
in Huntsville



University of Alabama

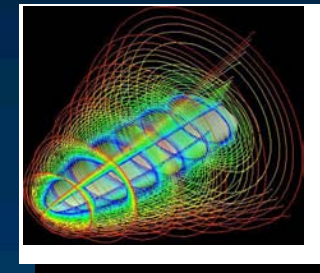


University of South
Alabama

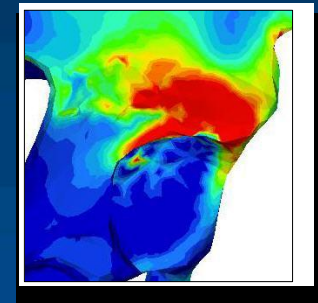


Alabama Supercomputer Center
Cummings Research Park

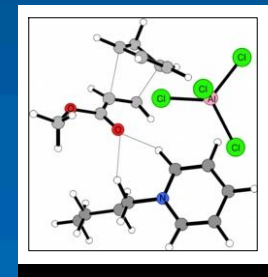
Alabama State University
Athens State University
Auburn University -Montgomery
Bevill State College
Jacksonville State University
Troy University
U.S. Air Force
U.S. Army
NASA
Intel Corporation
Operon Biotechnologies
Time Domain



Alabama A&M University



University of Alabama
at Birmingham



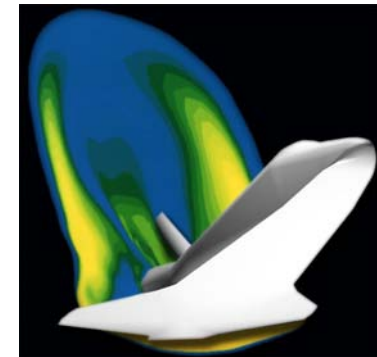
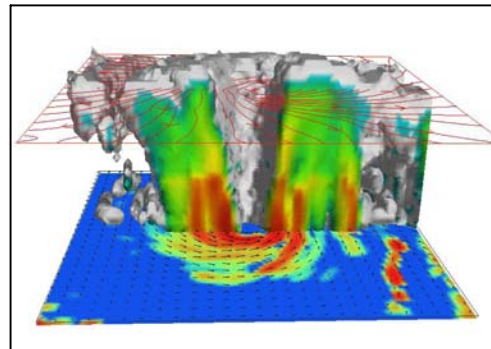
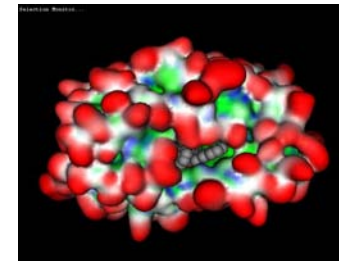
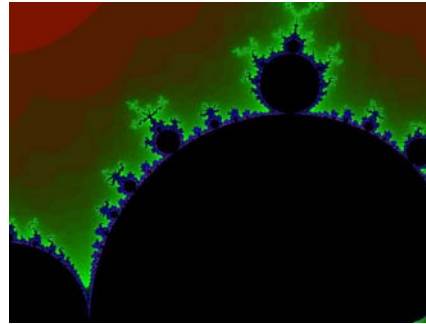
Auburn University



Alabama Supercomputer Applications



- **Materials Science**
- **Computational Fluid Dynamics**
- **Computer Science**
- **Medical**
- **Social Science**
- **Education**
- **Electromagnetics**
- **Computational Chemistry**
- **Structural Dynamics**
- **Physics**
- **Earth Science**





Compilers and Programming



Compilers

-  GNU C/C++ Fortran 77 (Altix & XD1)
-  Intel C/C++ Fortran 77/90/95 (Altix & XD1)
-  Portland Group Fortran 77/90/HP (XD1 Only)
-  Pathscale Fortran 77/90 (XD1 only)

Parallel Programming

-  OpenMP
-  MPI
-  Pthreads
-  Java threads
-  Math libraries; ACML, SLATEC, MKL, SCSL, IMSL.



Cray XD1 Supercomputer



- 144 AMD Opteron Processor System
 - 634 GFLOPS Peak
- Distributed Memory, Rapid Array Network.
- 6 Xilinx Virtex 4 LX160 FPGAs



- Memory
 - 240 Gigabytes
- Nodes with 2,4,8 Gigabytes
- Disk Storage
 - 7 Terabytes



