

SAND REPORT

SAND2003-2814

Unlimited Release

Printed August, 2003

Radiation Transport Algorithms on Trans-Petaflops Supercomputers of Different Architectures

Thomas W. Christopher
Scalable Computing Systems

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.doe.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/ordering.htm>



SAND2003-2814
Unlimited Release
Printed August 2003

Radiation Transport Algorithms on Trans-Petaflops Supercomputers of Different Architectures

Thomas W. Christopher
Scalable Computing Systems
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-1110

Abstract

We seek to understand which supercomputer architecture will be best for supercomputers at the Petaflops scale and beyond. The process we use is to predict the cost and performance of several leading architectures at various years in the future. The basis for predicting the future is an expanded version of Moore's Law called the International Technology Roadmap for Semiconductors (ITRS). We abstract leading supercomputer architectures into chips connected by wires, where the chips and wires have electrical parameters predicted by the ITRS. We then compute the cost of a supercomputer system and the run time on a key problem of interest to the DOE (radiation transport). These calculations are parameterized by the time into the future and the technology expected to be available at that point.

We find the new advanced architectures have substantial performance advantages but conventional designs are likely to be less expensive (due to economies of scale). We do not find a universal "winner," but instead the right architectural choice is likely to involve non-technical factors such as the availability of capital and how long people are willing to wait for results.

Intentionally Left Blank

Contents

| | |
|---------------------------------------|----|
| Introduction | 7 |
| Hardware | 8 |
| Architecture..... | 9 |
| The Radiation Transport Problem | 10 |
| Proposed Solution Method..... | 11 |
| Results | 15 |
| Conclusions | 17 |
| References | 19 |
| Distribution..... | 20 |

Figures

| | |
|---|----|
| Figure 1 PIM, no DRAM..... | 9 |
| Figure 2 RS Red Storm-like architecture..... | 10 |
| Figure 3 PIM+DRAM..... | 10 |
| Figure 4 Processors/chip..... | 14 |
| Figure 5 Chip cost of minimal systems to solve the problem..... | 15 |
| Figure 6 Sweep times..... | 15 |
| Figure 7 Cost per sweep..... | 16 |
| Figure 8 Cost of system with various PIM chip prices..... | 17 |
| Figure 9 Cost per sweep with various PIM chip prices..... | 17 |

Tables

| | |
|---|----|
| Table 1. Hardware (ITRS) Parameters..... | 8 |
| Table 2. Relative performance for a single sweep..... | 13 |
| Table 3. Assumed Chip Prices..... | 14 |

Nomenclature

| | |
|------------|---|
| ASCI..... | Advanced Simulation and Computing (the I is silent) |
| DIMM..... | Dual In Line Memory Module |
| DRAM..... | Dynamic Random Access Memory |
| FLOPS..... | Floating Operations Per Second |
| IPS..... | Instructions Per Second |
| ITRS..... | International Technology Roadmap for Semiconductors |
| MPP..... | Massively Parallel Processor |
| MPU..... | Microprocessor Unit |
| PIM..... | Processor-In-Memory |
| RISC..... | Reduced Instruction Set Computer |
| RS..... | Red Storm |
| SNL..... | Sandia National Laboratories |

Introduction

We seek to increase the throughput of supercomputers from today's 10 Teraflops to 1 Petaflops or more – an increase of 100x or more. To understand the challenges, let's begin with Little's Law from queuing theory:

$$\text{throughput} = \frac{\text{concurrency}}{\text{latency}}$$

In this expression, *concurrency* is the number of activities that take place at once, *latency* is the time per activity, and *throughput* is the number of activities that are completed per unit time.

To increase throughput by 100x or more would require increasing concurrency and decreasing latency by a combined factor of 100 or more.

In past generations, supercomputer throughput increased largely by speeding the clock rate of the microprocessors. By conventional reasoning, an increase in clock rate speeds the time for every part of a calculation and gives the effect of decreasing latency.

Unfortunately, this trick won't work again. Microprocessor clock rates have sped up so much that the performance of the memory subsystem has become the rate-limiting attribute.

This leaves two solutions, both of which are explored in this paper:

1. Reducing latency further by an architectural change to the memory subsystem.
2. Increasing concurrency by increasing the number of processors.

However, the "economy of scale" principle runs in opposition to any innovative solution. To be specific, innovative designs that require custom chips incur substantially higher costs.

To judge what approach is best for future generation supercomputers, we need to estimate the overall effectiveness of ASCI applications on conventional plus alternative architectures.

