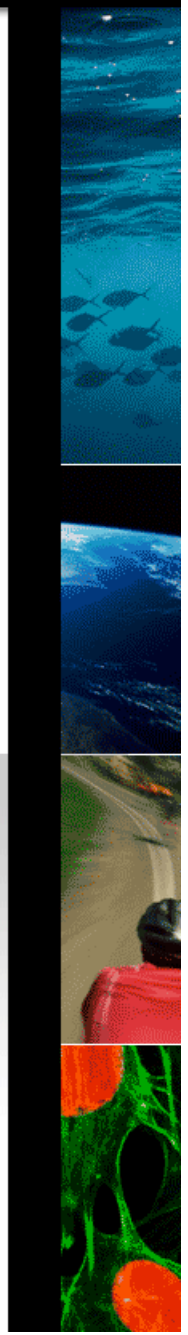
The Supercomputer Company

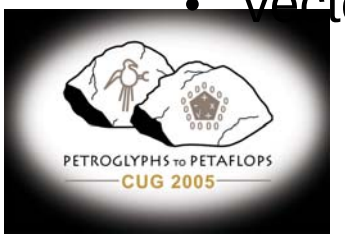
Comparisons of X1 and X1E

John Levesque
Director
Cray Supercomputer Center of Excellence



X1 After Two Years…

- many improvements in PE, OS, optimization knowledge base
- typical % peak (12.8 GF) on decent vector code 15-40%
- typical speedup over (5.2 GF) Power 4 ~ 7-30x
- finding X1 often limited more by CPU and memory *latencies* rather than memory *bandwidth* (i.e., % time E < -- > M wires busy < 50% total) ➔ good for Cray X1E (lower bandwidth and lower latency than X1)
- X1 network:
  - plenty of bandwidth (except GUPS)
  - MPI latency good
  - CAF/UPC exceptional for collective operations
- customers are doing problems on X1 they could never do before
  - vectorization can be work, but it is a demonstrable path to HPC



Cray X1 Multi-Streaming Processor

12.8 Gflops (64 bit)

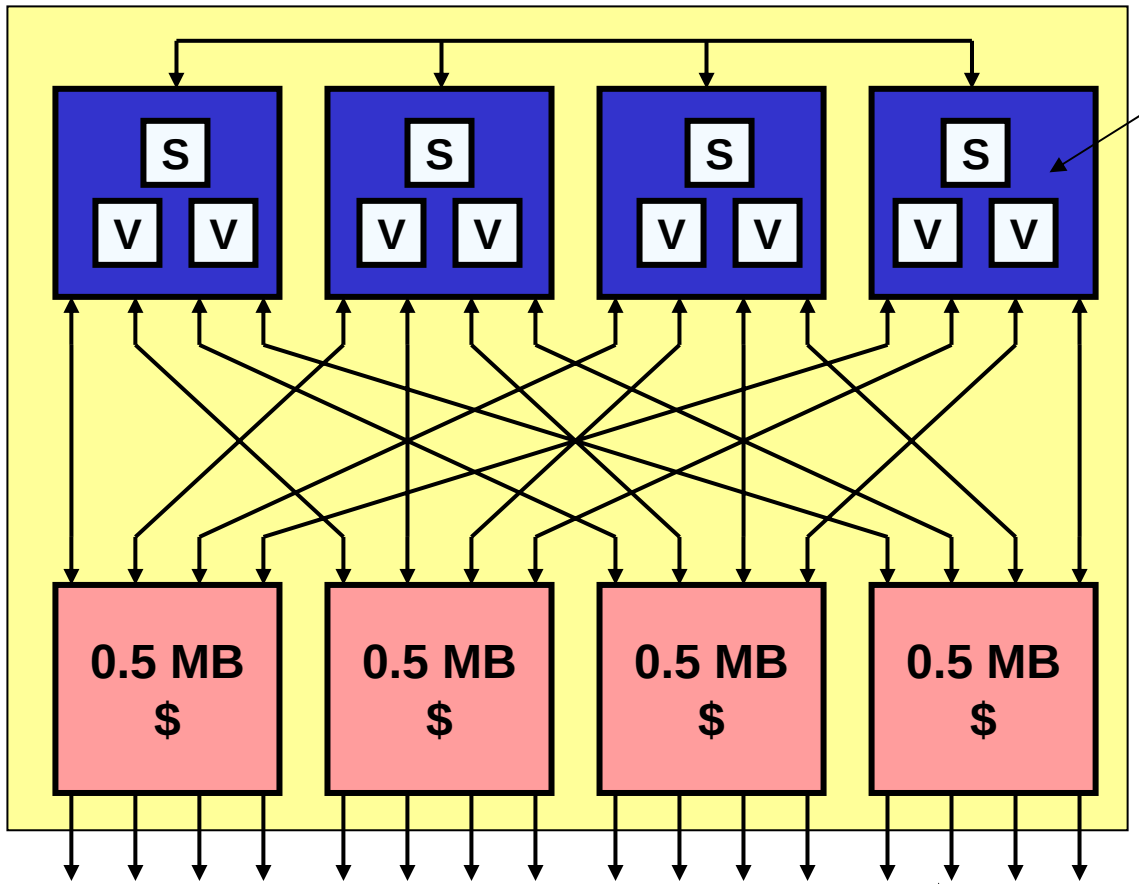
25.6 Gflops (32 bit)