Balanced Interconnect



GigaBytes ← → GFLOPS ← → GigaBytes per Second

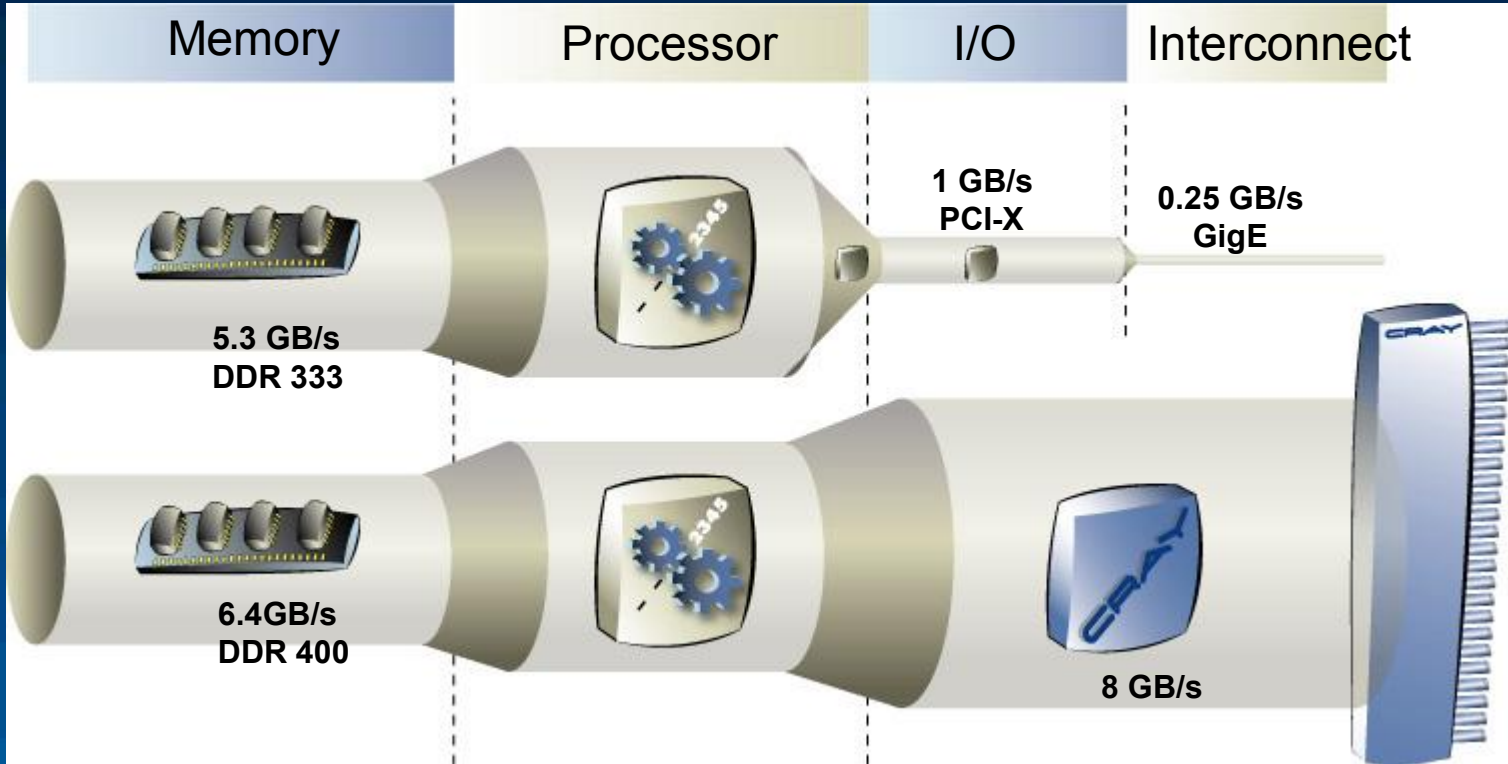
Memory

Processor

I/O

Interconnect

Xeon Server



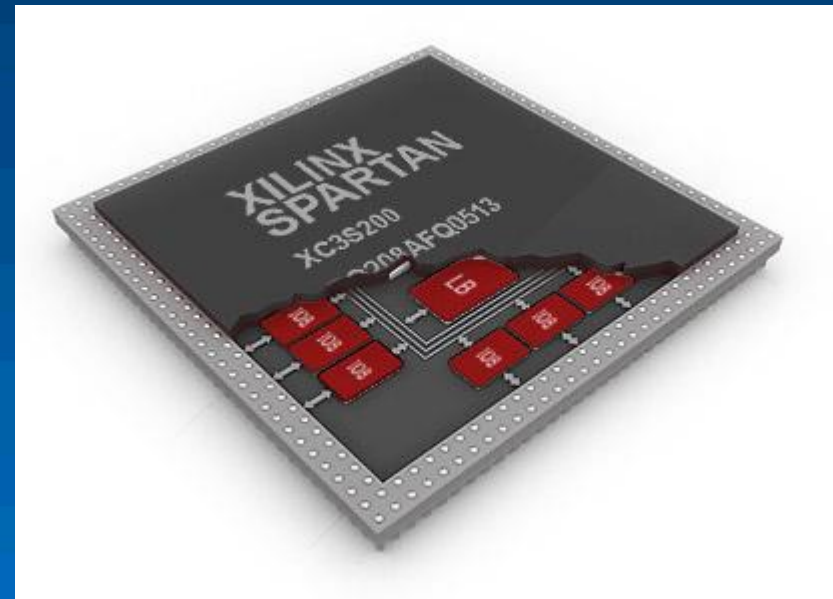
Removing the communications bottleneck



FPGAs

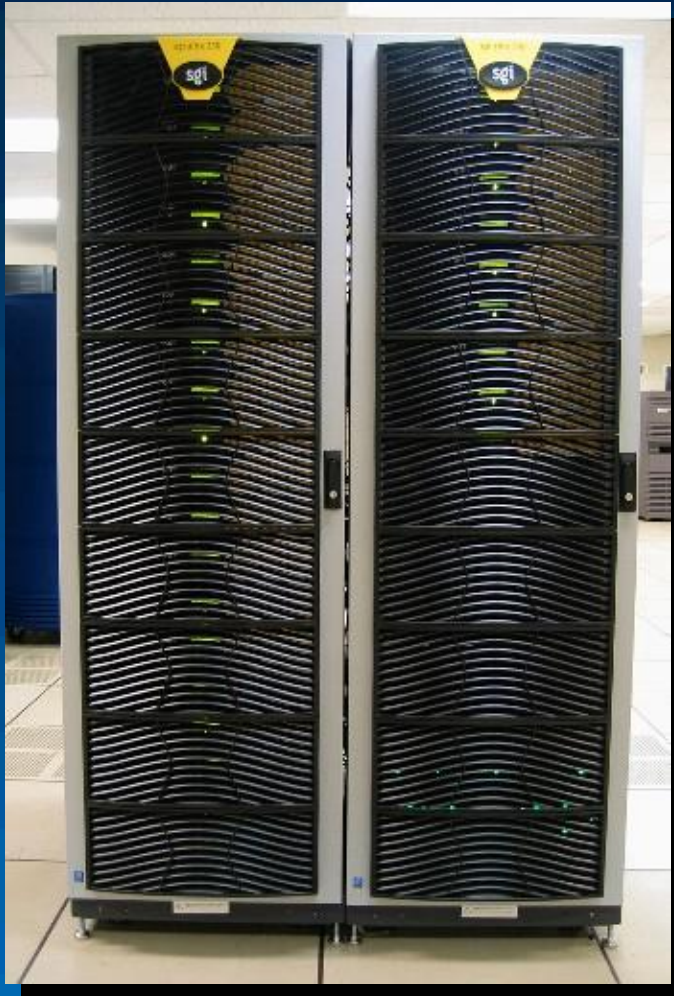


- 📖 **FPGA stands for Field-Programmable Gate Array**
- 📖 **This is a chip that can be reconfigured to have different circuits for every job that runs.**
- 📖 **A circuit can be designed to do a critical function in a program. When the program starts, perhaps 100 copies of this circuit are flashed onto the FPGA chip. When the job gets to that step of the calculation, it runs as though it were running on 100 CPUs.**
- 📖 **Programming FPGAs is more like designing circuits than writing computer code.**
- 📖 **Some C like programming tools are under development.**
- 📖 **ASC has FPGA chips and software for programming them available on the Cray XD1.**