We will explore a benchmark radiation transport problem with a 1000^3 array of cells, two particle types, 1000 energy levels, and 5000 angles^[ref. 1] and ask how future PIM-based and Red Storm-like MPPs would perform for this problem.

We answer the question using the following methods:

1. We use the ITRS tables to project the performance of chips over the years.
2. We compare three computer designs: Red Storm-like, pure PIM, and PIM+DRAM.
3. We estimate the cost of the hardware, the time to perform one sweep, and the cost of that sweep (the cost of the hardware per one second of the machine life-time times the number of seconds in the sweep).

Hardware

The relevant ITRS information is summarized in Table 1. R is the serial data rate available per signal pad (pin), N_{pads} is the number of signal pads available on the chip. $B_{chip} = R \cdot N_{pads}/2$ is overall bandwidth per chip. The division by 2 converts the number of pads into the number of differential pairs. The memory capacity per chip, Gb/chip, is computed from DRAM Gb/cm² and the chip size. Cells/chip is based on the bits required for the radiation transport problem (discussed later). The chip's clock rate allows us to compute how many instructions we can expect to execute per second at one per clock for each PIM RISC core and two per clock for superscalar MPUs.

Table 1. Hardware (ITRS) Parameters

| Year | R, Gb/sec. | N_{pads} , Number of signal pads | B_{chip} GB/s | Gb/cm ² | Chip size cm ² | Gb/chip | Cells/chip | C_{ps} , Clock MHz |
|--|---------------------------------------|--|--------------------|----------------------------------|-------------------------------------|---------|------------|----------------------------------|
| 2001 | 2.5 | 1500 | 1875.0 | 0.55 | 3.1 | 1.705 | 1.33 | 1684.0 |
| 2002 | 3.13 | 1600 | 2504.0 | 0.70 | 3.1 | 2.170 | 2.17 | 2317.0 |
| 2003 | 3.13 | 1700 | 2660.5 | 1.18 | 3.1 | 3.658 | 2.86 | 3088.0 |
| 2004 | 10.0 | 1800 | 9000.0 | 1.49 | 3.1 | 4.619 | 3.61 | 3990.0 |
| 2005 | 10.0 | 2000 | 10000.0 | 1.89 | 3.1 | 5.859 | 4.58 | 5173.0 |
| 2006 | 40.0 | 2100 | 42000.0 | 2.39 | 3.1 | 7.409 | 5.79 | 5631.0 |
| 2007 | 40.0 | 2200 | 44000.0 | 3.03 | 3.1 | 9.393 | 7.34 | 6739.0 |
| 2010 | 40.0 | 2400 | 48000.0 | 6.10 | 3.1 | 18.910 | 14.77 | 11511.0 |
| 2013 | 40.0 | 2700 | 54000.0 | 18.42 | 3.1 | 57.102 | 44.61 | 19348.0 |
| 2016 | 40.0 | 3000 | 60000.0 | 37.00 | 3.1 | 114.700 | 89.61 | 28751.0 |
| ITRS ^{ITRef.} 2) Source | Tables 23a & 23b 2002 update | Tables 3a & 3b 2002 update | | Tables 1e & 1f 2002 update | Tables 1i & 1j 2002 update | | | Tables 4c & 4d 2002 update |

Architecture

We are taking an abstract approach to computer architecture with the objective of determining an upper bound on the possible performance of various architectures. More specifically, we define each architecture by its principal chips and interconnections between them. We assume a future engineer will fill the principal chips with logic and memory such that performance on this algorithm will be optimized. This is a tall order particularly for the PIM architecture, as its proponents often have well-developed ideas about PIM internal logic – with different people having different and non-overlapping ideas.

We are assuming that each design is connected in a 3-D mesh. The PIM node design is shown in Figure 1. Each node consists of a single PIM chip and each connection uses one-sixth of the signal pads.

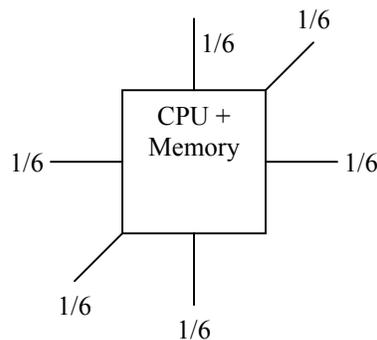


Figure 1. PIM, no DRAM.

The RS (Red Storm-like) system's node is shown in Figure 2. The router chip is connected to six neighbors and a single MPU chip. Each connection is assigned $1/7$ of the router's pads. The MPU assigns $1/7$ of its pads to the connection with the router and the other $6/7$ to the memory bus. The memories, assumed to have the same number of pads as the router and MPU, leave $1/7$ of their signal pads unused.

