Architectural Implications of a Family of Irregular Applications

David O'Hallaron
Jonathan Shewchuk
Thomas Gross

School of Computer Science
Carnegie Mellon University

droh@cs.cmu.edu

Motivating question

What performance will be needed from the communication systems of future computers so that they run unstructured finite element simulations efficiently?

Why unstructured finite element simulations

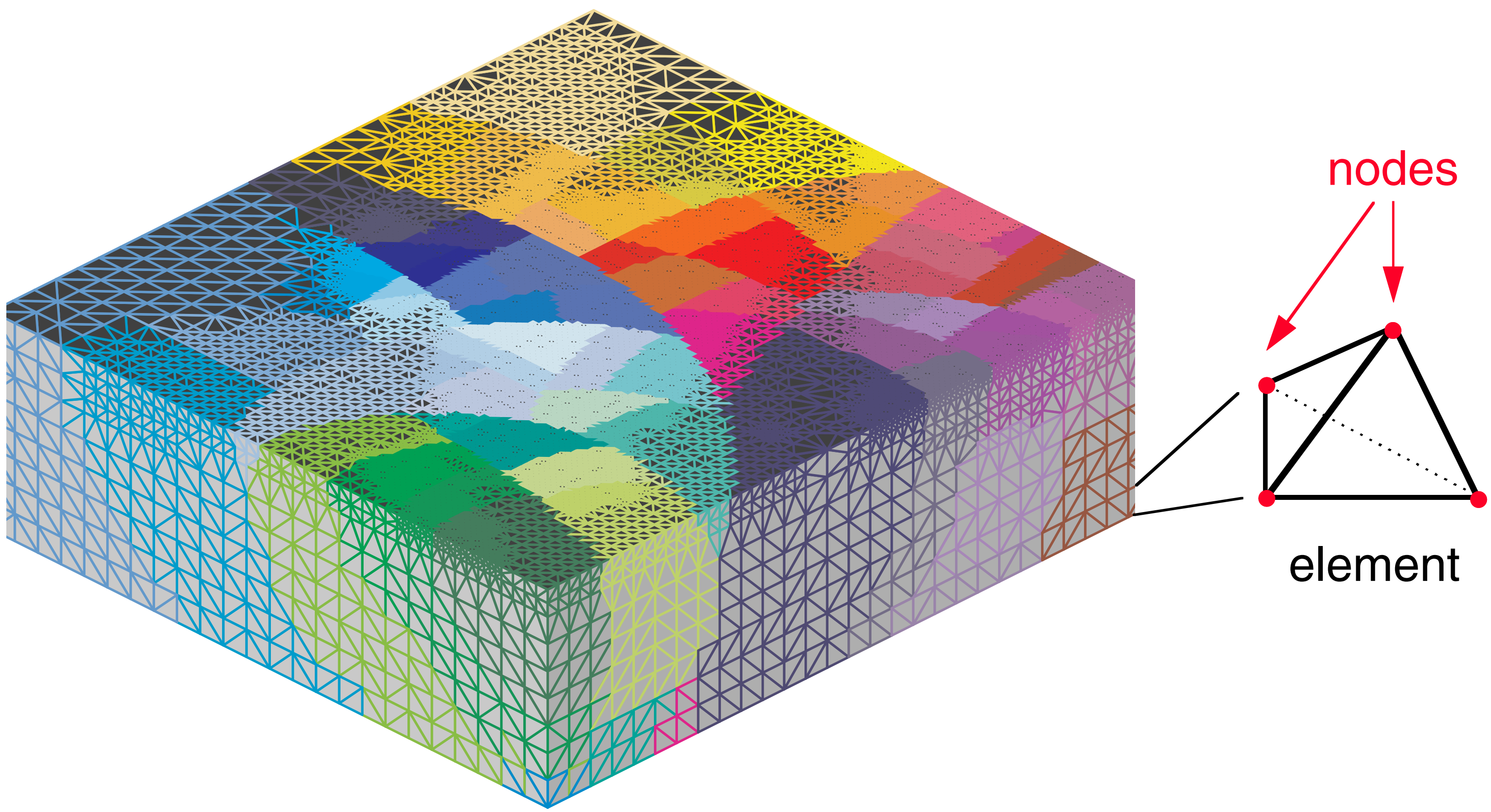
- Irregular memory access and communication patterns.
  - irregular -> cannot be predicted at compile time.
- Important method for physical simulations.
- Big user of cycles.
- Access to real applications as members of the Quake project.
  - NSF Grand Challenge Application Group