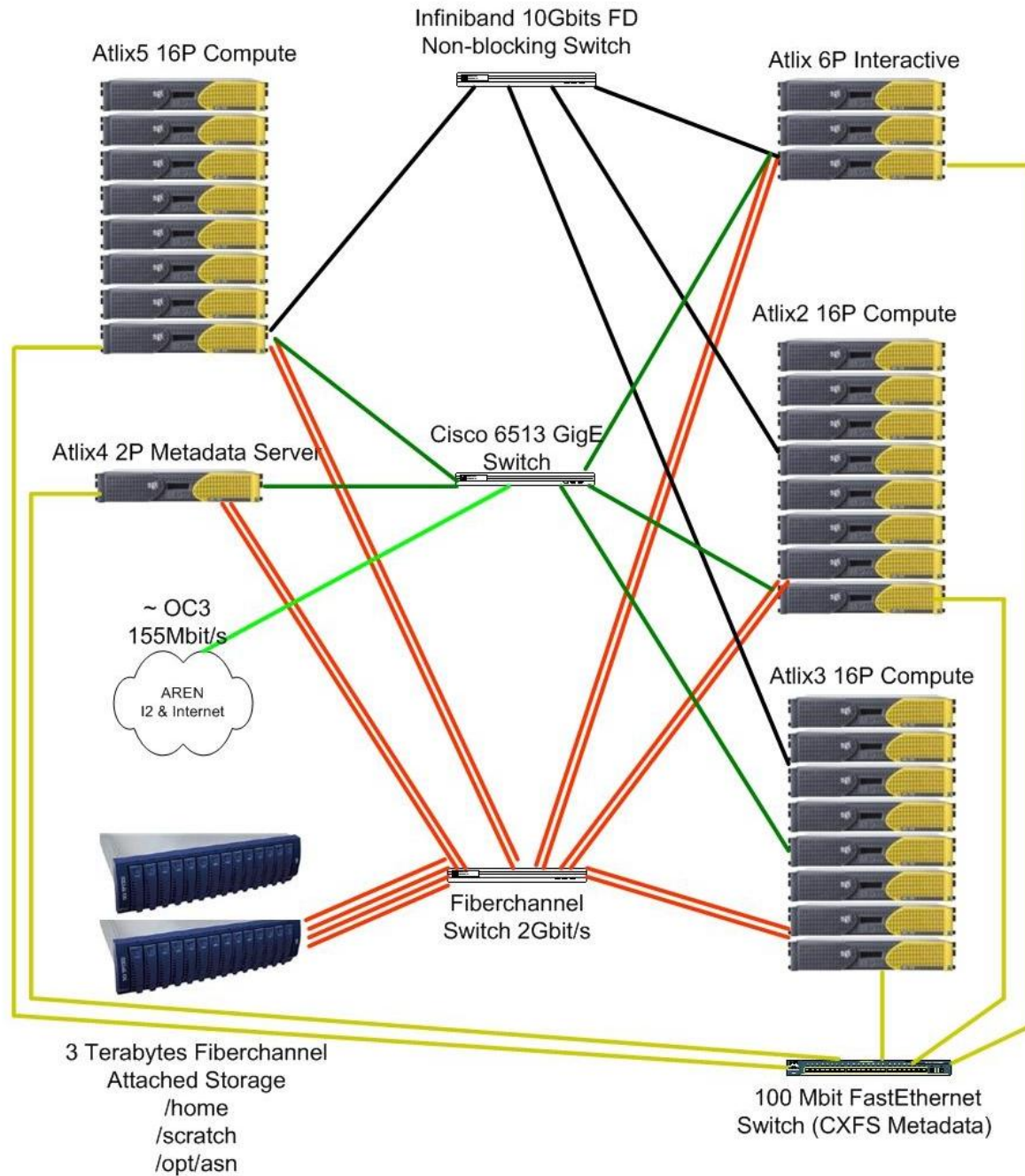




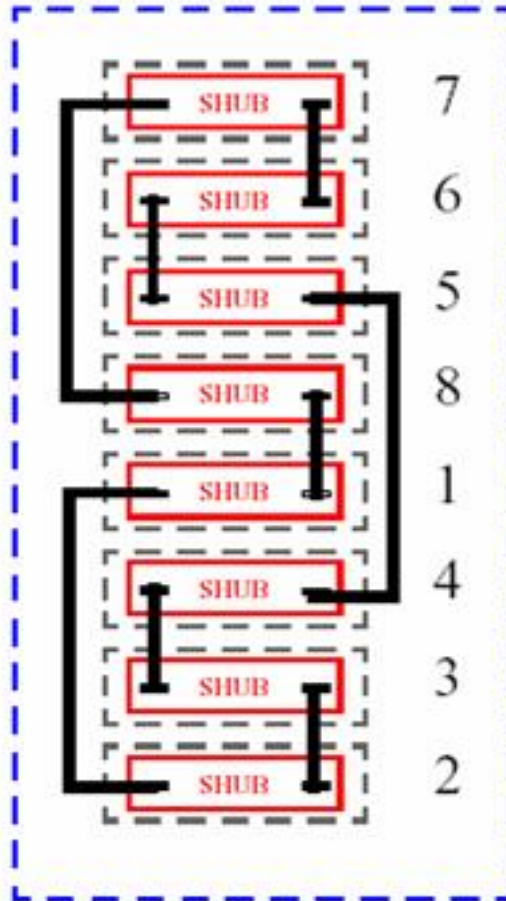
SGI Altix Supercomputer



- 100 Itanium 2 Processors
 - 525 GFLOPS Peak
- Shared Memory Architecture
 - NUMALink, Infiniband, fiber channel and gigabit ethernet data networks.
- Memory
 - 304 Gigabytes
- Disk Storage
 - 5.4 Terabytes



Memory access from one cpu to memory in another brick can have up to 5.5x the Latency vs. local memory access (worse case node 1 to 7)



SHUB Hops

Node	1	2	3	4	5	6	7	8
1	0	1	2	3	4	5	6	1
2	1	0	1	2	3	4	5	2
3	2	1	0	1	2	3	4	3
4	3	2	1	0	1	2	3	4
5	4	3	2	1	0	1	2	3
6	5	4	3	2	1	0	1	2
7	6	5	4	3	2	1	0	1
8	1	2	3	4	3	2	1	0

Average Hops = 2.25

Remote Latency (ns)

	1	2	3	4	5	6	7	8
1	134	301	388	477	566	654	742	301
2	301	134	301	388	477	565	655	388
3	388	301	134	301	388	477	565	477
4	477	388	301	134	301	388	477	565
5	565	477	388	301	134	301	388	477
6	654	565	477	388	301	134	301	388
7	742	654	565	477	389	301	134	301
8	301	388	477	565	477	388	301	134



Performance Comparison



	Cray SV1	SGI Altix	Cray XD1
# CPUs	16	100	144
CPU Type	Cray	Itanium 2	Opteron
SPECFP/cpu	56	1931	1553
Relative power	1	215	250
Memory (GB)	16 shared	304 shared	240 dist.
Disk (GB)	480	980	5328
Storage (TB)	2.2	5.4	7.1
GFLOP	19.2	525	634
Clock (GHz)	0.25 – 1 vector	1.4, 1.5, 1.6	2.2



Job Queue System Purpose



- 1. Ensure each job gets the requested resources (CPUs, memory, disk).**
- 2. Run jobs in an order that is in accordance with the facility management policies.**
- 3. Utilize the system as fully as possible, within the constraints of items 1 and 2. This implies choosing which job to run next, and choosing which node to run that job on.**

The batch queue standard is IEEE POSIX 1003.2d



Queue system genealogy



📖 **NQS – The very first batch queue system, first from Cray, later Monsano**

- NQE – Cray specific variant of NQS

📖 **PBS – developed for NASA, then commercialized by Veridian, then Altair**

- PBS Pro – commercial version of PBS
- OpenPBS – pre-commercialization version of PBS
- Torque – community developed PBS derivative
- MOAB – a commercial scheduler, often used with Torque

📖 **Maui Scheduler – for torque, PBS or SGE**

📖 **Sun Grid Engine (SGE) – open source developed primarily by Sun**

📖 **LSF – developed by platform computing, mostly used by US federal government sites**

📖 **Loadleveler – from IBM**

📖 **To the casual user, NQS, NQE, PBS, Torque, and SGE all feel similar.**



PBS Pro queue system



User commands

- **qsub** – run a job
- **qstat** – see status
- **tracejob** – see accounting

PBS runs jobs to ensure maximum utilization of the system, without over-subscription, and ensures jobs get the requested amount of memory/CPU's.

Login Node

- **Server** – keeps track of queues and jobs
- **Scheduler** – chooses when/where to run jobs

Compute node

- * **MOM daemon** – runs jobs and reports available CPUs/memory



ASC queue list



Queue	CPU	Mem	File #	CPUs	Run	Pri
batch						
large-serial	168:00:00	6gb		1	32	45
express	01:00:00	500mb	8gb	1	32	70
medium-serial	90:00:00	4gb	16gb	1	32	50
medium-parallel	90:00:00	14gb	16gb	4-8	11	30
small-parallel	40:00:00	4gb	8gb	2-4	32	40
large-parallel	168:00:00	45gb		4-16	11	20
small-serial	40:00:00	1gb	8gb	1	44	60

A single user may have 10 jobs running at once, not to exceed 16 CPUs.
Additional jobs will be held queued.