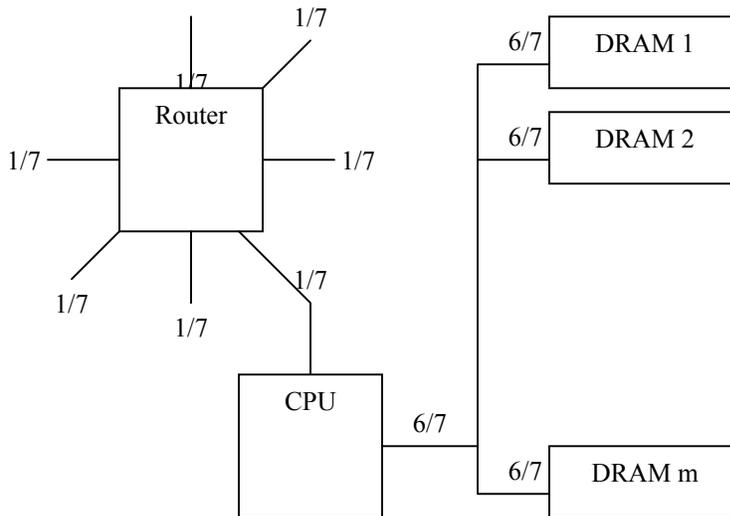


Figure 2. RS Red Storm-like architecture.

The PIM+DRAM design is shown in Figure 3. The PIM is connected to six neighbors and to a memory bus. Here we are assuming that 1/7 of the pads are devoted to each use, so its communication rate is the same as the Red Storm design, but the memory bandwidth is much lower.

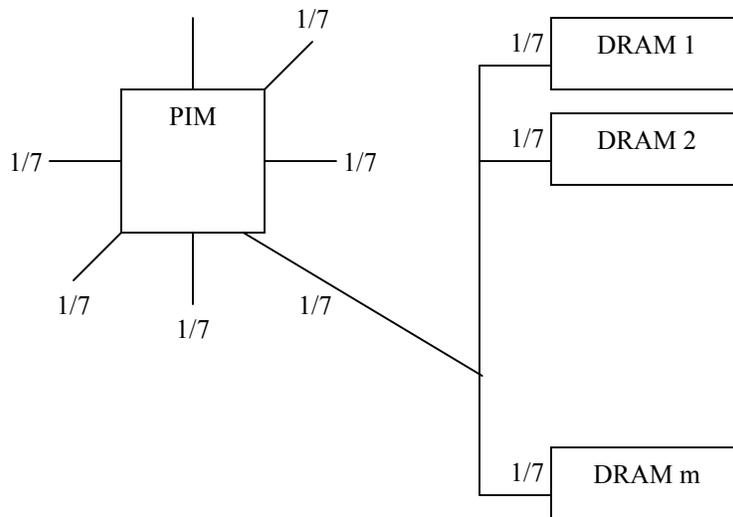


Figure 3. PIM+DRAM.

The Radiation Transport Problem

A radiation transport problem with a 1000^3 array of cells, two particle types, 1000 energy levels, and 5000 angles is of a ferocious size, even if we begin by

assuming the 5000 angles are the total number meaning there are 625 per octant. (Otherwise, we can multiply by eight later.) Let

- $N_a = 5000/8 = 625$ be the number of angles per octant,
- $N_s = 2$ be the number of particle species,
- $N_e = 1000$ be the number of energy levels,
- $D = 1000$ be the dimension of the problem (the number of cells in D^3),
- S_{fit} be the size of a float in bytes.

Thus, $S_{cell} = 2 \cdot 8 \cdot N_a \cdot N_s \cdot N_e \cdot S_{fit} = 160$ MB is the size of a cell. The factor of 2 allows two floats for old and new values. The 8 converts the angles per octant back into total angles. The overall space requirements for the 10^9 cells is 1.6×10^{17} bytes. With 1GB memory DIMMs at \$5 apiece, the machine would cost \$800 million for memory chips alone.

We will restrict ourselves to PIM systems that can contain at least one cell entirely within the on-chip memory. If a cell will not fit, the PIM will begin to resemble an MPU with caching. By this rule, the problem will not run on the Blue Gene/ Cyclops (BG/C) currently being developed by IBM: the cells are more than 26 times too large to fit in a BG/C PIM.