51 GB/s ↑

25-41 GB/s ↓

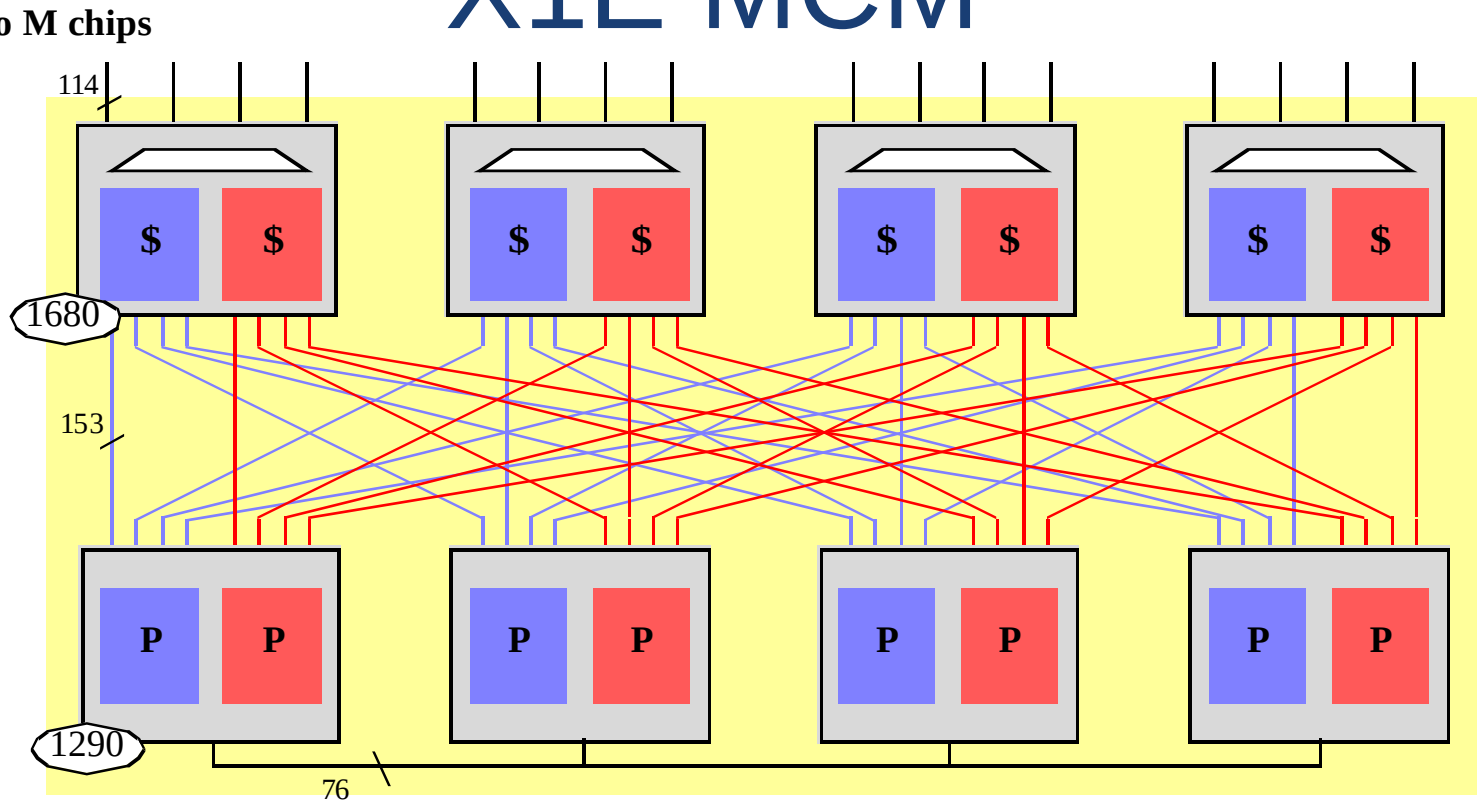
2 MB Ecache

At frequency of 400/800 MHz



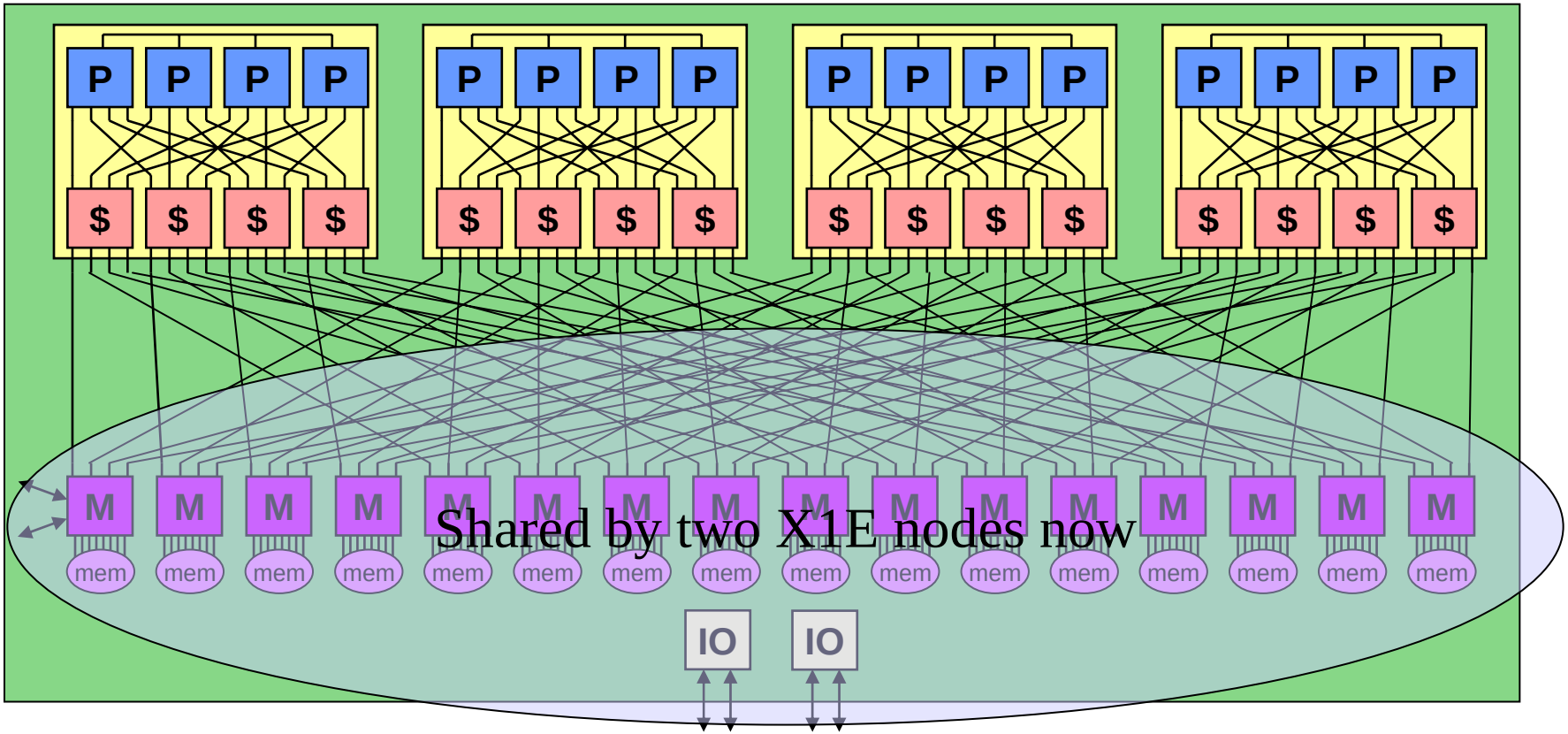
To local memory and network:

X1E MCM



- Re-implement P and E chips in 0.08mm IC technology
- Place two MSPs on each MCM
  ⇒ Double the processor density (2 –4 MSP nodes/module)
- MCM frequency increase (400 MHz – 565 MHz, 1.41x)

Cray X1 Node



Inter node network:

- 2 ports per M chip
- 1.6 GB/s full duplex per link
- 102.4 Gbyte/s to the network

I/O connections:

- 4 ports per node
- 1.2 GB/s full duplex per link

Local memory:

- 200 GB/s peak bw
- 16-32 GB per node

PETROGLYPHS TO PETAFLOPS
CUG 2005

HPCC Comparisons

Computer System / Processor Type and Speed, Interconnec / Affiliation / Submission Date/ Version	SIZE	BASE	Procs	HPL (system total)	HPL (per CPU)	PTRANS	Random Access	*STREAM - per CPU		G-FFTE	EP_DGEM M	Bandwidth		Latency	
							Global CUMM	Triad	Triad CUMM	Global CUMM	per cpu	RR / CPU	Random Ring CUMM	Random Ring/CPU	RR SM BW
Cray X1e X1 MSP 2-D Torus	125555	opt	124	1.31	10.56	56.4	1.77	12.74	1579.76	15.1	7.77	0.592	73.41	13.9	71.37
Cray X1e X1 MSP 1 2-D Torus		opt	124	1.19	9.60	56.2	1.75	12.9	1599.6	14.74	7.62	0.593	73.53	15.4	64.42
Cray X1 X1 MSP 0.8 2-D Torus		opt	60	0.5633	9.39	38.25	0.821	21.2	1272	6.96	8.52	1.07	64.20	17.99	26.68
Cray X1 X1 MSP 0.8 2-D Torus	95555	opt	64	0.595541	9.31	44.42	2.17	22.03	1409.92	14.3	11.3	0.939	60.10	14.8	34.59
Cray X1 X1 MSP 0.8 2-D Torus	125555	opt	124	0.9458	7.63	22.8	3.89	22.5	2790	18.2	8.46	0.863	107.01	15.9	62.39

Kernel Timings

- Run on a very busy system AHPCRC
- Run in SSP mode, not streaming
- Influenced by other tasks running on the same node
- From excellent book “Guidebook to Fortran on Supercomputers”