Summary of Conclusions

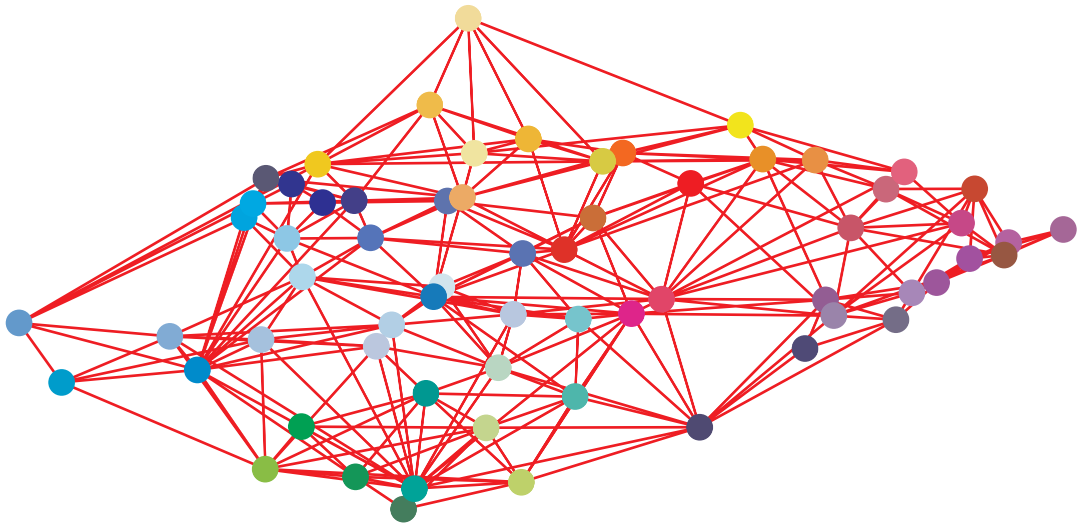
For future systems running unstructured finite element codes:

- Bisection bandwidth: not an issue.
- Sustained PE bandwidth: tough but doable.
- Block latency: might be a problem.

Partitioned unstructured finite element mesh of San Fernando



Communication graph



Vertices: processors
Edges: communications

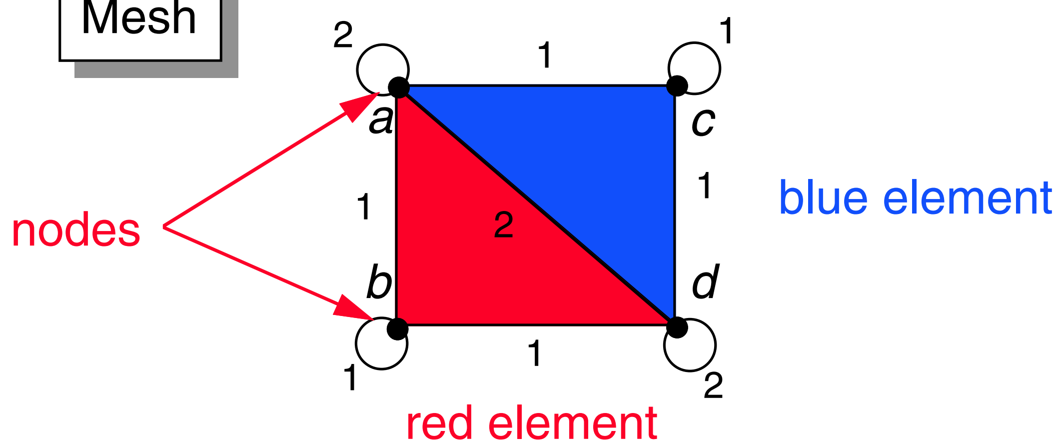
The San Fernando Meshes

sfk: “sf”=“San Fernando”, “k”=wavelength (secs)

	sf10	sf5	sf2	sf1
nodes	7,294	30,169	378,747	2,461,694
elements	35,025	151,239	2,067,739	13,980,162
edges (nonzeros)	44,922	190,377	2,509,064	16,684,112
MBytes	8	36	454	2,954

Sparse Matrix Vector Product (SMVP)

Mesh



Sparse matrix

$$K = \begin{array}{c c} \begin{matrix} a & b & c & d \end{matrix} & \\ \begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 1 & 2 \end{bmatrix} & \begin{matrix} a \\ b \\ c \\ d \end{matrix} \end{array}$$

