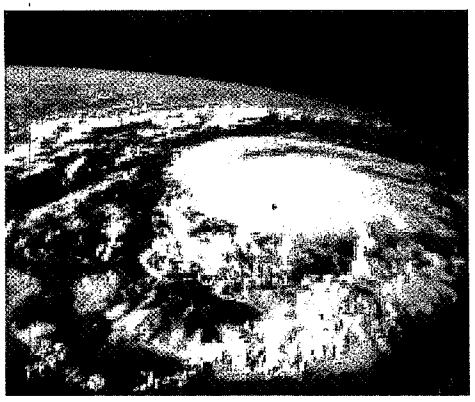
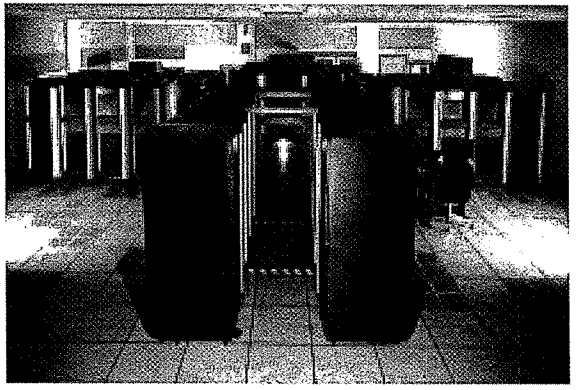
Information Sciences & Technology
Ames Research Center

NASA

Scaling NASA Applications to 1024 CPUs on Origin 3K



NRC Review
June 13, 2002



Jim Taft
NASA AMES Research Center
jtaft@nas.nasa.gov

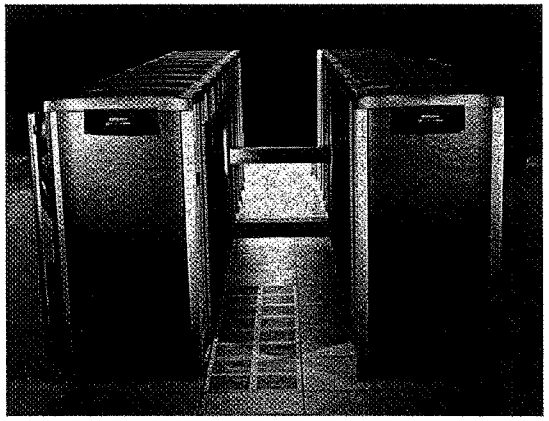
Information Sciences & Technology

Ames Research Center

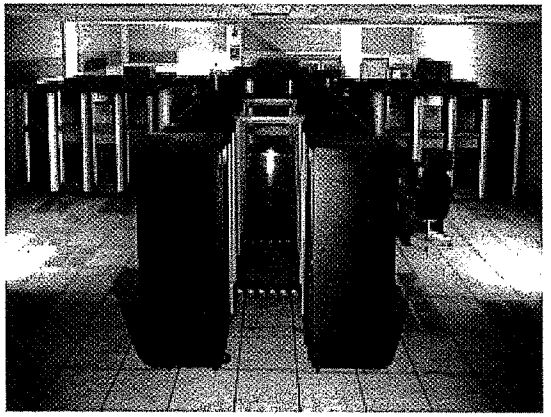
The Origin SSI Revolution at NASA Ames (1998-2001)

NASA

Steger: O2K 256CPUs 250MHz



Lomax: O2K 512CPUs 400MHz



Chapman: O3K 1024CPUs 600MHz

MLP a History of Success

The long and highly successful joint SGI-NASA research effort in ever larger SSI systems was to a large degree the result of the successful development of the MLP scalable parallel programming paradigm developed at ARC:

- *MLP scaling in real production codes justified ever larger systems at NAS*
- *MLP scaling on 256p Origin 2000 gave SGI impetus to productize 256p*
- *MLP scaling on 512 gave SGI courage to build 1024p O3K*
- *History of MLP success resulted in IBM Star Cluster based MLP effort*

Information Sciences & Technology

Ames Research Center

MLPlib - Simple Scalable Parallelism at NASA Ames

Performance - The Focus is on Parallelism

Parallelism is the key to performance on any system manufactured today. If you don't scale to hundreds of CPUs, you won't get to the 100+ GFLOPS you need today to stay competitive in high end computing. Parallelism was being aggressively pursued on two fronts. Now there are three.

