Effect of Tripling the Memory Bandwidth on the CRAY MTA-2

Wendell Anderson

Marco Lanzagorta

NRL Upgraded CRAY MTA-2

- 40 220Mhz processors
- 160 Gigabytes of Memory
- 128 hardware streams per processor
- 3 flops per cycle
- 1 terabyte of scratch disk

MTA-2 Improvements

- 10% increase in clock rate
- More paths into each processor
- Full network bandwidth

Measurement Tools

- UNIX date command
- F90 SYSTEM_INTRINSIC
- mtatop
- Dashboard
- Canal

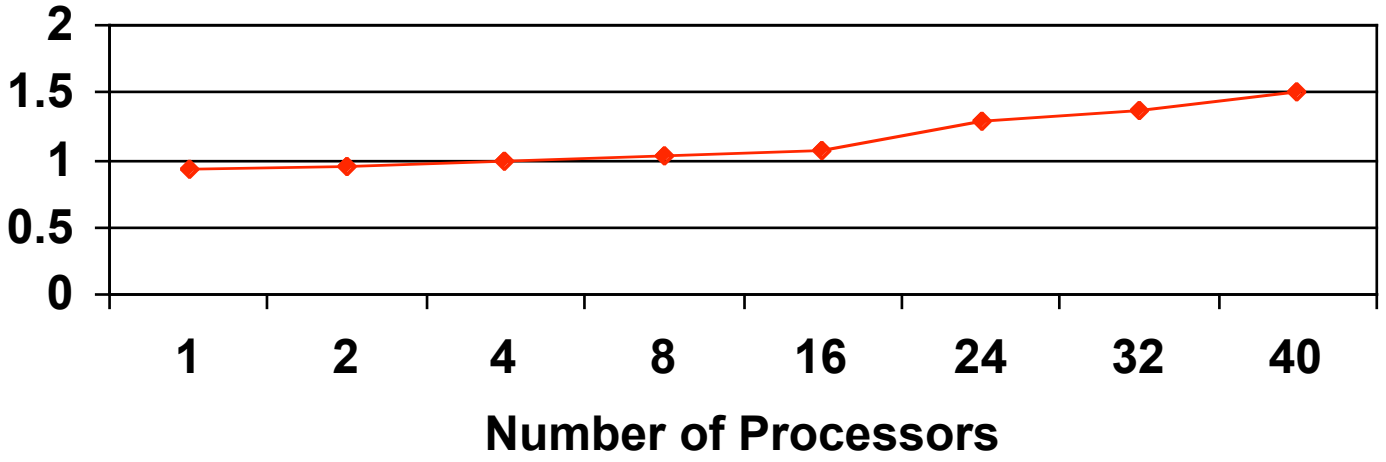
Flux

- Iterative solver over 4-D grid
- F90 code
- Four deep nested do loops
- FFT's over each grid axis

CUG 2004

Flux Timings

Ratio of Old Time to New TIme

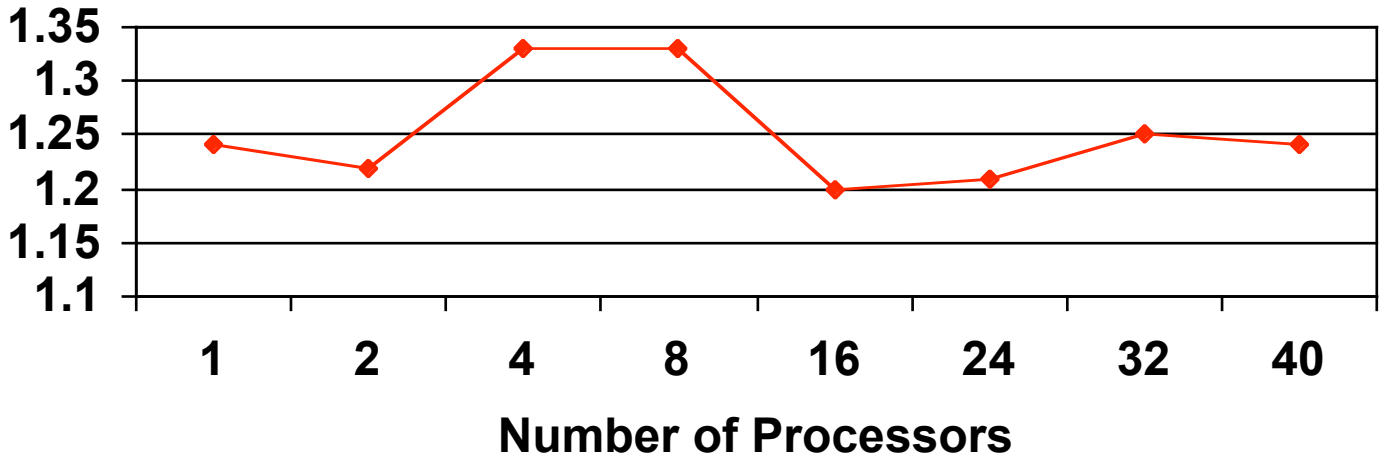


Alla

- Adaptive Mesh CFD
- Uses Fully Threaded Tree
- F90 code
- Overlap computations and I/O