We will also abandon here the idea that we can run the standard code SWEEP3D^[ref 3.] with its two dimensional partition of space. In the reference SWEEP3D model, the memory per node has to contain some number of columns. Each column contains D cells of 160MB each, or 1.6×10^{11} bytes. With one column per node, that is 160 1GB DIMMS per node and 1,000,000 nodes in the machine. With future semiconductor technology, we can reduce the number of DIMMs per node, but still 1.6×10^{11} bytes of DRAM per node is steep, and it leaves the nodes memory-heavy and processor-starved.

It does not appear viable to solve the problem with the current generation of semiconductors. What are the prospects for running the problem on a future system?

Proposed Solution Method

We will follow a simple estimation technique and consider only the data size and data movement required, and number of instructions to be executed. We will do the following:

- Redesign the algorithm to use 3-D partitioning, rather than SWEEP3D's 2-D.
- Calculate the number of cells that can be stored at a node, not bothering to convert to integers, but allowing fractional cells per node in the calculations.

- Assume one or more perfectly cubic blocks of cells are allocated to each node.
- Ignore the idle time between sweeps and just calculate the time required for a node to participate in a single sweep: the sweeps are long enough that we are within 1/1000 of the correct figure.
- Calculate the FLOPS or IPS (instructions per second) rate provided and the time required to execute the sweep.
- Calculate the time to execute one sweep.
- Estimate the cost of the chips in a machine and, the cost per second of machine time assuming a three-year lifetime.
- Estimate the cost of a sweep.

For all the machines we need to consider the cost of passing sweeps of data in and out of the nodes and the cost of processing each value in the sweep. For those designs with external DRAM, we need to calculate the cost of loading and storing cells from the DRAM.

Let

- $s_{flt} = 8 \cdot S_{flt}$ be the size of a float in bits. (Substitute some other number of bits per byte if you wish. They would need parity.)
- $s_{cell} = 2 \cdot 8 \cdot N_a \cdot N_s \cdot N_e \cdot s_{flt} = 1.28 \times 10^9$ be the size of a cell in bits. The 2 allows two floats for old and new values. The 8 converts the angles per octant to total angles.
- $s_{msg} = s_{flt}$ be the size in bits of a value sent from one cell to another in a sweep.

Recall, $N_{sweep} = N_a \cdot N_s \cdot N_e = 1,250,000$ is the number of floats sent from one cell to a single neighbor in a sweep.

$T_{sweep} = N_{sweep} \cdot s_{msg} / (R \cdot N_{pads}/2) = 1.667 \times 10^{-7}$ is the time required to move the entire number of bits in a sweep from one cell to another cell off-chip using all the pads. It needs to be divided by the fraction of the pads being used, but since that will be different in the different systems, we will leave it out of this formula. Since the nodes will contain blocks of cells, T_{sweep} must also be multiplied by the number of cells exposed along the side of the block and by the number of sides sharing a single link (six in the case of router-MPU link on the RS, one for a neighbor link in the PIMs).

Formulae for the execution time and chips required are shown in Table 2. They compare the amount of time required to send and/or receive data from the neighbors, the amount of time to swap cells from and to DRAM memory, the instruction execution time, and the number of chips required.

Table 2. Relative performance for a single sweep

| | PURE PIM | RS | PIM+DRAM |
|--|--|---|---|
| Communications Time for an Iteration | $6 \cdot C_c^{2/3} \cdot T_{sweep}$ | $m \cdot 6 \cdot 7 \cdot C_c^{2/3} \cdot T_{sweep}$ | $m \cdot 7 \cdot C_c^{2/3} \cdot T_{sweep}$ |
| Swap Time (Moving All Cells To and From the Processor) | 0 | $\frac{2 \cdot C_c \cdot m \cdot s_{cell} \cdot 7/6}{R \cdot N_{pads}/2}$ | $\frac{2 \cdot C_c \cdot m \cdot s_{cell} \cdot 7}{R \cdot N_{pads}/2}$ |
| Instruction execution time per sweep | $\frac{N_{sweep} \cdot U}{P_{PIM} \cdot C_{ps}}$ | $\frac{N_{sweep} \cdot U}{I_{pc} \cdot P_{MPU} \cdot C_{ps}}$ | $\frac{N_{sweep} \cdot U}{P_{PIM} \cdot C_{ps}}$ |
| PIM Chips Required | $\frac{D^3 \cdot s_{cell}}{s_{chip}}$ | | $\frac{(D^3 \cdot s_{cell})}{s_{chip}} \cdot \frac{1}{m}$ |
| Memory Chips Required | | $\frac{D^3 \cdot s_{cell}}{s_{chip}}$ | $\frac{D^3 \cdot s_{cell}}{s_{chip}}$ |
| MPU Chips Required | | $\frac{(D^3 \cdot s_{cell})}{s_{chip}} \cdot \frac{1}{m}$ | |
| Router Chips Required | | $\frac{(D^3 \cdot s_{cell})}{s_{chip}} \cdot \frac{1}{m}$ | |