- *Message Passing Interface (MPI)*
  - *Arcane and complex user interface - 100 routines, 50,000+ lines of source*
  - *Explicit "messages" – large latencies – very slow*
  - *User provides all parallel decomposition/code modification*
  - *Often requires simplification of physics for scaling*

- *Shared Memory Parallelism (OpenMP)*
  - *Really acceptable only for small processor counts*
  - *Very difficult to scale to 100's of CPU's without major rewrite*

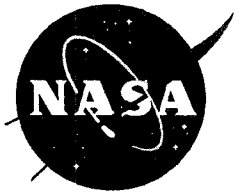
- *NASA's Shared Memory Multi-Level Parallelism (MLP)*
  - *Simple extension to Cray parallel/vector programming model - 3 routines, 150 lines of source*
  - *No messaging - All communication via shared memory*
  - *Much easier to build/port code than MPI (Man months vs. Man years)*
  - *Minimum changes OVERFLOW (MPI/MLP=20,000/800 lines), FVCORE (8000/400 lines)*
  - *Dramatically better performance with increasing processor count*

What is MLP?

Shared Memory Multi-level Parallelism (MLP) is the utilization of multiple levels of parallelism within an application executing on a NUMA based system architecture in order to increase its parallel efficiency during execution. It is an open system design (runs on any SMP) and has the following attributes:

- *Two levels of parallelism (the so-called "hybrid" approach)*
- *Coarse grained parallelism provided by Unix forked processes*
- *Fine grained parallelism provided by the compiler at loop level (OpenMP)*
- *No messaging - communication through "global" common blocks*
- *Targeted for the new large CPU count NUMA SMP systems*
- *But method has been adapted to execute across clusters as well*

How do you use it - MLPlib

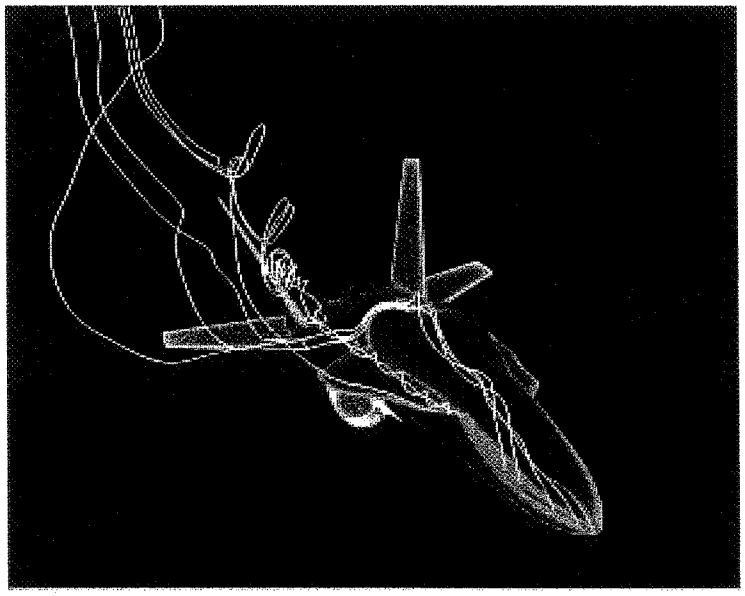


MLPlib consists of 3 routines - how hard can it be?

- Subroutine MLP_GETMEM(numvar,ipoint,isize)
  - Called once only from main program
  - Allocates the global common blocks
- Subroutine MLP_FORKIT(numpro,nowpro,numcps,idopin)
  - Called once only from main program
  - Forks the processes
- Subroutine MLP_BARRIER(nowpro,numpro)
  - Called as often as needed
  - Barrier synchs the processes

Information Sciences & Technology

Ames Research Center



CFD at NASA

OVERFLOW-MLP

MLP - A New Concept for Multi-zonal CFD

NASA's Multi-zonal CFD codes like OVERFLOW, CFL3D, LAURA, INS3D, and TLNS3D to name a few, are ideal candidates for MLP parallelism. These codes decompose a large region of interest into many linked smaller 3D regions. These smaller regions can be solved mostly in parallel, with the occasional exchange of boundary information at the end of a time step.