Alla Timings

Ratio of Old Time to New TIme



Lanczos

- Low Temperature Quantum Systems
- Large Sparse Matrices
- Determine largest/smallest eigenvalues by Lanczos method
- Originally C++/MPI program
- Converted to F90 for MTA

Lanczos Real Canal Analysis

```
Do index = 1, Imax
     Y(index) = 0
 Do index2 = I(index)+1, I(index+1)
         Y(index) = Y(index)+M(index2)*Vvec(J(index2))
 enddo
enddo
```

Loop summary: 3 memory operations, 2 floating point operations
3 instructions, needs30 streams for full utilization
pipelined

Lanczos Complex Canal Analysis

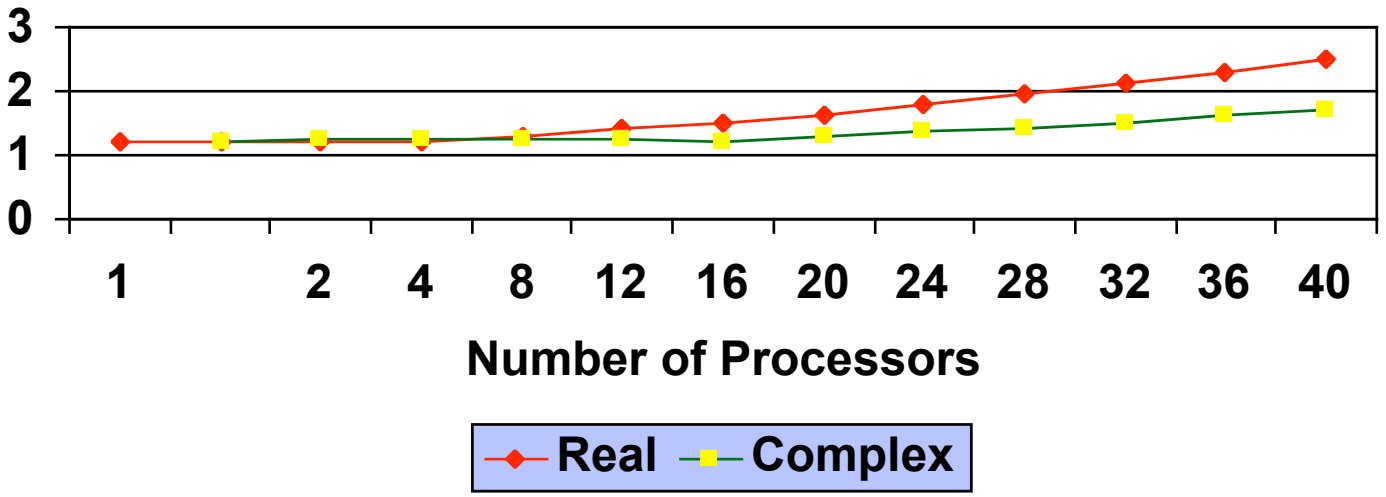
```
Do index = 1, Imax
    Y(index) = 0
 Do index2 = I(index)+1, I(index+1)
       Y(index) = Y(index)+M(index2)*Vvec(J(index2))
 enddo
enddo
```

Loop summary: 5 memory operations, 8 floating point operations
7 instructions, needs 56 streams for full utilization
pipelined
1 instructions added to satisfy dependences