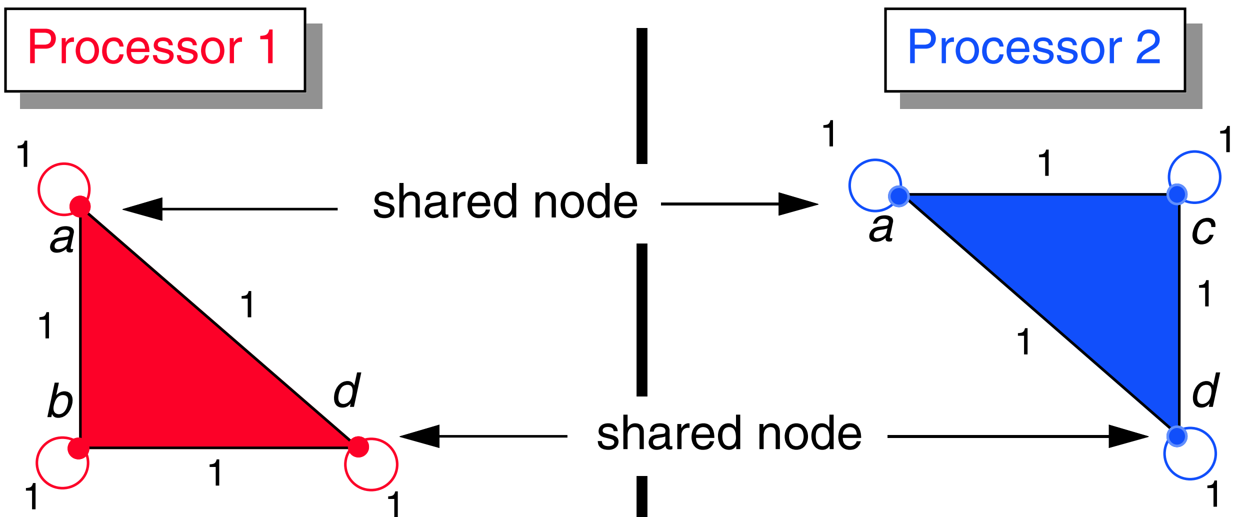
vector

$$x = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \begin{matrix} a \\ b \\ c \\ d \end{matrix}$$

SMVP

$$y = Kx = \begin{bmatrix} 12 \\ 6 \\ 6 \\ 12 \end{bmatrix} \begin{matrix} a \\ b \\ c \\ d \end{matrix}$$

Problem partitioning for parallel SMVP



Processor 1

Sparse matrix

vector

$$K_1 = \begin{array}{c} \begin{matrix} a & b & c & d \end{matrix} \\ \begin{bmatrix} 1 & 1 & - & 1 \\ 1 & 1 & - & 1 \\ - & - & - & - \\ 1 & 1 & - & 1 \end{bmatrix} \end{array} \begin{matrix} a \\ b \\ c \\ d \end{matrix} \qquad x_1 = \begin{bmatrix} 2 \\ 2 \\ - \\ 2 \end{bmatrix} \begin{matrix} a \\ b \\ c \\ d \end{matrix}$$

Processor 2

Sparse matrix

vector

$$K_2 = \begin{array}{c} \begin{matrix} a & b & c & d \end{matrix} \\ \begin{bmatrix} 1 & - & 1 & 1 \\ - & - & - & - \\ 1 & - & 1 & 1 \\ 1 & - & 1 & 1 \end{bmatrix} \end{array} \begin{matrix} a \\ b \\ c \\ d \end{matrix} \qquad x_2 = \begin{bmatrix} 2 \\ - \\ 2 \\ 2 \end{bmatrix} \begin{matrix} a \\ b \\ c \\ d \end{matrix}$$