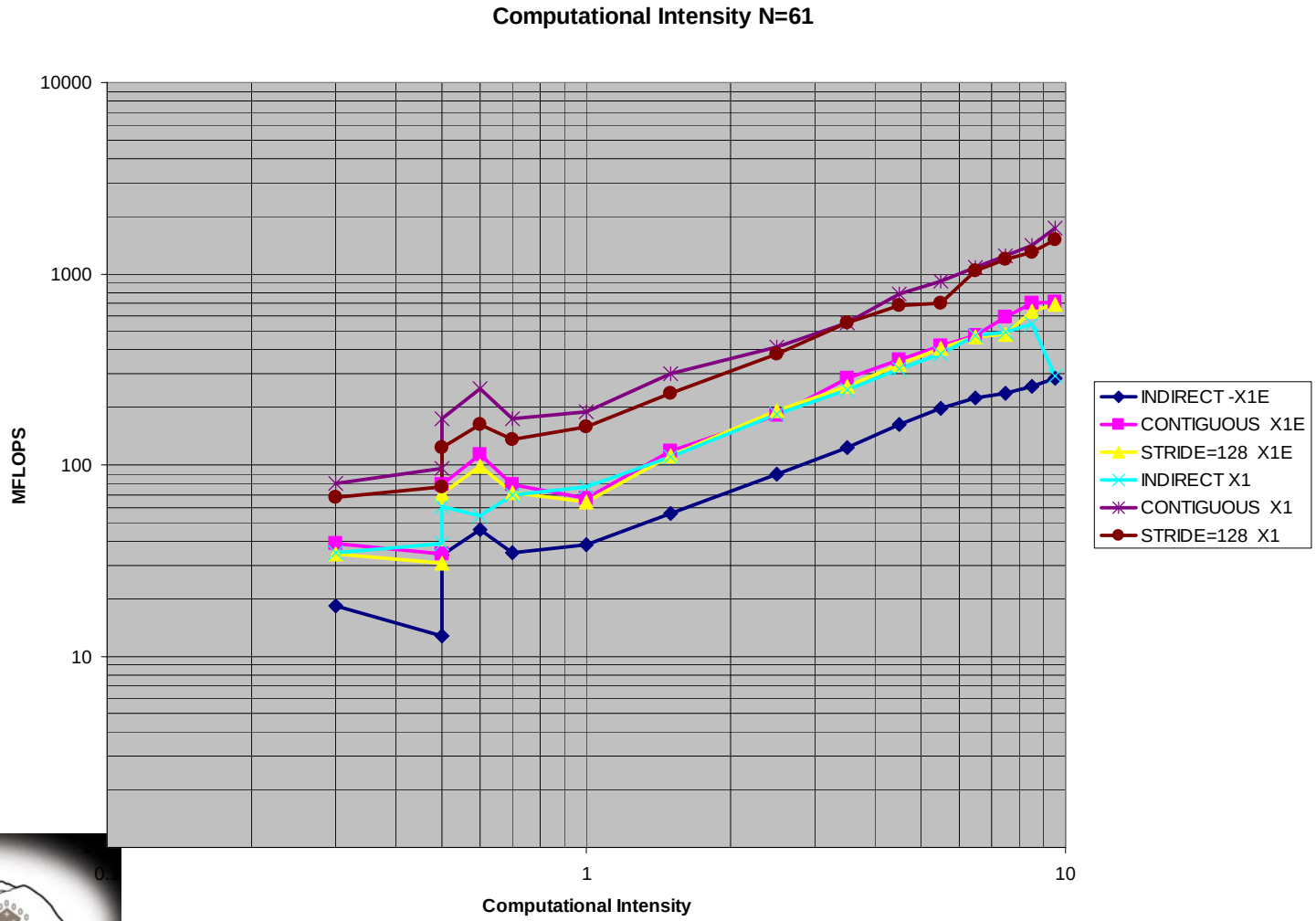
Computational Intensity

- Divide the number of Vector operations/Vector Operands

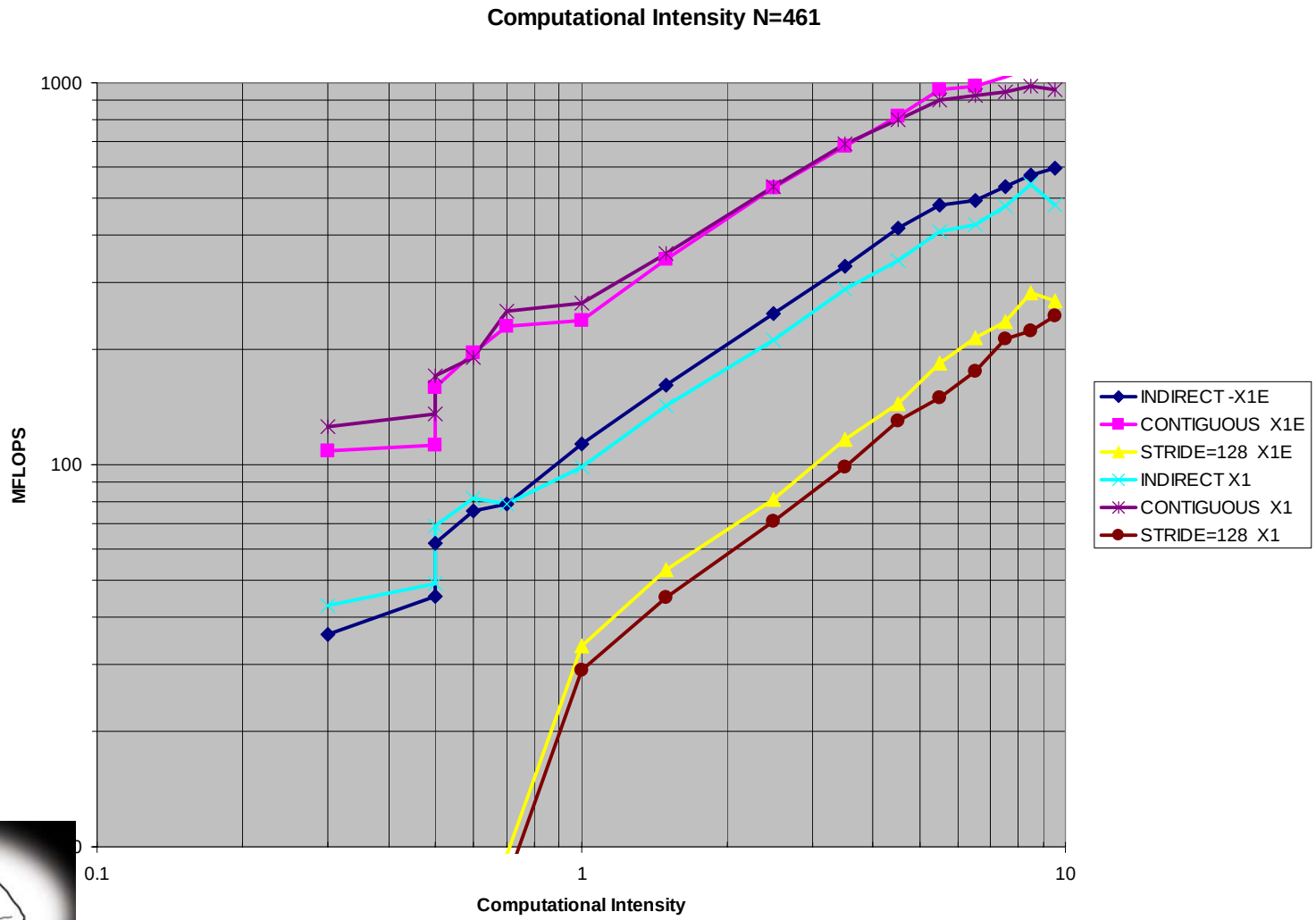
$$A(I) = B(I) + C(I) \qquad CI = 1/3$$

$$A(I) = C0 + B(I)*(C1+B(I)*(C2+C3*B(I))) \qquad CI = 6/2 = 3$$

Short Vector Length N = 61



Longer Vector Length n = 461



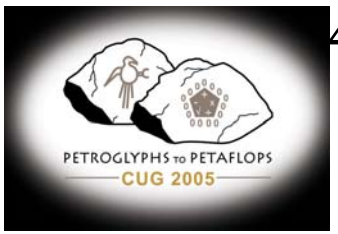
Kernel Investigation

Memory Access Kernels

CRAY USER GROUP INCORPORATED

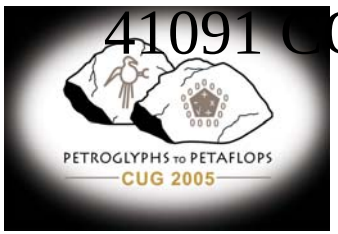
Loop with poor Memory Access

```
      COMMON A(8,8,IIDIM,8),B(8,8,iidim,8)
      DO 41090 K = KA, KE, -1
        DO 41090 J = JA, JE
          DO 41090 I = IA, IE
            A(K,L,I,J) = A(K,L,I,J) - B(J,1,i,k)*A(K+1,L,I,1)
     *    - B(J,2,i,k)*A(K+1,L,I,2) - B(J,3,i,k)*A(K+1,L,I,3)
     *    - B(J,4,i,k)*A(K+1,L,I,4) - B(J,5,i,k)*A(K+1,L,I,5)
41090 CONTINUE
```