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

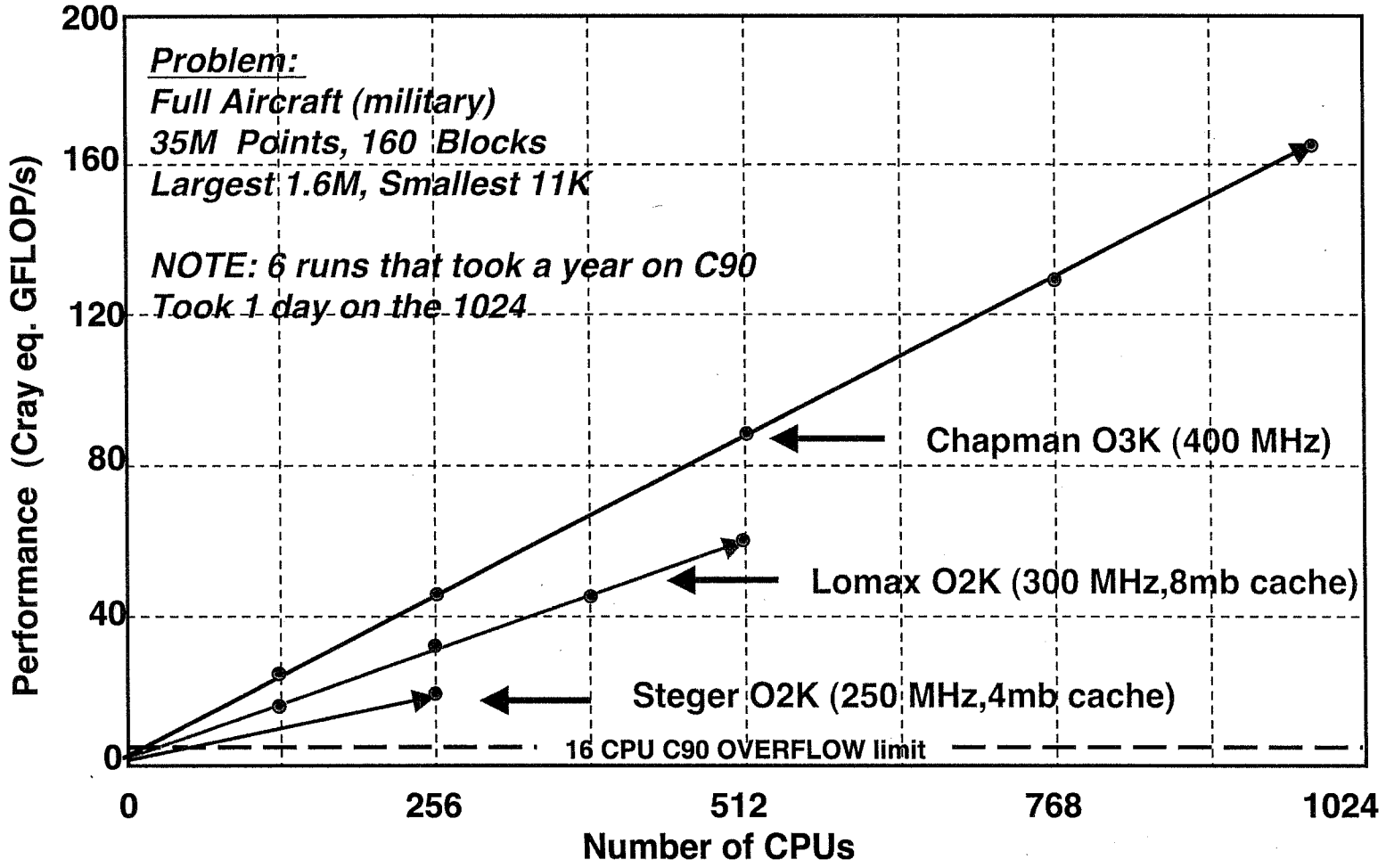
- *Spawn MLP parallel processes*
- *Assign groups of 3D zones to each MLP process*
- *Solve the groups of zones in parallel*
- *Assign groups of CPUs to each MLP process*
- *Use the CPUs in a group for fine grained parallelism for each zone*
- *Use shared memory arenas to hold all global data (BCs etc)*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