CUG 2004

Lanzcos Speedups

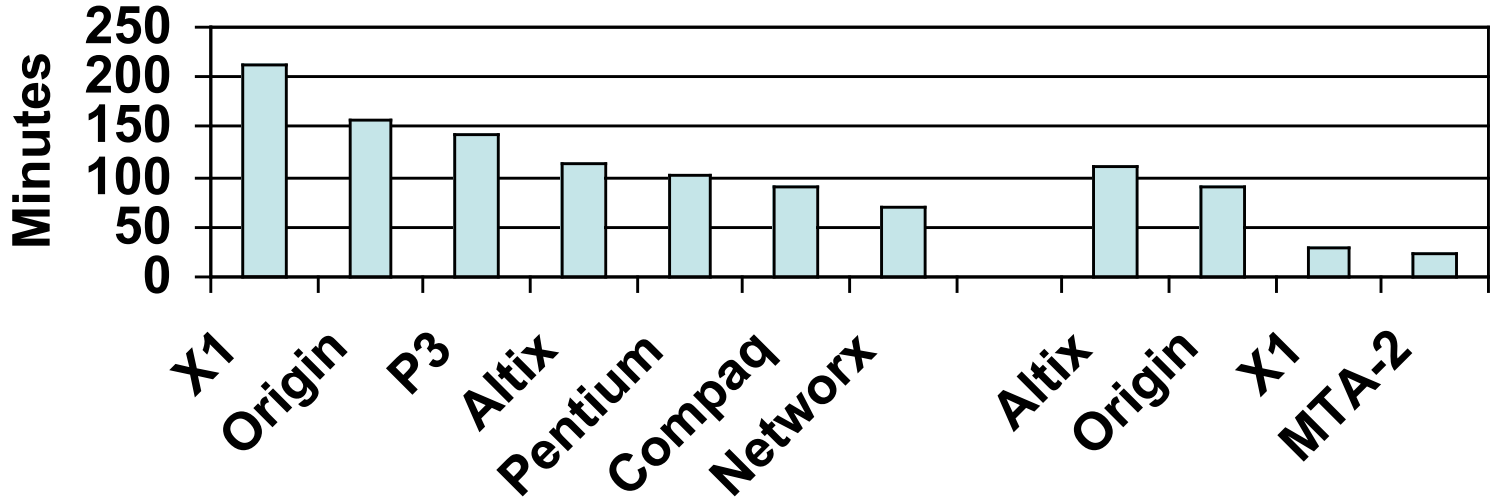
Ratio of Old Time to New TIme



CUG 2004

Lanczos Timings

Wall Clock Time for 16 processors



Causal

- 2-D FDTD Acoustic Wave Propagation
- Includes Dispersive Medium everywhere
- F90 code
- Inner product calculations dominate

Causal

- Achieves 7.5 Giga Memory Ops per second
- 31.4 Scaling from 1 to 40 processors

Causalrd2

- Dispersion only on a small subset of the grid
- Wave equation stencil dominates
- Load balancing a factor
- 3.6 Giga Memory Ops
- 30 speedup going from 1 to 40 processors

Causalrd2 Canal Analysis

```
fsq=cvelsq(i,j)*dx*dx
grad=fsq*(-60.0*u(i,j,i1)+16.0*(u(i+1,j,i1)+u(i,j+1,i1)
          +u(i-1,j,i1)+u(i,j-1,i1))-(u(i+2,j,i1)+u(i,j+2,i1)
          +u(i-2,j,i1)+u(i,j-2,i1)))/(12.0*dx*dz)
    +fsq*fsq*(20.0*u(i,j,i1)-8.0*(u(i+1,j,i1)+u(i-1,j,i1)
              +u(i,j+1,i1)+u(i,j-1,i1))+2.0*(u(i+1,j+1,i1)
              +u(i-1,j+1,i1)+u(i+1,j-1,i1)+u(i-1,j-1,i1))
              +u(i+2,j,i1)+u(i-2,j,i1)+u(i,j+2,i1)+u(i,j-2,i1))
              /(12.0*dx*dx*dz*dz)
 u(i,j,i0)=2.*u(i,j,i1)-u(i,j,i2)+grad
expecting 36566209 iterations
Loop summary: 44 instructions, 40 floating point operations
        16 loads, 2 stores, 15 reloads, 0 spills, 1 branches, 0 calls
```

Causalrd2 Canal Analysis

```
grad=e1(i,j)*u(i,j,i1)
      +e2(i,j)*(u(i+1,j,i1)+u(i,j+1,i1)+u(i-1,j,i1)+u(i,j-1,i1))
      + e3(i,j)*(u(i+2,j,i1)+u(i,j+2,i1)+u(i-2,j,i1)+u(i,j-2,i1))
      +e4(i,j)*(u(i+1,j+1,i1)+u(i-1,j+1,i1)
      +u(i+1,j-1,i1)+u(i-1,j-1,i1))
   u(i,j,i0)=2.0*u(i,j,i1)-u(i,j,i2)+grad
```

expecting 36566209 iterations
Loop summary: 19 memory operations, 19 floating point operations 21 instructions, needs 106 streams for full utilization
pipelined
2 instructions added to reduce register requirements

CUG 2004

Summary

Per cent Utilization