Interactive use is limited to 10 minutes of CPU time.



qsub



```
$PBS_BIN/qsub -q $queue -N $jobname -a $stime -r n -j eo -c s  
-M $userid -l cput=$time,mem=$memory,ncpus=$num_cpus  
-W umask=022 -W group_list=classq
```

- q** which queue
- N** job name
- a** time to start the job
- r** not rerunable
- j** merge stderr and stdout to log file
- c** disable checkpointing
- M** user to email about job failures
- l** resource settings
- W** additional attributes





qstat



```
altix:asndcy> qstat -a
```

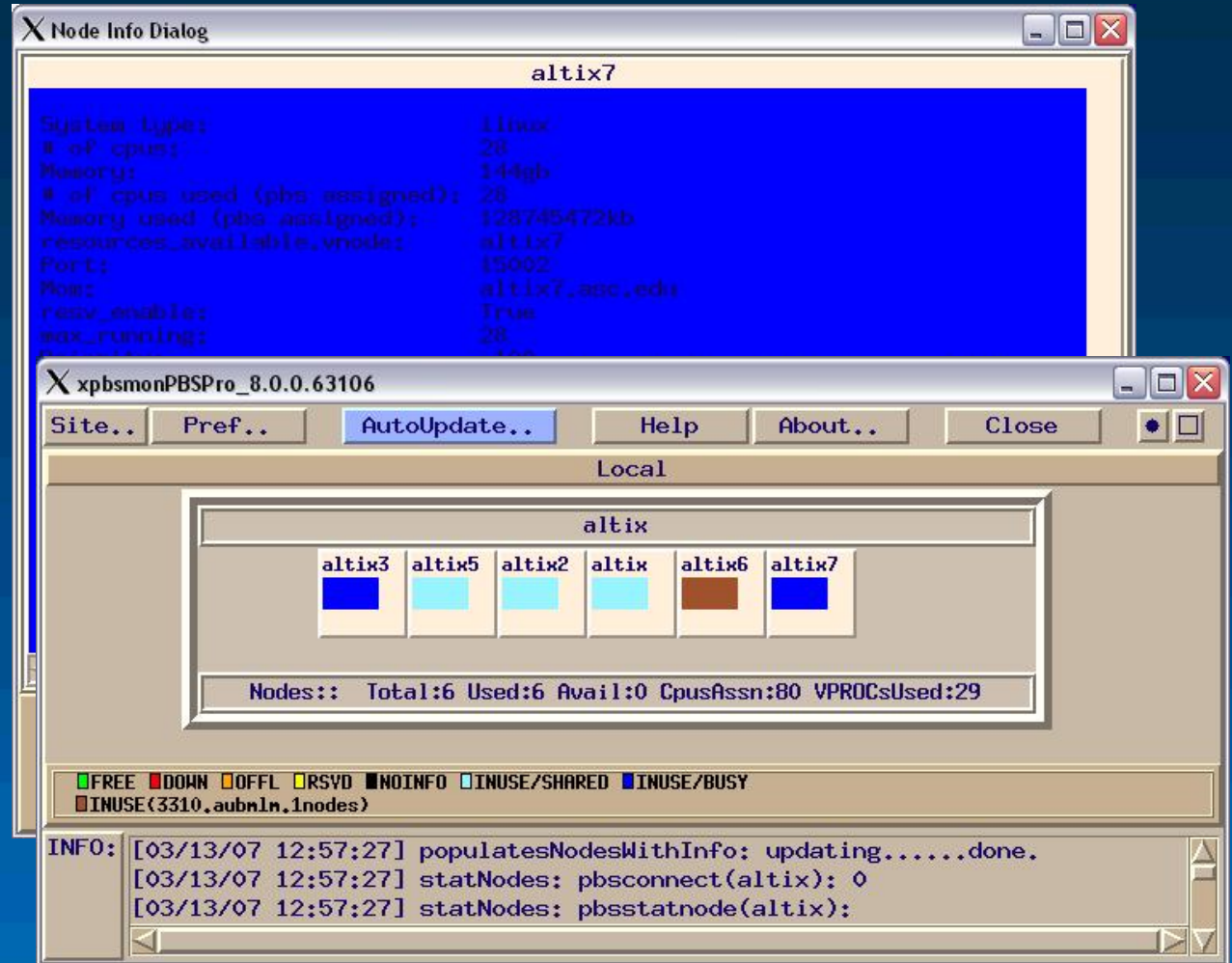
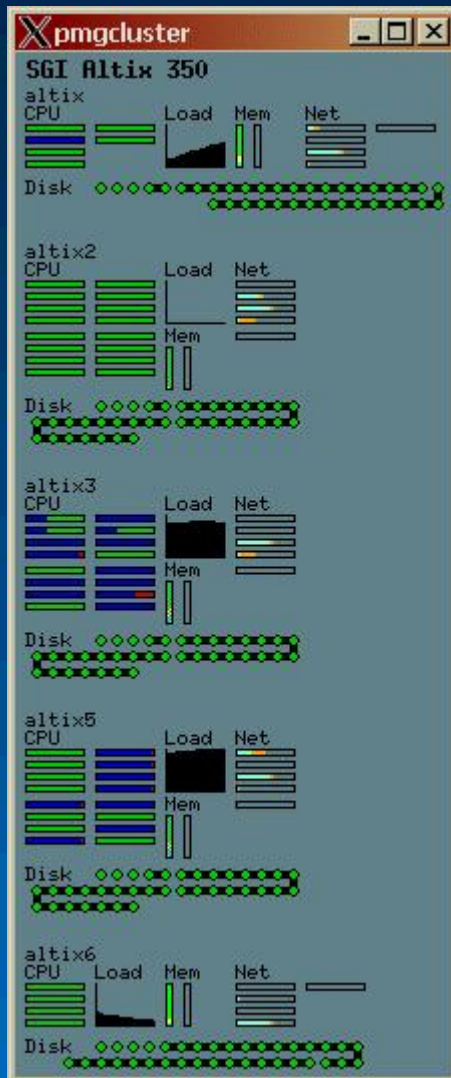
```
altix:
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	Elap S	Time
3179.altix	ualsxl	large-pa	w3o9m12a1p	30584	1	4	36gb	672:0	R	268:5
3188.altix	ualcka	large-pa	R1	25957	1	8	8gb	2000:0	R	569:4
3204.altix	aubdkh	large-se	g3p4o10	27444	1	1	10gb	168:0	R	64:05
3242.altix	ualsxl	large-pa	w3o9m12a1c	10634	1	4	36gb	672:0	R	111:5
3249.altix	ualcka	large-pa	R2	30294	1	8	8gb	2000:0	R	204:3
3252.altix	uahbam	medium-s	test-1	15797	1	1	1gb	90:00	R	25:32
3253.altix	uahbam	medium-s	test	15966	1	1	1gb	90:00	R	25:32
3255.altix	usaeas	medium-p	P5GtsinpG0	11667	1	4	4gb	360:0	R	56:43
3266.altix	uahbam	small-se	test-2	29950	1	1	1gb	40:00	R	22:24
3270.altix	usaeas	medium-p	P4Gredo3in	9324	1	4	4gb	360:0	R	47:18
3271.altix	uahbam	small-se	test-3	9510	1	1	1gb	40:00	R	21:15
3274.altix	ualrxc	large-pa	pyrpdpph2f	18821	1	8	4gb	1344:0	R	115:5
3275.altix	ualrxc	medium-p	pdpph2fcom	13123	1	4	2gb	360:0	R	54:29
3279.altix	uahgwh	large-pa	fdvjob3	19106	1	4	8gb	300:0	R	75:15
3284.altix	uahgwh	medium-p	fdvjob1	13390	1	4	8gb	300:0	R	74:48