Information Sciences & Technology

Ames Research Center

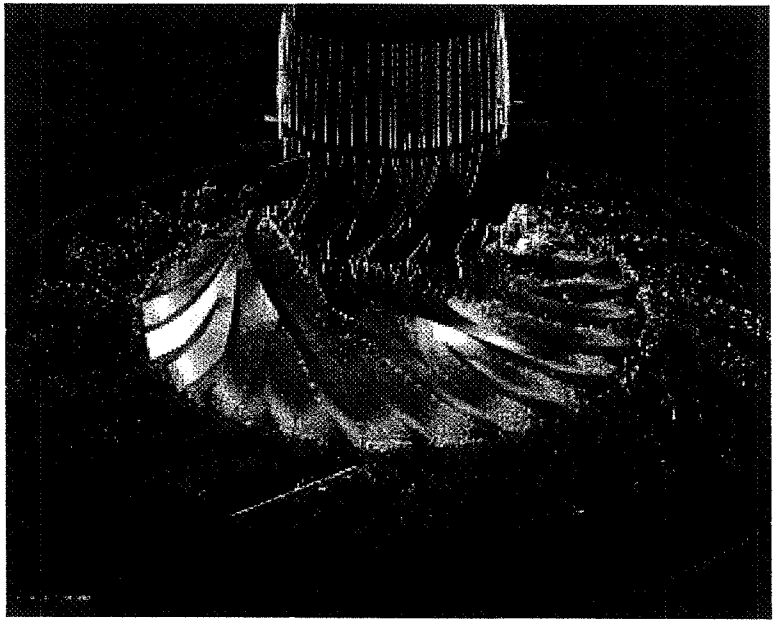
OVERFLOW-MLP Performance vs CPU Count

Systems: 1024 CPU O3K , 256&512 CPU O2KS



Information Sciences & Technology

Ames Research Center



Turbomachinery

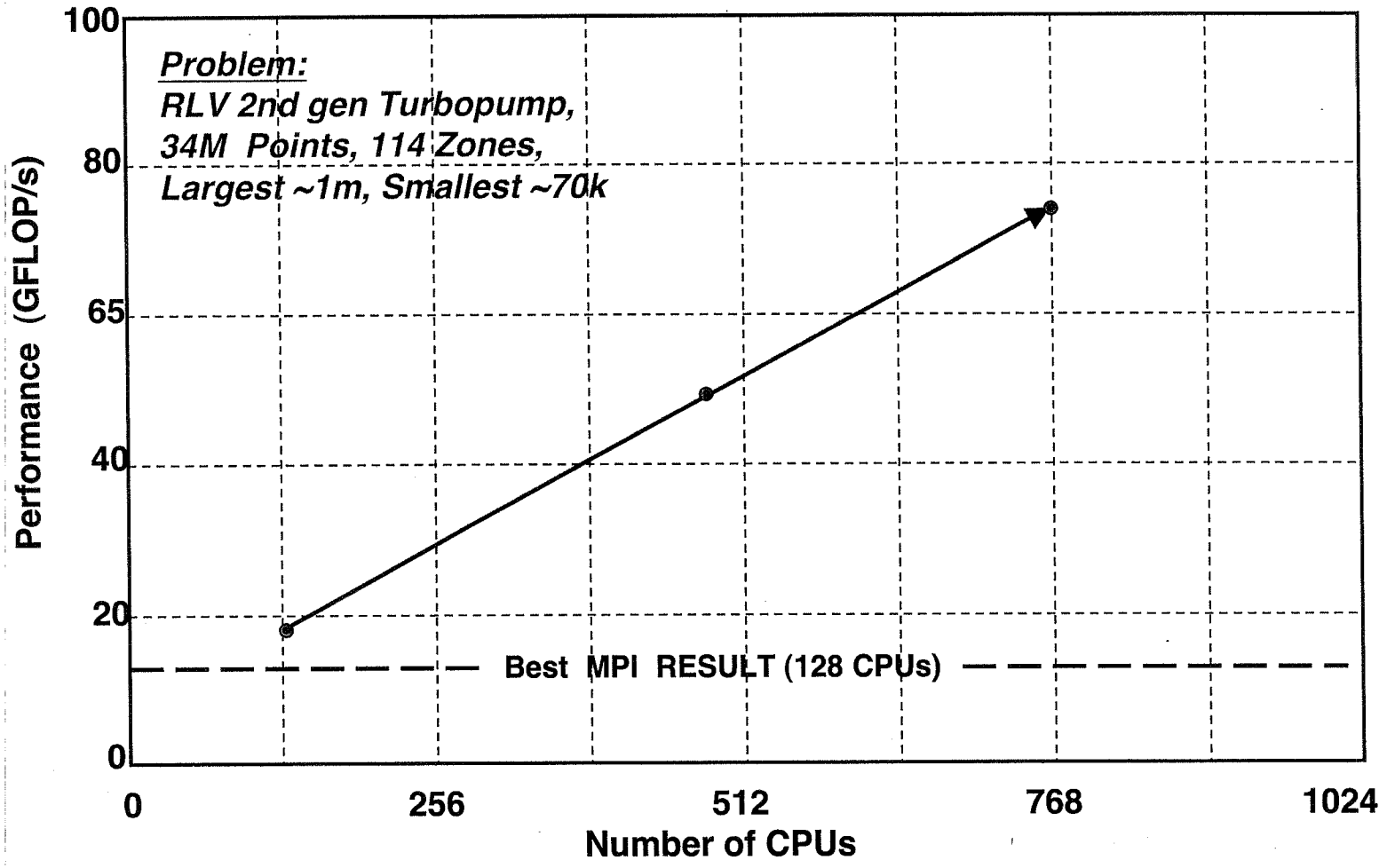
CFD at NASA

INS3D-MLP

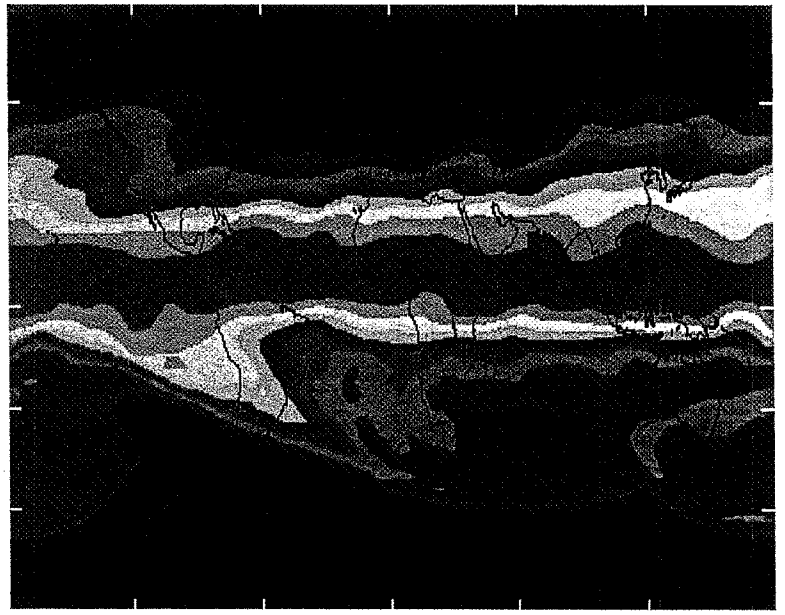