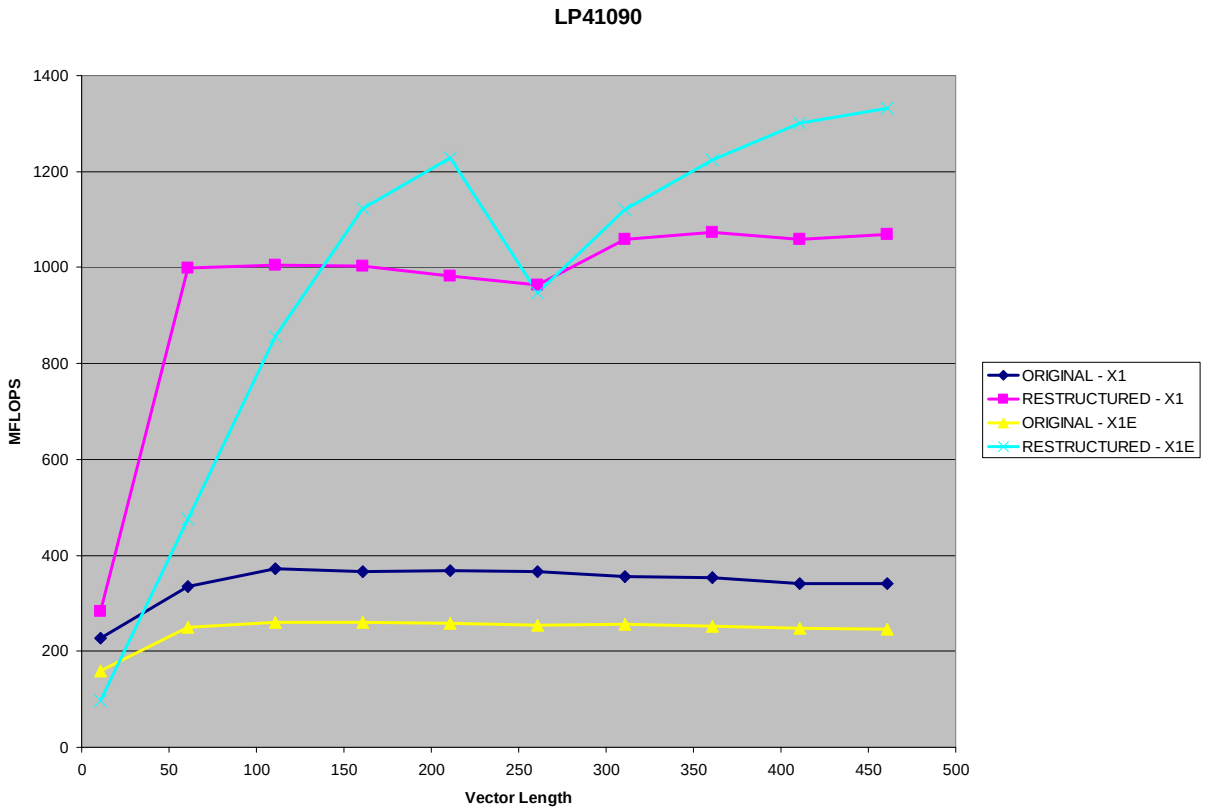


Loop with better Memory Access

```
      COMMON AA(IIDIM,8,8,8),BB(IIDIM,8,8,8)
      DO 41091 K = KA, KE, -1
       DO 41091 J = JA, JE
        DO 41091 I = IA, IE
          AA(I,K,L,J) = AA(I,K,L,J) - BB(I,J,1,K)*AA(I,K+1,L,1)
     *    - BB(I,J,2,K)*AA(I,K+1,L,2) - BB(I,J,3,K)*AA(I,K+1,L,3)
     *    - BB(I,J,4,K)*AA(I,K+1,L,4) - BB(I,J,5,K)*AA(I,K+1,L,5)
41091 CONTINUE
```

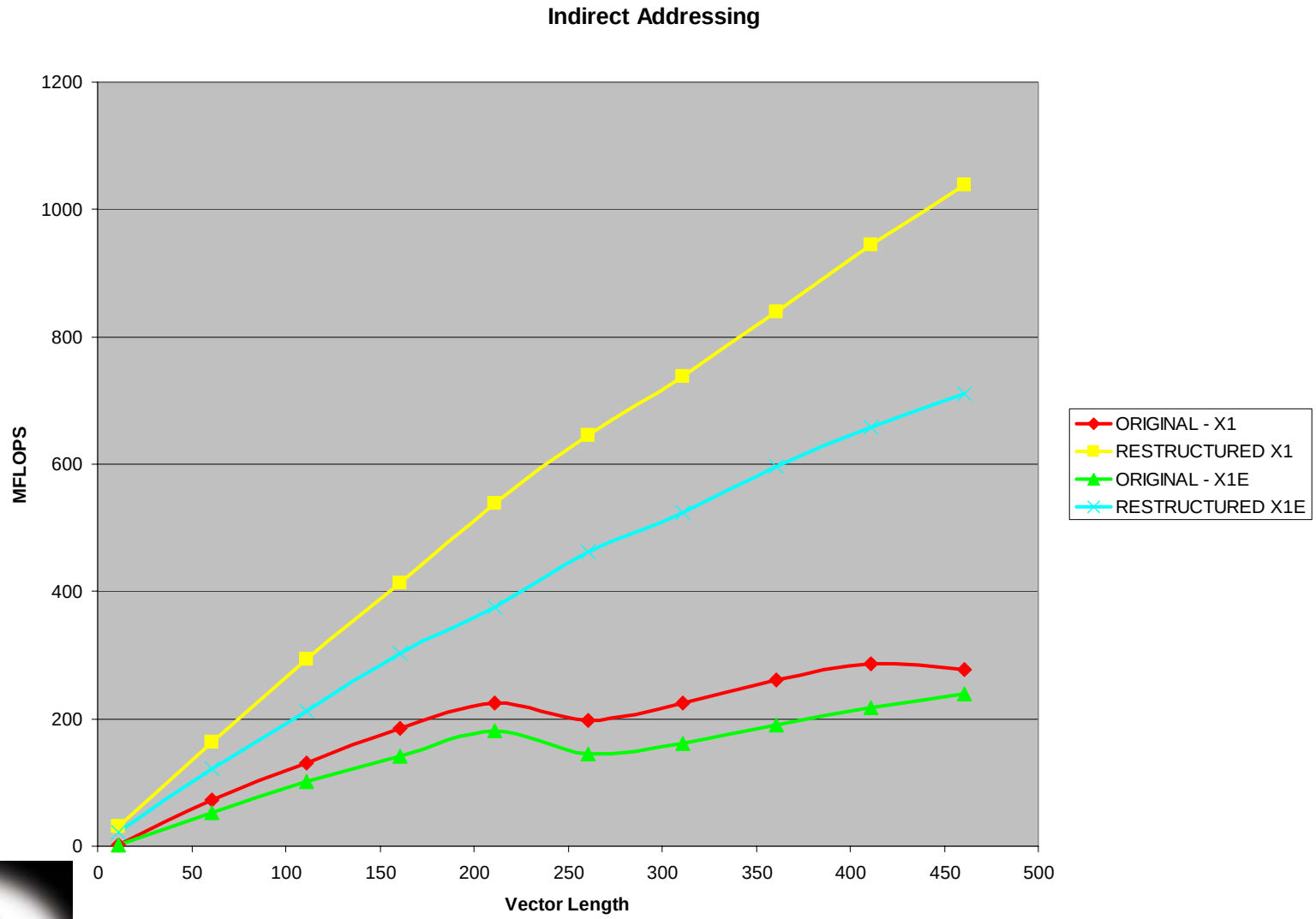


X1E is better when memory Access methods are improved



Indirect Addressing

```
C      THE ORIGINAL
         DO 43070 I = 1, N
           A(IA(I)) = A(IA(I)) + C0 * B(I)
43070 CONTINUE
C      THE RESTRUCTURED
CDIR$ IVDEP
         DO 43071 I = 1, N
           A(IA(I)) = A(IA(I)) + C0 * B(I)
43071 CONTINUE
```