3310.altix	aubmlm	large-pa	xxxqq199co	4947	1	4	45gb	672:0	R	19:19
3311.altix	asndcy	sysadm	cupytest1G	24357	1	1	20gb	168:0	R	04:15
3312.altix	aubjxo	large-pa	wbrunwithS	--	1	10	10gb	1680:0	Q	--





pmgcluster and xpbsmon





PBS Pro Features



Priority based scheduling*

- 📖 Back Filling
- 📖 Starving job support
- 📖 Fairshare
- 📖 Prime time scheduler options
- 📖 Preemptive scheduling
- 📖 Dedicated time
- 📖 Advanced reservation
- 📖 Job arrays
- 📖 Restartable jobs
- 📖 Waiting for another job
- 📖 Hard and soft run limits
- 📖 Node limits and priorities.
- 📖 Node grouping (i.e. same chassis)

Non scheduling features

- 📖 Access lists
- 📖 Grid computing (Globus or peer scheduling)
- 📖 Virtual nodes*
- 📖 Routing queues*
- 📖 Accounting*
- 📖 Custom resources (i.e. FPGA)
- 📖 Prologue & epilogue

* Additional slides follow on this item





More queue terminology



- 📖 **Chunk** – a set of resources (CPUs, memory, or disk) within a physical node that is assigned to a given job. A job might have one chunk for each MPI process.
- 📖 **Load balance** – a policy of utilizing all nodes equally. The ASC computers use the opposite policy of utilizing the most heavily subscribed node, in order to leave large chunks available for subsequently submitted jobs.
- 📖 **Node attribute** – a setting defining how much memory, etc. the queue system is allowed to utilize.
- 📖 **Virtual processor** – define how many processes to run on a set of CPUs, thus allowing overloading CPUs. At ASC, one job process can be run for each physical CPU core.
- 📖 **Account** – used for charging for resources... not utilized at ASC



tracejob



tracejob -n 14 3226

Job: 3226.altix