Information Sciences & Technology

Ames Research Center

INS3D-MLP Performance vs CPU Count
System: 1024 CPU O3K (400 MHz)



Ames Research Center



Climate Modeling

at NASA Ames

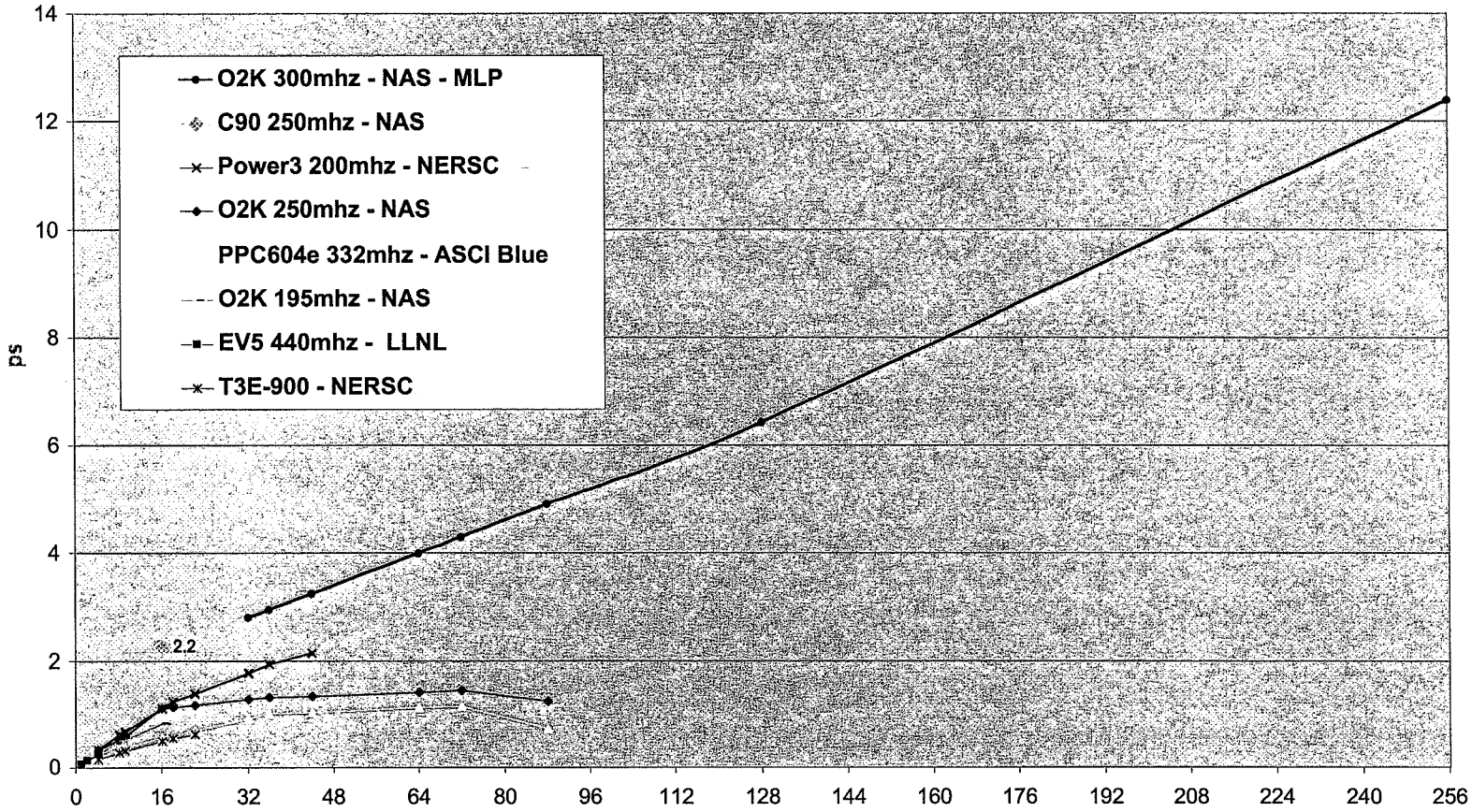
GEOS4-MLP

NASA Climate Modeling
Initial FVCORE-MLP Scaling on Popular Systems 06/00
(NAS MLP work completed in 21days)