Parallel SMVP: Compute phase

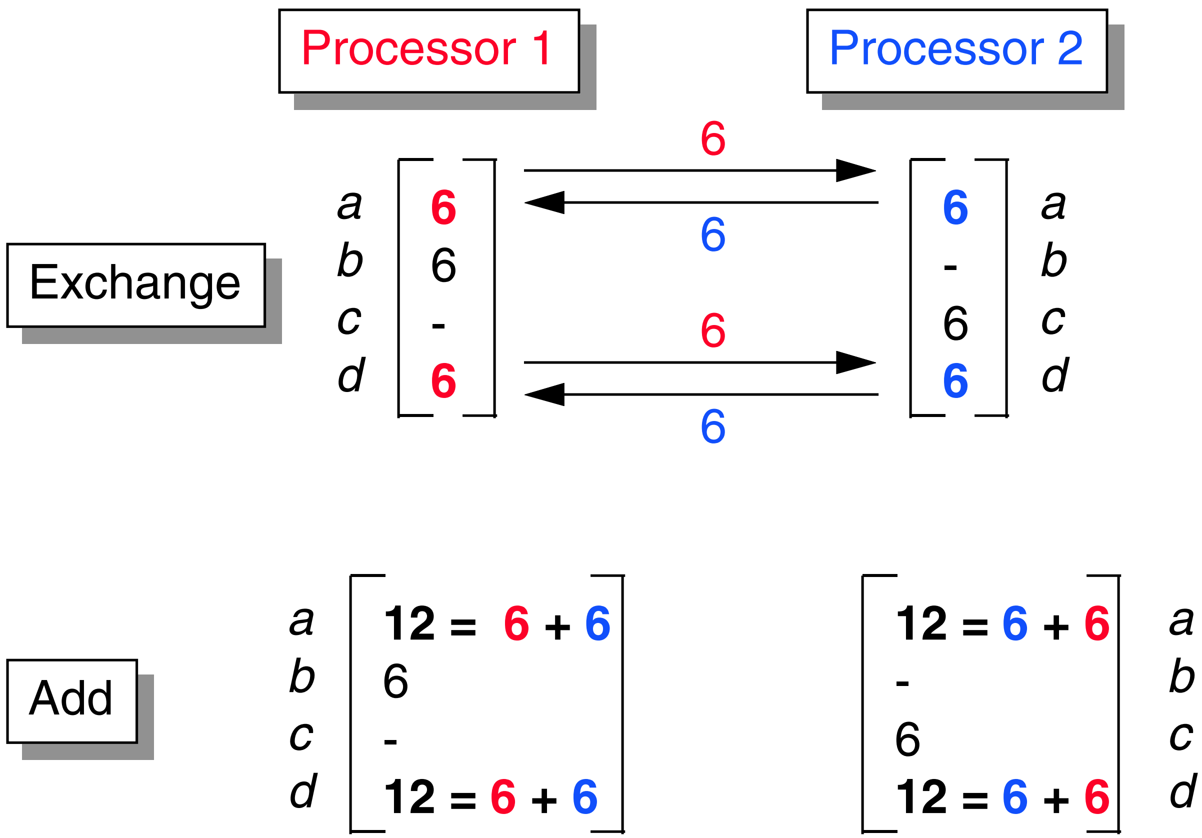
Processor 1

$$y_1 = K_1 x_1 = \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 1 & - & 1 \\ 1 & 1 & - & 1 \\ - & - & - & - \\ 1 & 1 & - & 1 \end{bmatrix} \end{array} \begin{bmatrix} 2 \\ 2 \\ - \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \\ - \\ 6 \end{bmatrix}$$

Processor 2

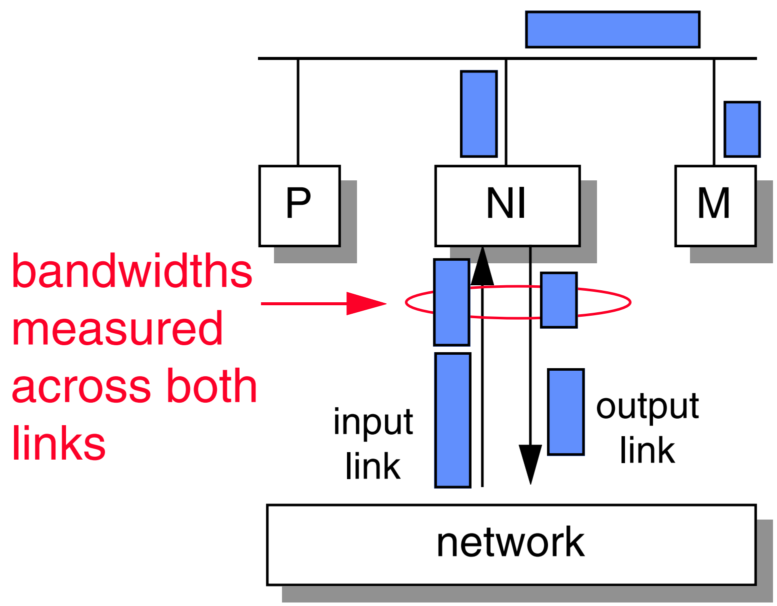
$$y_2 = K_2 x_2 = \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & - & 1 & 1 \\ - & - & - & - \\ 1 & - & 1 & 1 \\ 1 & - & 1 & 1 \end{bmatrix} \end{array} \begin{bmatrix} 2 \\ - \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ - \\ 6 \\ 6 \end{bmatrix}$$

Parallel SMVP: Communication phase



System model

data moves between PEs
in blocks (e.g., msgs, pages,cache lines)



Network latency assumed to be zero

System parameters

T_f : average time per flop during compute phase.
inverse of sustained MFLOPS.
T3E: 14 ns (70 MFLOPS)

T_l : block transfer latency.
time on PE to send/receive first word of a block.
T3E: 22 us

T_w : average time per additional block word.
inverse of burst bandwidth.
T3E: 55 ns (145 MB/s)