```
03/11/2007 10:41:14 L Considering job to run
03/11/2007 10:41:14 S enqueueing into large-serial, state 1 hop 1
03/11/2007 10:41:14 S Job Queued at request of ualsxl@altix.asc.edu, owner =
                        ualsxl@altix.asc.edu, job name = cupy1a1c2vmp2a, queue = large-serial
03/11/2007 10:41:14 S Job Run at request of Scheduler@altix.asc.edu on hosts
                        (altix6:mem=10485760kb:ncpus=1)
03/11/2007 11:47:16 S Exit_status=271 resources_used.cput=98
03/11/2007 11:47:16 A user=ualsxl group=ualchem
                        accounting_id="0x42811a0900000113"
                        jobname=cupy1a1c2vmp2a queue=large-serial
                        ctime=1173627673 qtime=1173627674 etime=1173627674
                        start=1173627679 exec_host=altix6/3
                        exec_vnode=(altix6:mem=10485760kb:ncpus=1)
                        Resource_List.cput=168:00:00 Resource_List.mem=10gb
                        Resource_List.ncpus=1 Resource_List.nodect=1
                        Resource_List.pcpus=168:00:00 Resource_List.place=pack
                        Resource_List.select=1:mem=10gb:ncpus=1
                        Resource_List.walltime=252:00:00 session=16237
                        end=1173631636 Exit_status=271
                        resources_used.cput=98 resources_used.cput=01:02:45
                        resources_used.mem=12338784kb resources_used.ncpus=1
                        resources_used.vmem=12606288kb resources_used.walltime=01:06:02
```





qmgr Global