Information Sciences & Technology
Ames Research Center

NASA

FVCORE 2x2.5@55 Levels (B55)



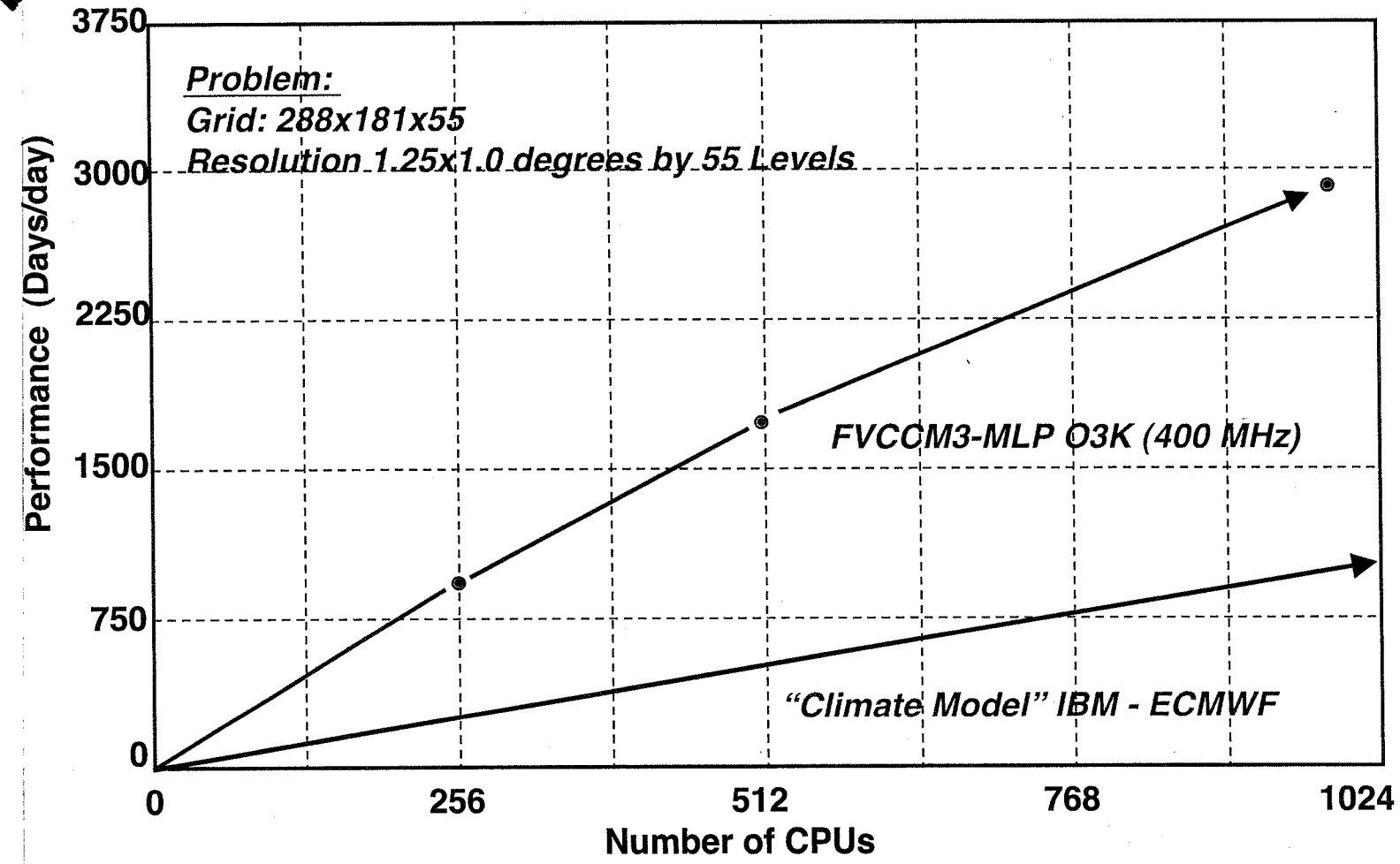
Non-O2k results derived from data provided by Mirin etal