Indirect Addressing
MFLOPS
1200
1000
800
600
400
200
0
0
50
100
150
200
250
300
350
400
450
500
Vector Length
ORIGINAL - X1
RESTRUCTURED X1
ORIGINAL - X1E
RESTRUCTURED X1E

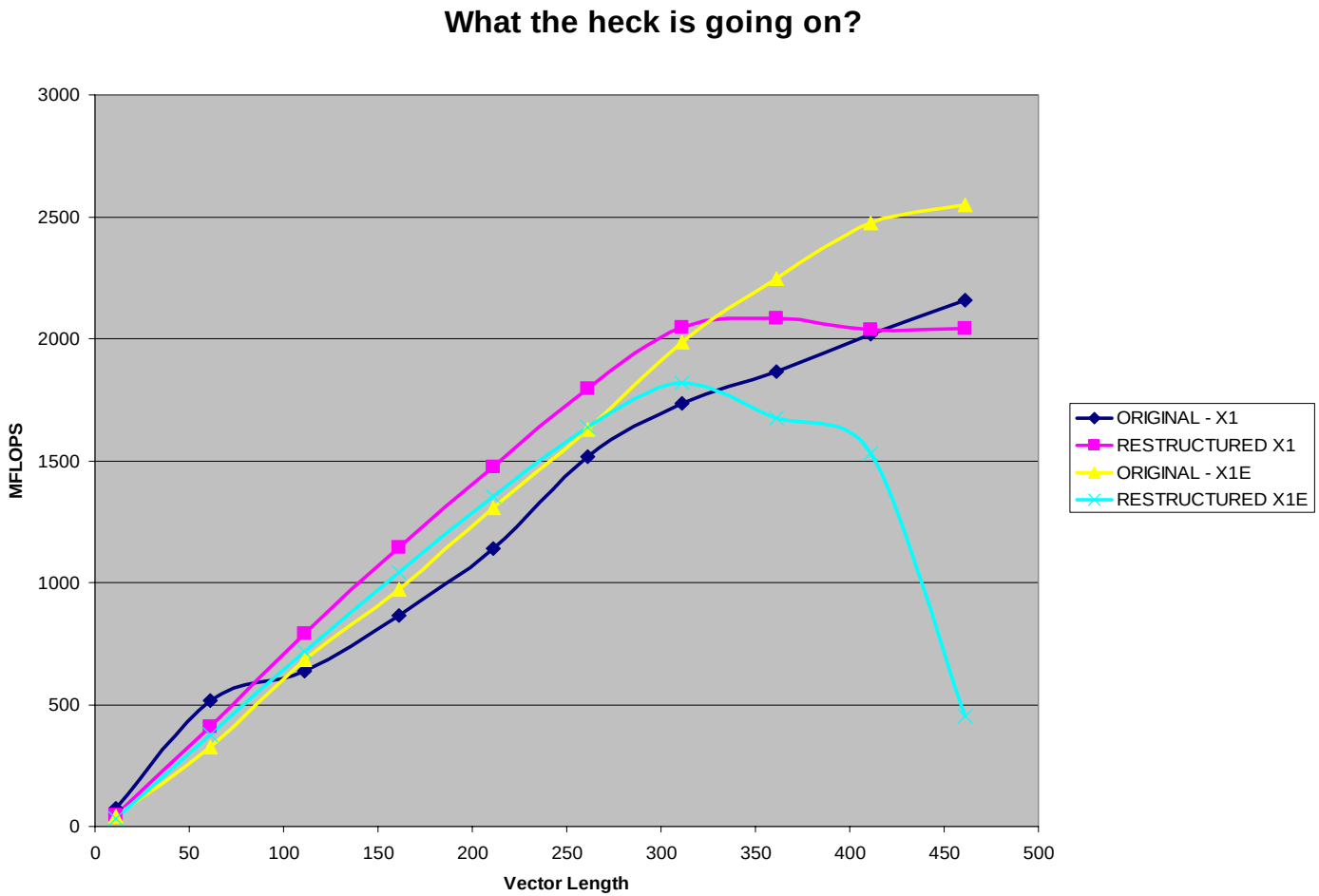

PETROGLYPHS TO PETAFLOPS
CUG 2005

A Bandwidth Problem

```
C      THE ORIGINAL

       DO 44060 I = 1, N
        A(I) = 0.0
        DO 44060 J = 1, I
         A(I) = A(I) + B(I,J) * C(J,I)
44060 CONTINUE
```

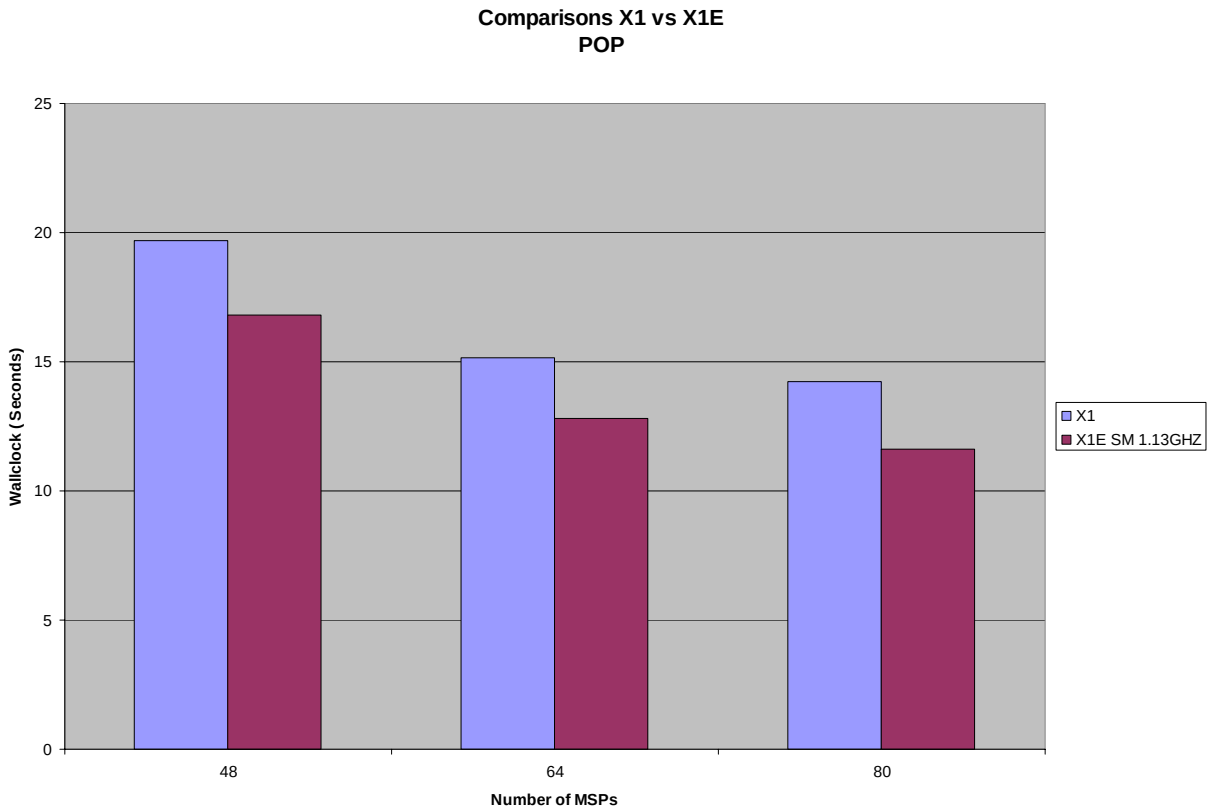
A Bandwidth Problem

```
C      THE RESTRUCTURED

       DO 44061 I = 1, N

          A(I) = 0.0

44061 CONTINUE

       DO 44062 J = 1, N

         DO 44062 I = J, N

            A(I) = A(I) + B(I,J) * C(J,I)

44062 CONTINUE
```

