Tera Scale Systems

Applications

NASA Ames Research Center
Advanced Supercomputing Division
User Services Director
Chuck Niggey

Moffett Field, CA 94035-1000
Planetary Modeling
Instrument Data Analysis and Support
Chemistry/Specien
Stellar Dynamics
Space Science
Chemistry
Nano Technology
Oceanography
Atmospheric Physics
Earth Science
Protein Folding and Beyond
Astrobiology
Aerospace Control and Simulation
Shuttle Upgrades
X-Vehicle Development
Aeronautics

NASA's Computational Challenges
Cray - J90 Cluster (vendor backed out of commitment)
- SGI - Power Challenge Cluster (reliability)
  - IBM - SP2 (performance)
  - Intel - Paragon (performance)
  - Intel - IPSC-860 (performance)
- Connection Machine - CM5 (performance)
- Connection Machine - CM2 (performance)

Many Attempts (these all failed...)

It's all about the interconnect

Have to move into the realm of hundreds or thousands of processors

As Reliable as the "Gold Standard"

Productive Environment for the Users (easy to use)

Less Expensive

Significantly Faster than previous generation

What is a Production Supercomputer?

Highly Parallel Supercomputer

Goal: Production Quality
Shared memory can go a long way towards SIMPLIFICATION.

- Space Science
- Molecular Dynamics
- DAO
- Computational Chemistry

For example, applications.

Many (most) productive scientists are simply unable to access supercomputing because it is either difficult or even not possible to effectively scale their productive computations. Many (most) supercomputers manufactured in 1993.

This can translate into several man-years of effort, and when you have done this, you'll likely not run faster than a C90 supercomputer manufactured in 1993. However, it's typical that applications may require substantial modifications to achieve even moderate levels of parallelism. There are many problems that require supercomputing for their research.

NASA

Barriers to Scientists Obtaining a Supercomputing Capability for Their Research
High-End Computing

England
CUG Summit 2002 -- Manchester

are needed to see this picture.

Cinepak decompreser and a QuickTime and a
Expects to sustain 20% of peak - 200 Gflops

- Processor speed upgrade (600 MHz) April 2002
- Topology in October
- Moved to higher bandwidth/lowert latency
- 1024p in initial topology configuration in Sep
- 2 operational 512p 400MHz 03000 systems

SGI O3K 1024 CPU System
What happened this past year:

- Hardware
  - Memory Failures -- 10
  - CPU Failures -- 15
  - Router Failures -- 6
  - Unknown/Other -- 28

- Software
  - Kernel Changes -- 15
  - MPI/User Code Hangs -- 20
(3) setting \texttt{OMP\_NUM\_THREADS}, \texttt{OMP\_NUM\_THRDSDS}, \texttt{MP\_SET\_NUM\_THREADS}, \texttt{OMP\_DYNAMIC}, \texttt{MPC\_CANG}, \texttt{MPI\_UNBUFFERED\_STUDIO}

(2) new PBS environment variable \texttt{PBS\_CPUS} which says

\texttt{just which CPUs a job is using (in a compact representation)}

the collector thread to finish one scan

\texttt{collector Interval = number of seconds allowed for collector\_interval (default: 5 seconds)}

\texttt{(schedule\_nodes\_per\_clump)}

\texttt{(should be the same as the number of nodes in a \texttt{clump})}

\texttt{num\_clumps\_size = number}\n
\texttt{PBS\_mom:}

\texttt{PBS\_mom Changes}
PBS Scheduler Enhancements

(1) New charging algorithm

(2) Software-defined "clumps" rather than previous "hardware-defined "clumps" of 64 nodes

(3) New configuration options: allows easy implementation of "emergency" deductions

SCHEDULETZ - basically, setting of TZ needed for schedule output

LOCAL_DEDTIME_FILE - allows easy implementation of "clumps" of 64 nodes

NODES_PER_CLUMP - size in nodes of software-defined "clumps"

STUCK TOO LONG - number of seconds to wait before dumping out process information about a stuck cpuset

ALLOW CPUSET - list of usernames allowed to use cpuset.

clumps
(6) on first iteration after server comes up, try to rerun any previously-running jobs in their original queue

(5) combining jobs to nodes specified in queue's assigned nodemask (allowing the system to be partitioned, if that's thought necessary)

(4) handling of &quot;group -l queuetag=...</nodeomask&quot; which lets empowered users specify the queue the cluster a job should be run in

PBS Scheduler Enhancements
NAS staff working together to resolve issues.

Lots of progress has been made due to SGI and space.

- Slow MPI startup due to cross mapping of addresses.
- Codes (Parallel Ocean Modelling) still experience interference when run against other.
- BUT, since Inx 6.5 I6 has installed system up 10 days.
- Not quite there yet.

Where are we today?
Scalable issues exist in WM and I/O subsystems

Systems Software

Work on general code scaling

Layouts and effects on performance

More instrumentation to precisely measure memory bandwidth on performance

More instrumentation to measure effects of

Applications

Where to go from here
Performance - The Focus is on Parallelism

Dramatically better performance with increasing processor count

Minimum changes OVERFLOW (MPI/MLP=20,000/800,000 lines), FCORE (8000/400 lines)

Much easier to build/port code than MPI (Man months vs. Man years)

No messaging - All communication via shared memory

Simple extension to Cray parallel/vector programming model - 3 routines, 150 lines of source

NASA’s Shared Memory Multi-Level Parallelism (MLP)

Very difficult to scale to 100’s of CPUs without major rewrite

Reddy acceptable only for small processor counts

Shared Memory Parallelism (OpenMP)

Often requires simplification of physics for scaling

User provides all parallel decomposition/code modification

Explicit “messages” - Large latency - Very slow

Arccos and complex user interface - 100 routines, 50,000+ lines of source

Message Passing Interface (MPI)

If you don’t scale to hundreds of CPUs, you won’t get to the 100+ Gflops

Parallelism is the key to performance on any system manufactured today.
But method has been adapted to execute across small clusters as well.

Targeted for the new large CPU count NUMA SMP systems.

No messaging - communication through "global" common blocks.

Fine grained parallelism provided by the compiler at loop level (OMP).

Coarse grained parallelism provided by Unix forked processes.

Two levels of parallelism (the so-called "hybrid" approach).

An open system design (runs on any SMP) and has the following attributes:

- Increased is parallel efficiency during execution.
- Levels of parallelism within an application executing on a NUMA based system.
- Shared Memory Multi-Level Parallelism (MLP) is the utilization of multiple.

What is MLP?
In short, the recipe for converting a multi-zonal CFD code to MLP is:

- Synchronize computation as needed with barriers
- Use shared memory arenas to hold all global data (BCs etc)
- Use the CPUs in a group for fine grained parallelism for each zone
- Assign groups of CPUs to each MLP process
- Solve the groups of zones in parallel
- Assign groups of 3D zones to each MLP process
- Spawn MLP parallel processes

These smaller regions can be solved mostly in parallel, with the occasional exchange of boundary information at the end of a time step. These smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved mostly in parallel, with these smaller regions can be solved most...
neutron stars

Newtonian inspiral of two
Stars
Head on collision of two neutron
Computational Nanosciences
Special Thanks