Information Sciences & Technology

Ames Research Center

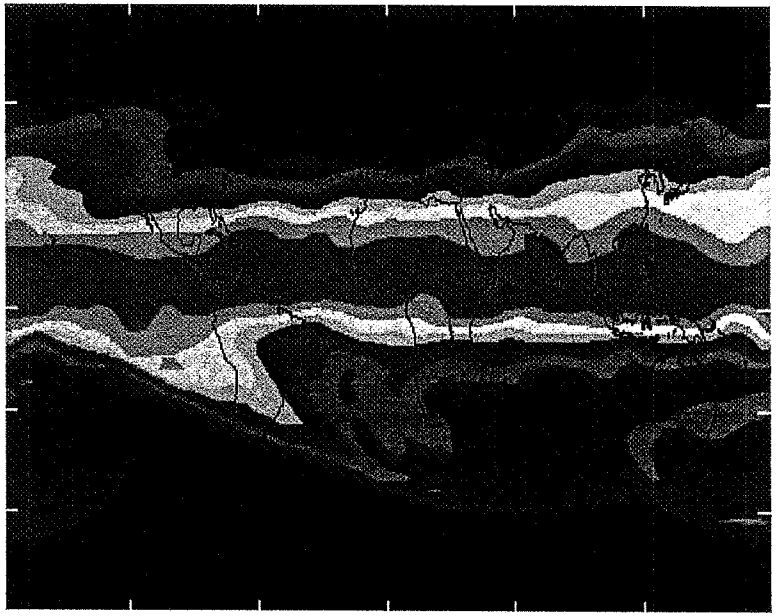
FVCCM3-MLP Performance vs CPU Count (2/02)

System: 1024 CPU O3K 400 MHz (chapman)



Information Technology

Ames Research Center



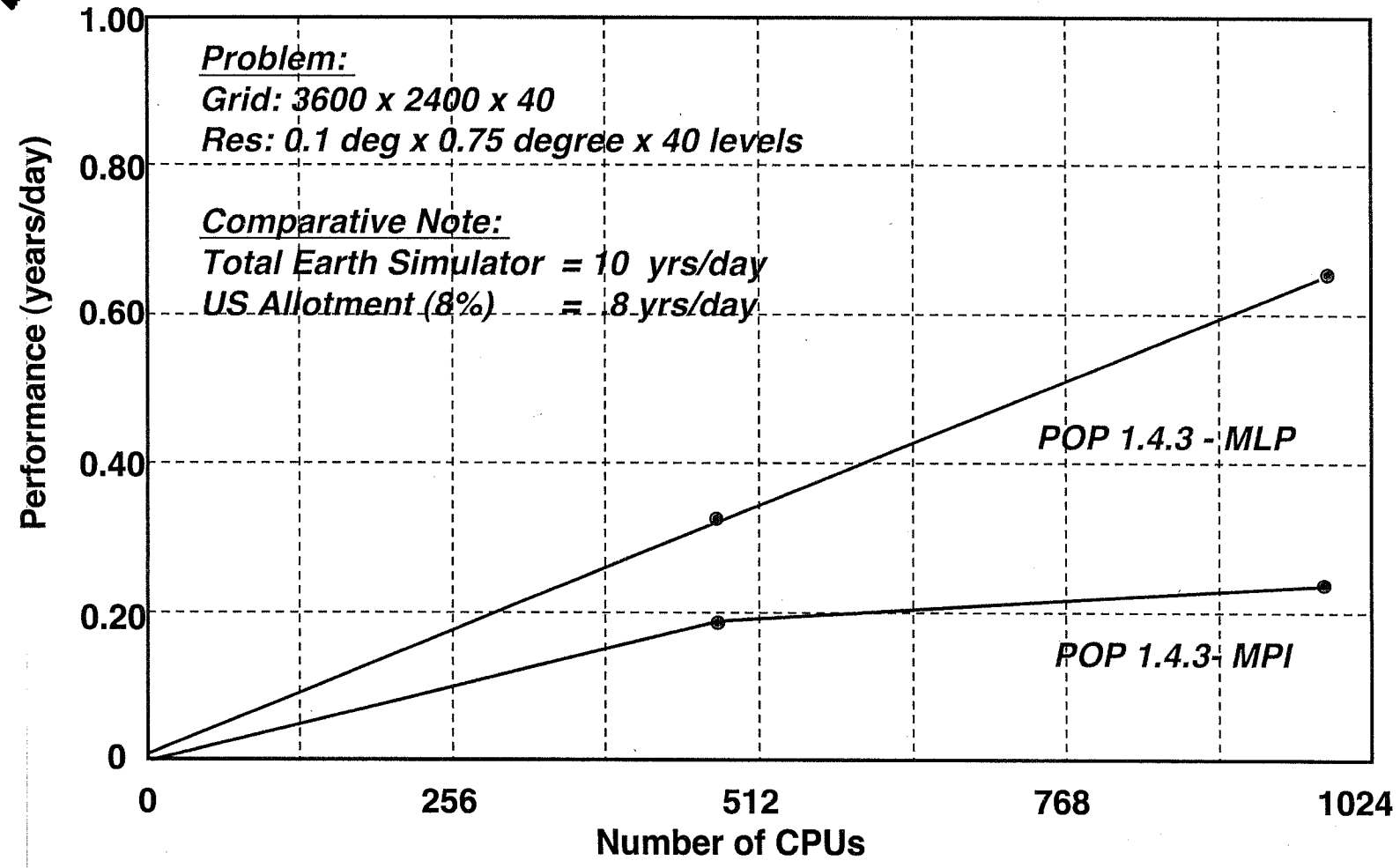
Parallel Ocean Modeling - POP

Information Sciences & Technology

Ames Research Center

POP Performance vs CPU Count (6/02)