Application parameters

F : flops/PE

C_{max} : max comm words per PE

B_{max} : max blocks(msgs) per PE

C_{max}/B_{max} : avg msg size

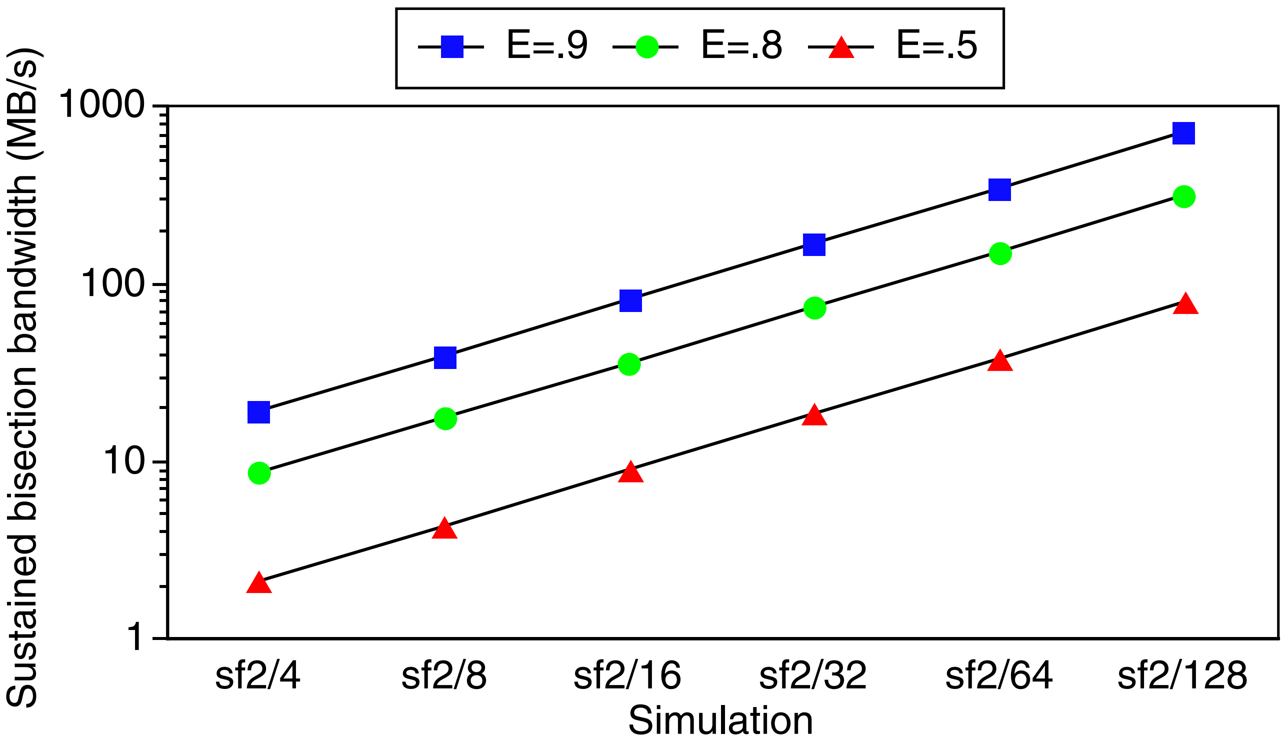
sf2/PEs	4	8	16	32	64	128
F	24,640,110	12,414,006	6,278,076	3,191,436	1,632,708	838,224
C_{max}	55,338	35,148	28,482	24,018	20,520	16,260
B_{max}	6	10	16	26	36	50
C_{max}/B_{max}	9223	3515	1780	923	570	325

Properties of unstructured finite element Quake applications

- big problems (e.g., 378K nodes, 2M element, 450 MBytes), but ...
- **lots of small messages.**
- **message overheads cannot be amortized by long messages.**

- almost identical to Cypher's irregular EXFLOW application [ISCA93].
  - EXFLOW and Quake both transfer more messages with an average smaller size than most regular applications [ISCA93].

Sustained bisection bandwidth required for sf2 SMVP



T_f = 5 ns (200 MFLOPS)

Efficiency (E) is the fraction of time spent in the computation phase.

Conclusion #1

Bisection bandwidth:
Not an issue for future systems running unstructured finite element simulations.

(700 MB/s)

(current link bandwidth : 300 MB/s)

High-level SMVP performance model

Relates sustained PE computation performance to sustained PE communication performance

Average time per communication word (inverse of sustained PE bandwidth)

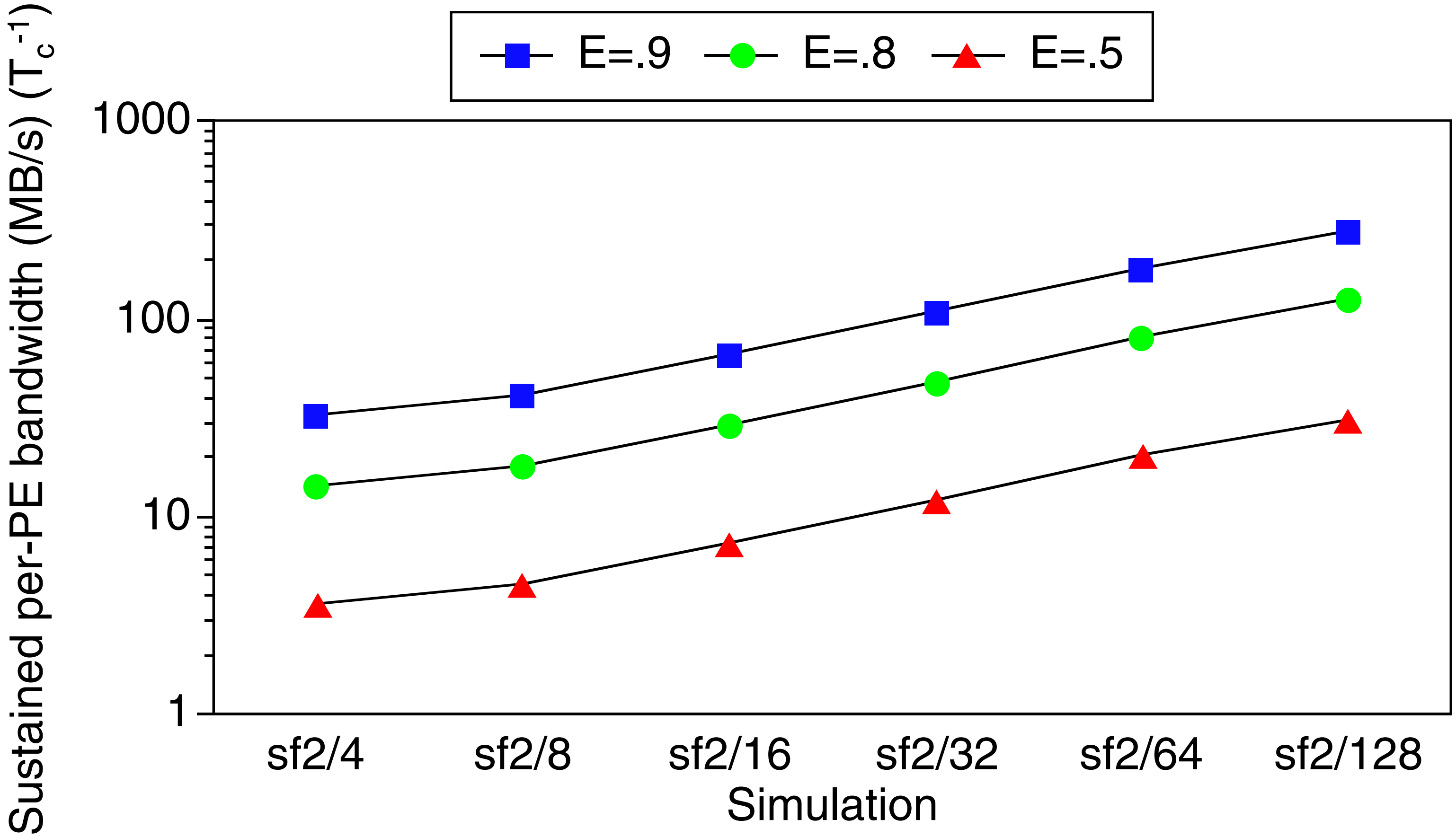
$$T_c = \frac{F}{C_{\max}} \frac{1-E}{E} T_f$$