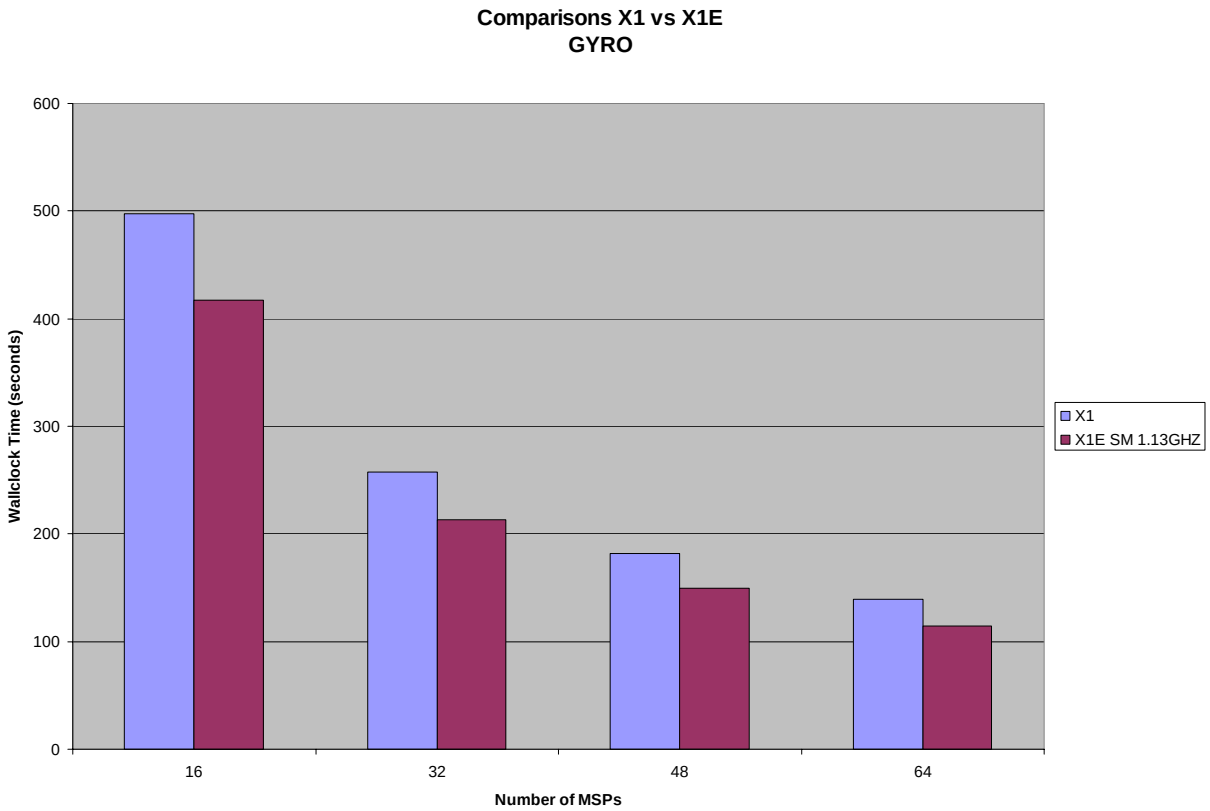

What the heck is going on?
MFLOPS
Vector Length
0
500
1000
1500
2000
2500
3000
0
50
100
150
200
250
300
350
400
450
500
ORIGINAL - X1
RESTRUCTURED X1
ORIGINAL - X1E
RESTRUCTURED X1E