```
set server max_user_run = 10
set server max_user_res.mem = 144gb
set server max_user_res.ncpus = 16
set server managers = asndcy@altix.asc.edu
set server default_queue = batch
set server log_events = 511
set server mail_from = dyoung@asc.edu
set server query_other_jobs = True
set server resources_available.ansyslic = 5
set server resources_available.cfdacejob = 8
set server resources_default.mem = 1gb
set server resources_default.ncpus = 1
set server default_chunk.ncpus = 1
set server scheduler_iteration = 61
set server resv_enable = False
set server node_fail_requeue = 0
set server max_array_size = 10000
```





QMGR queue config



```
create queue class
set queue class queue_type = Execution
set queue class Priority = 80
set queue class max_running = 16
set queue class resources_max.cput = 16:00:00
set queue class resources_max.file = 8gb
set queue class resources_max.mem = 30gb
set queue class resources_max.ncpus = 16
set queue class resources_max.pcpu = 01:00:00
set queue class resources_max.walltime = 01:30:00
set queue class resources_min.ncpus = 1
set queue class resources_default.cput = 01:00:00
set queue class resources_default.file = 8gb
set queue class resources_default.mem = 1gb
set queue class resources_default.ncpus = 1
set queue class acl_group_enable = True
set queue class acl_groups = analyst
set queue class acl_groups += root
set queue class acl_groups += classq
set queue class enabled = True
set queue class started = True
```





Scheduler configuration



The scheduler sched.config file contains 41 parameters.