Average time per flop (inverse of sustained PE MFLOPS)

Computation/ communication ratio

Efficiency (E) is the fraction of time spent in the computation phase

Sustained per-PE bandwidth required for the sf2 SMVP



T_f = 5 ns (200 MFLOPS)

Conclusion #2

Sustained per-PE bandwidths:
Aggressive but doable

(300 MB/s sustained)

(Current systems 20-40 MBytes/s)

Low-level performance model

Relates burst bandwidth and block latency to sustained PE communication performance

Average time per communication word (inverse of sustained PE bandwidth)

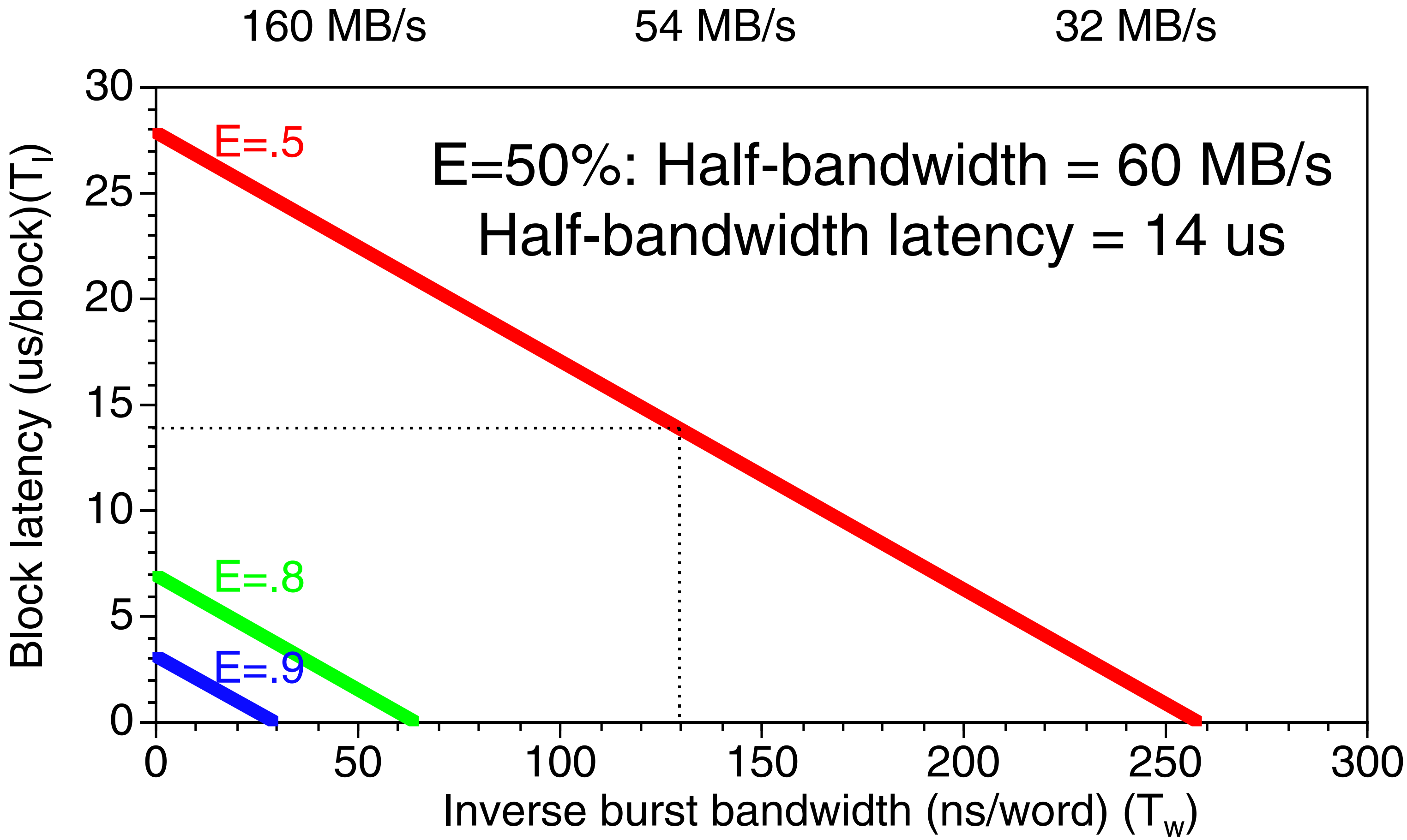
$$T_c <= \frac{B_{max}}{C_{max}} T_l + T_w$$