PETROGLYPHS TO PETAFLOPS
CUG 2005

Parallel Ocean Model

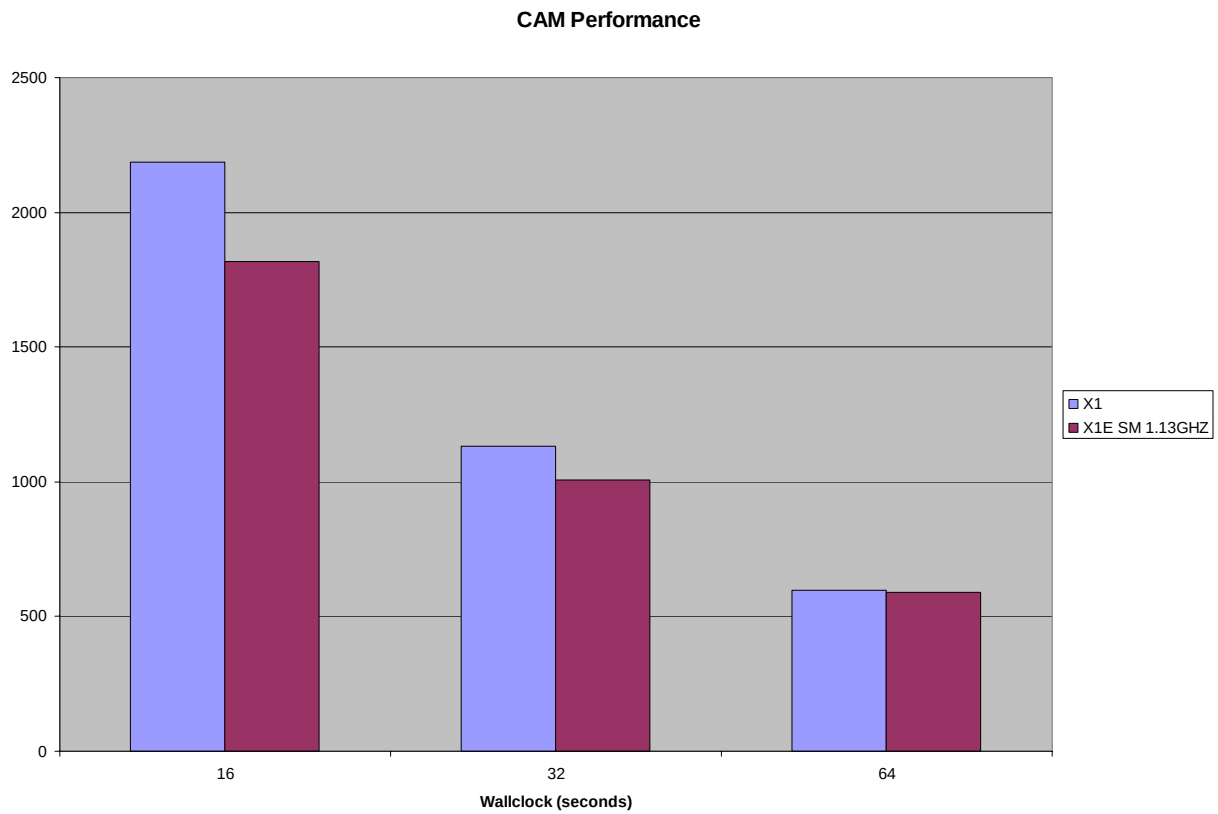


GYRO

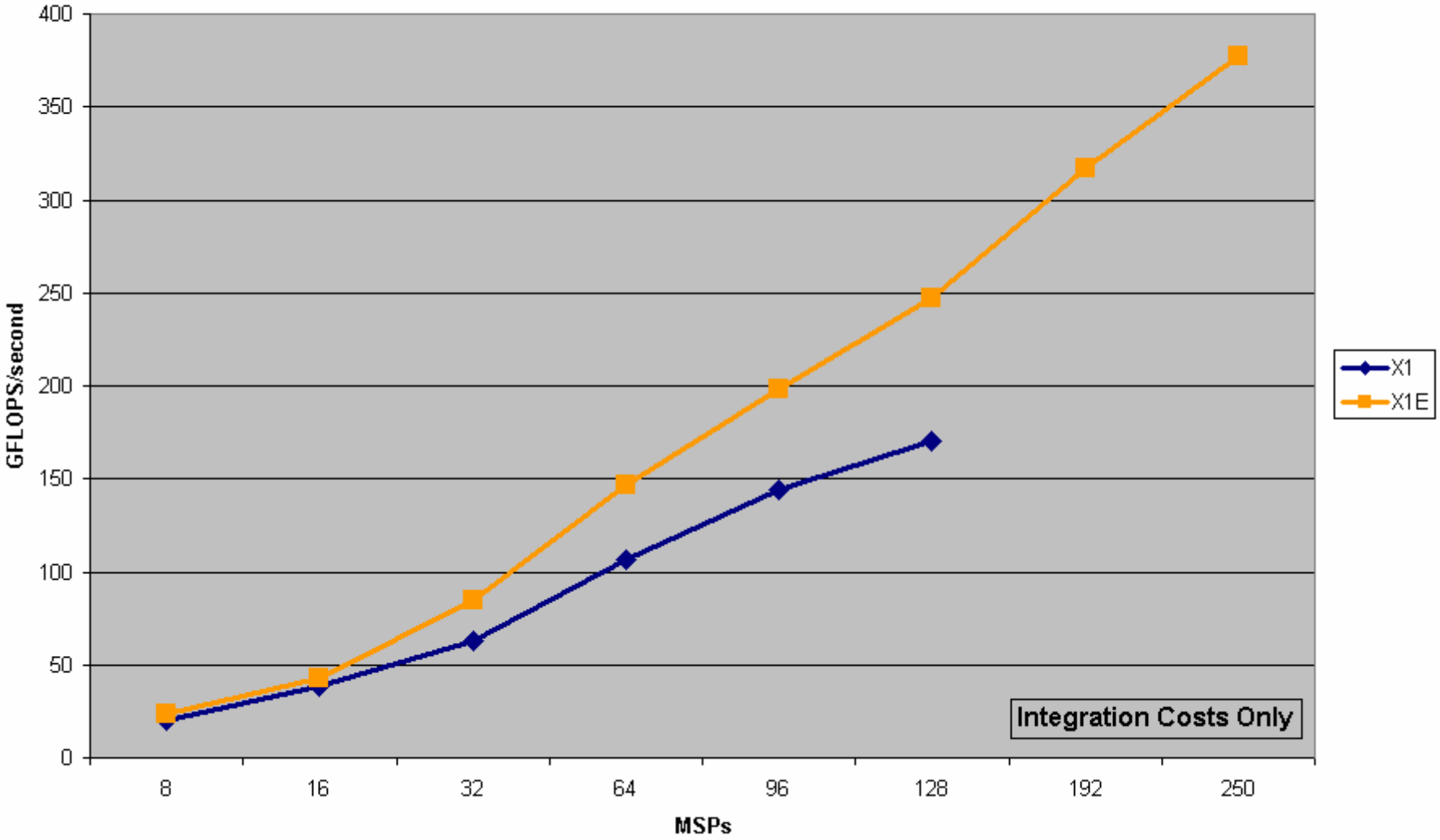

Comparisons X1 vs X1E
GYRO
Wallclock Time (seconds)
600
500
400
300
200
100
0
16
32
48
64
Number of MSPs
X1
X1E SM 1.13GHZ


PETROGLYPHS TO PETAFLOPS
CUG 2005

Community Atmospheric Model



WRF NewConus Benchmark X1/X1E



Summary

- Individual Kernels show X1 superior due to bandwidth limitations
  - Much on single processor run can be attributed to other jobs running on same MSP
- On multi-processor runs X1E shows clear advantage over X1
- Profile your X1 codes to see if certain computational kernels are using appropriate more time on the X1E
- Generally unless the app is memory bound X1E runs 1.05 – 1.1 times faster than the X1 and remember there are twice as many of them