System: 1024 CPU O3K 600 MHz (chapman)



Information Sciences & Technology
Ames Research Center

The Future of HPC at NASA

At a November, 2001 presentation to Dan Goldin we were asked:
Why haven't you ordered an even larger system?

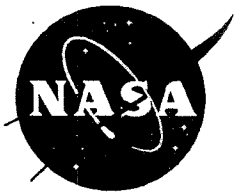
He asks this question because:

The scalable software techniques are in place
MLP parallelism at NASA Ames scales for real science - not kernels
MLP is a practical means for migrating codes in our lifetime - weeks

The hardware architecture exists
SSI NUMA architectures let scientists do science - NOT MPI DEBUG
Migration to larger SSI systems is transparent to users
MLP+NUMA supports "Cray programming model", the "standard"

Cost is more than competitive
SSI NUMA supports higher fractions of sustained vs peak.
Time to market of applications is years earlier

So what got Accelerated by NASA Ames?



Hardware:
SGI developed 256, 512, and 1024 CPU systems not in their plan

MLPlib:
A new very fast way of getting to large parallel scaling unlike MPI
Maintains shared memory "Cray programming model" for users

True Production codes and their Accelerated Performance:

Code	Speedup	Effort	Code size
OVERFLOW (CFD)	35X	1 man-year	100,000 lines
INS3D (CFD)	33X	2 man-years	50,000 lines
COSMOS core (MD)	35x	6 man-months	30,000 lines
DAS (Climate)	10x	2.5 man-years	350,000 lines
FVCCM3 (Climate)	40x	4 man-months	65,000 lines

Note: MPI versions of codes above took about 5-10x longer to build

THIS IS THE LEGACY OF MLP SHARED MEMORY PROGRAMMING

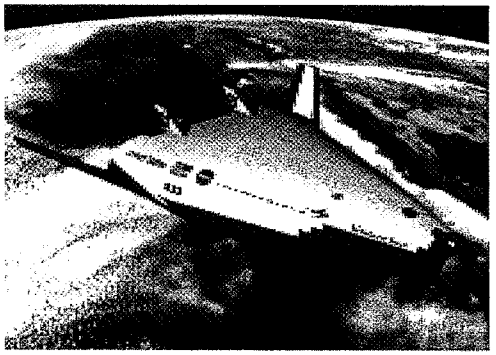
Information Sciences & Technology

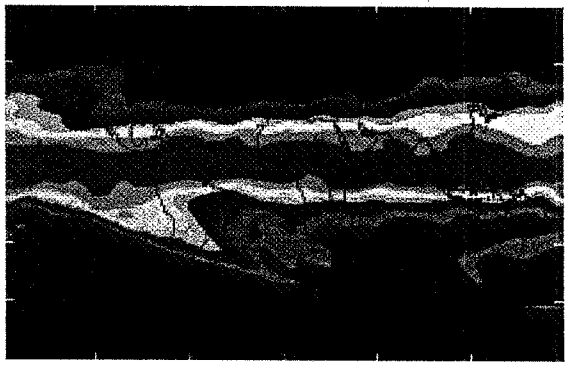
Ames Research Center

NASA

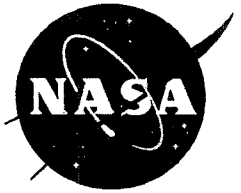
The Next Step in MLP Performance on SGI's Origin 1024p SSI System

Climate and CFD





Optimization Efforts - CFD/Climate Codes



The bulk of NASA,s NAS supercomputing cycles remain CFD related, with additional cycles devoted to climate and molecular modeling. This year, the NAS-ACL will focus on resolving remaining barriers to higher performance in CFD/Climate on the SGI shared memory Origin platforms. Major efforts will be:

- OVERFLOW-MLP - Sustain 250 GFLOP/s on 1024p system
  - Increase single CPU performance
  - Reduce overall memory size and traffic
  - Important because almost interactive electronic wind tunnel
- FVCCM3-MLP - Sustain 8000 days/day on 1024p system
  - Reduce communication burden
  - Increase single CPU performance
  - Important because same as Fujitsu 100 CPU result (ECMWF)

Information Sciences & Technology

Ames Research Center

Star Cluster - The Next Step in MLP