inverse of burst bandwidth

Inverse of average message size

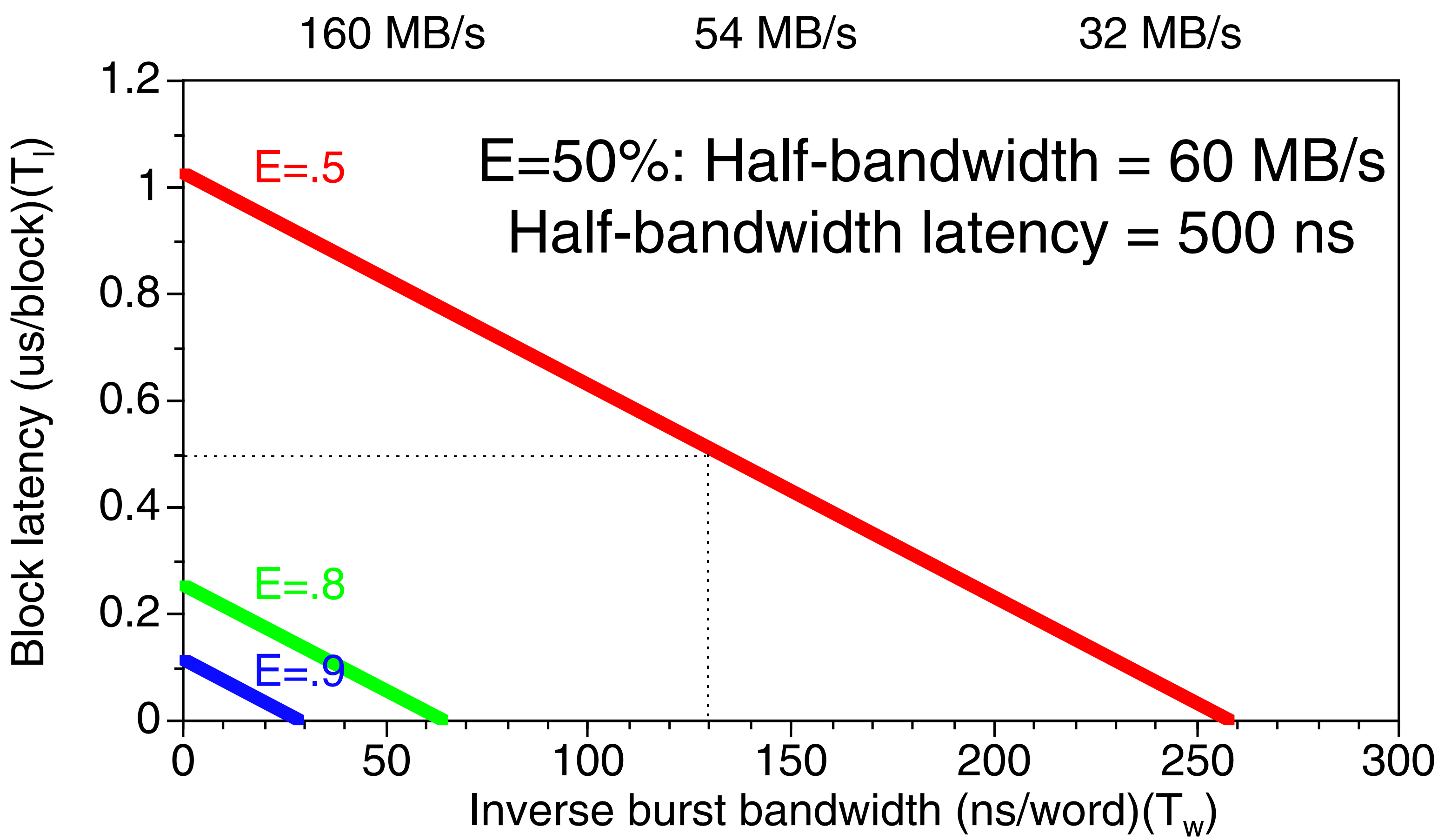
block latency

Burst bandwidth and latency tradeoffs for sf2/128



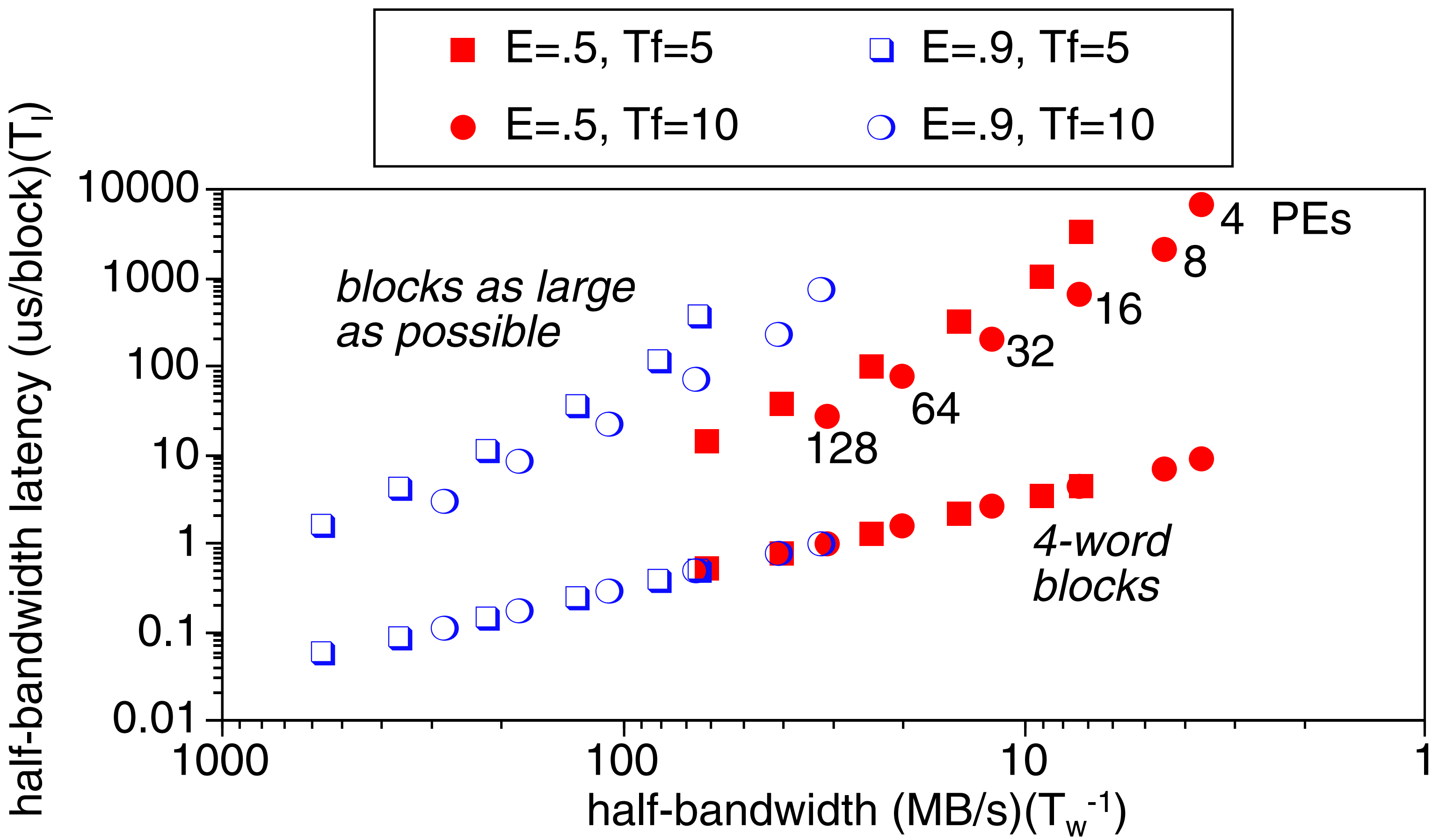
maximally aggregated blocks, Tf = 5ns (200 MFLOPS)

Burst bandwidth and latency tradeoffs for the sf2/128 SMVP



4-word block sizes, Tf = 5ns (200 MFLOPS)

Half-bandwidth and latencies for the sf2 SMVP



Conclusion #3

Block latency:
severe requirements; require aggressive latency reduction.

(2 us for large blocks, 70 ns for small blocks)

(current systems 22 us)

Conclusions

For unstructured finite element simulations based on SMVP:

- Bisection bandwidth: not an issue.
- Sustained PE bandwidth: aggressive but doable.
- Block latency: challenging.

Future

- **Similar characterizations needed for other important irregular applications:**
  - sparse preconditioners
  - sparse direct solvers
  - tree codes
- **Characterizations must be based on real applications.**