```
round_robin: False  all
by_queue: False    all
strict_ordering: false ALL
help_starving_jobs: true  ALL
max_starve: 24:00:00
backfill: true     ALL
backfill_prime: false  ALL
prime_exempt_anytime_queues: false
primetime_prefix: p_
nonprimetime_prefix: np_
job_sort_key: "cput LOW"  ALL
node_sort_key: "sort_priority HIGH"  ALL
sort_queues: false  ALL
resources:
  "ncpus,mem,arch,host,vnode,ansyslic,abaquslic,fidapjob,fidapcpu,fluentjob,flue
  ntcpu,nastranlic,jaguarlic,cfdacejob,cfdfastranjob,cfdfastrancpu"
load_balancing: false ALL
smp_cluster_dist: pack
fair_share: false  ALL
half_life: 24:00:00
sync_time: 1:00:00
preemptive_sched: false ALL
dedicated_prefix: ded
log_filter: 1280
```



vnode



A vnode is a virtual node. It is a collection of resources that can be managed as a unit by the queue system. A vnode can be;

- 📖 An entire computer system or cluster of physical nodes.**
 - 📖 A single physical node.**
 - 📖 A subset of the CPUs and memory within a physical node.**
 - 📖 Some subset of the physical nodes within a cluster.**
 - 📖 Used to overload multiple processes onto each CPU.**
- 📖 Note that a physical node is defined as a collection of CPUs running under a single instance of the operating system.**



Internal scheduler algorithm



- 📖 The API for creating new schedulers is well documented, but the documentation does not give a detailed description of the algorithm used by the PBS Pro scheduler.
- 📖 Internally, PBS has a type of priority system that determines which job is “most deserving” of running next.
- 📖 This seems to take into account the queue priority, how long the job has been waiting, etc.
- 📖 In essence, the starving job support option trumps other priorities. Thus starving jobs should be run FIFO before other jobs are run, regardless of priority.
- 📖 When a soft run limit is reached, subsequent jobs from that user are forced to the bottom of the priority list.



Internal scheduler algorithm



Job properties

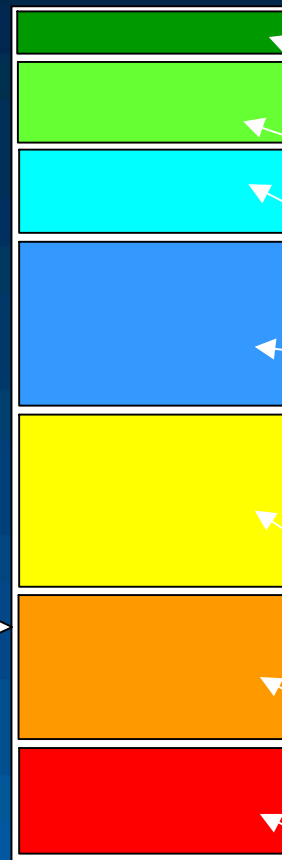
- Memory
- CPUs
- Owner
- Queue priority
- Time pending

System resources exhausted

Node properties

- Memory
- CPUs
- Priority
- Dedicated queue

Run first



Reserved or preemptive

Starving jobs

Fairshare priority

Jobs in high priority queues, or low priority jobs that have been waiting a longer time.

Jobs in medium priority queues

Large, low priority jobs.

Jobs exceeding soft quotas

Jobs exceeding hard quotas.

Run last



Don't Run



PBS Pro Problems



- 📖 **Multiple queue schedulers can't feed from one flexlm server cleanly.**
- 📖 **Users usually request more CPU time than they need.**
- 📖 **Bugs fixed in ver 8.0...**
 - Ignoring resource limits
 - Starving job support was broken in version 7.x
- 📖 **Scheduler doesn't always work according to the documentation. i.e. if two jobs have same priority, etc. the one belonging to the user with fewer jobs running will be run first.**
- 📖 **No interaction with dplace, runon, or taskset.**
 - Software written at ASC takes care of this
- 📖 **No allocation of FPGA blades**
 - Application of custom resources
- 📖 **Routing queue algorithm is too simplistic.**

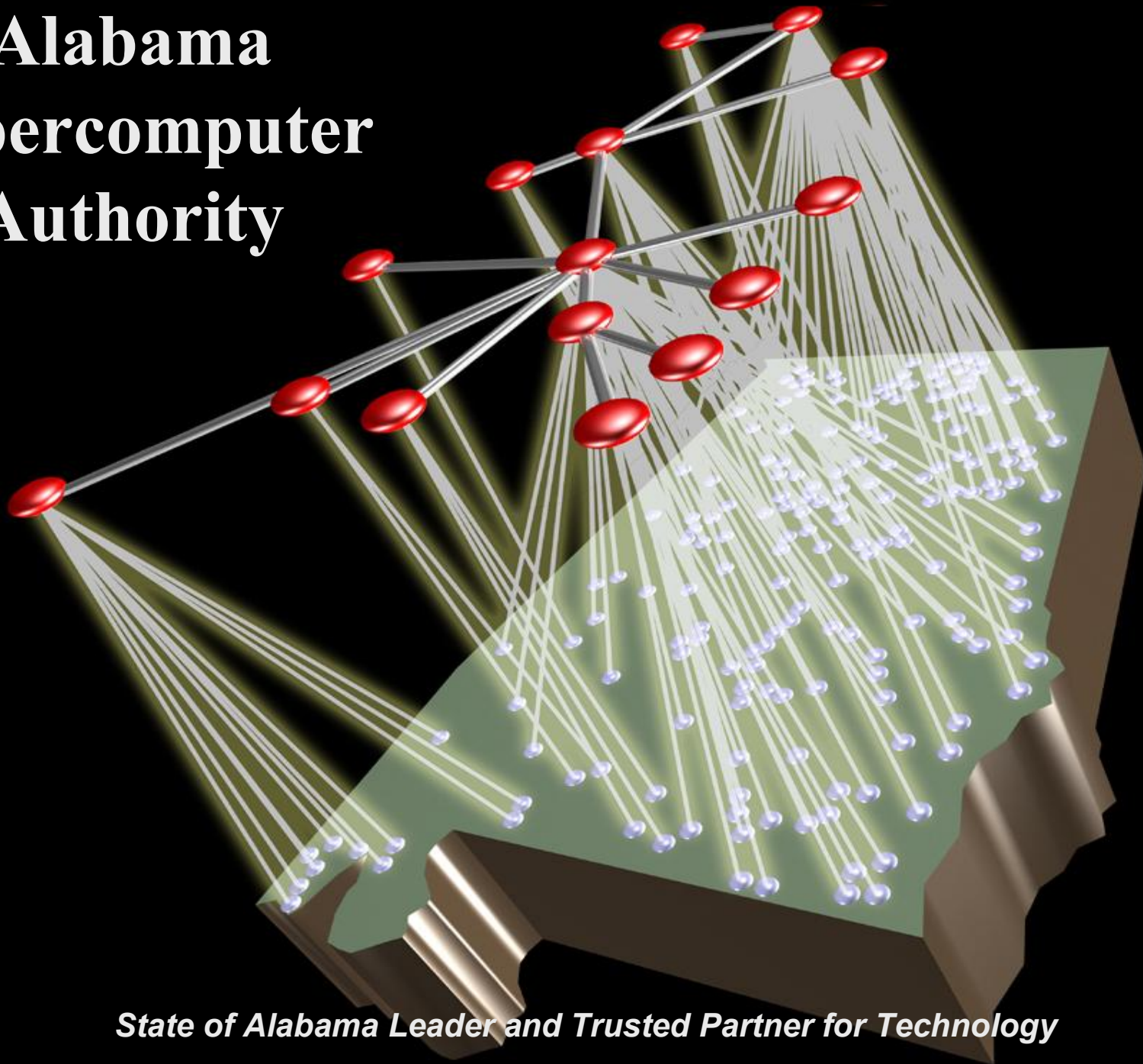


Summary



- 📖 **The Alabama Supercomputer Authority provides two high performance computing systems. These are free of charge for use by state funded educational institutions in Alabama.**
- 📖 **The SGI Altix 350/450 is a shared memory system with NUMA (non-uniform memory access) and Intel Itanium 2 CPUs.**
- 📖 **The Cray XD1 is a distributed memory system using Cray's RapidArray interconnect and AMD Opteron CPUs.**

Alabama Supercomputer Authority



State of Alabama Leader and Trusted Partner for Technology



SGI Altix Supercomputer



- 1 - 6 CPU Login Node (altix.asc.edu)
2 CPUs Interactive + 4 CPUs Running Jobs
- 4 - 16 CPU Compute Nodes
 - 3 1.4 Ghz Nodes (altix2,3,5)
 - 1 1.5 Ghz Node (altix6)
- 1 – 28 CPU Altix 450 Compute Node (altix7)
 - 1.6 Ghz Dual Core Itanium 2
 - 144 GB of RAM
- 1 – 2 CPU CXFS Metadata Service Node (altix4)