For the pure PIM machine, each node can contain only as many cells, C_c , as can fit on a single chip. $C_c^{2/3}$ is the number of cells exposed along one side of the cube to communicate with off-PIM neighbors. The communication time is $6 \cdot C_c^{2/3} \cdot T_{sweep}$, since the PIM can communicate with all neighbors simultaneously and 1/6 of the pads are used for each.

One would imagine that RS and PIM+DRAM machines would contain single blocks of size $m \cdot C_c$. Unfortunately, that would require all the cells from the m memory chips to be loaded and stored for each set of messages received from neighboring node. We need to use a form of striping, where the overall space of cells is partitioned into m blocks each of which is partitioned among the nodes. With $m = 64$ DRAMs per node, we would partition the 1000^3 cells into 4^3 blocks of 250^3 cells. These blocks are partitioned among the nodes, giving C_c to each. The blocks are processed one at a time in overall sweep order performing their parts of the sweep. Assuming C_c cells fit in cache, the RS and PIM+DRAM can load and store each cell only once per sweep, albeit much faster in the RS with its higher memory bandwidth.

The communication time $m \cdot 6 \cdot 7 \cdot C_c^{2/3} \cdot T_{sweep}$ for RS includes 7, to account for the speed of the router links and 6 to count the number of neighbors sharing the single router/MPU link.

The factor U in the formulae for instruction execution time indicates the number of instructions on the average executed updating a cell for each particle/energy/angle element of the sweep streams. P_{PIM} is the number of RISC processors per PIM. Figure 4 shows the values we assume for P_{PIM} over the range of years. It is based on 2.5% of the chip space being devoted to processor cores and 3,000,000 transistors per processor. I_{pc} is the number of instructions executed per cycle by a superscalar (MPU) processor (we assume it is two), and C_{ps} is the number of cycles executed per second.

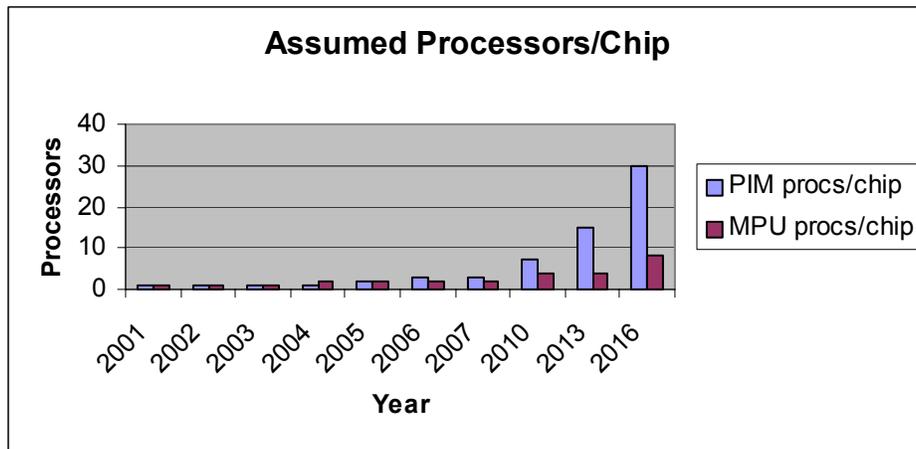


Figure 4. Processors/chip.

We let m , the number of DRAM chips per node, be 64 for both the RS and the PIM+DRAM systems. We are assuming the chip prices are those given in Table 3*. These will allow us to compute the price of the minimal system necessary to solve the problem. We assume the lifetime of a machine is three years. The price per second of the machine times the number of seconds required to perform one sweep gives us the cost of the sweep.

Table 3. Assumed Chip Prices

| | |
|------------------|----------|
| price per PIM | \$300.00 |
| price per DRAM | \$5.00 |
| price per MPU | \$150.00 |
| price per router | \$300.00 |

* The figures shown are based on the authors' experience. However, actual prices depend on many factors beyond the scope of this document, such as chip size and negotiating position. A reader disagreeing with these figures is encouraged to get the spreadsheet that was used to compute subsequent tables and change the prices.

Results

The costs of the minimal systems to solve are shown in Figure 5, and the sweep times are shown in Figure 6. These give us the costs per sweep shown in Figure 7.

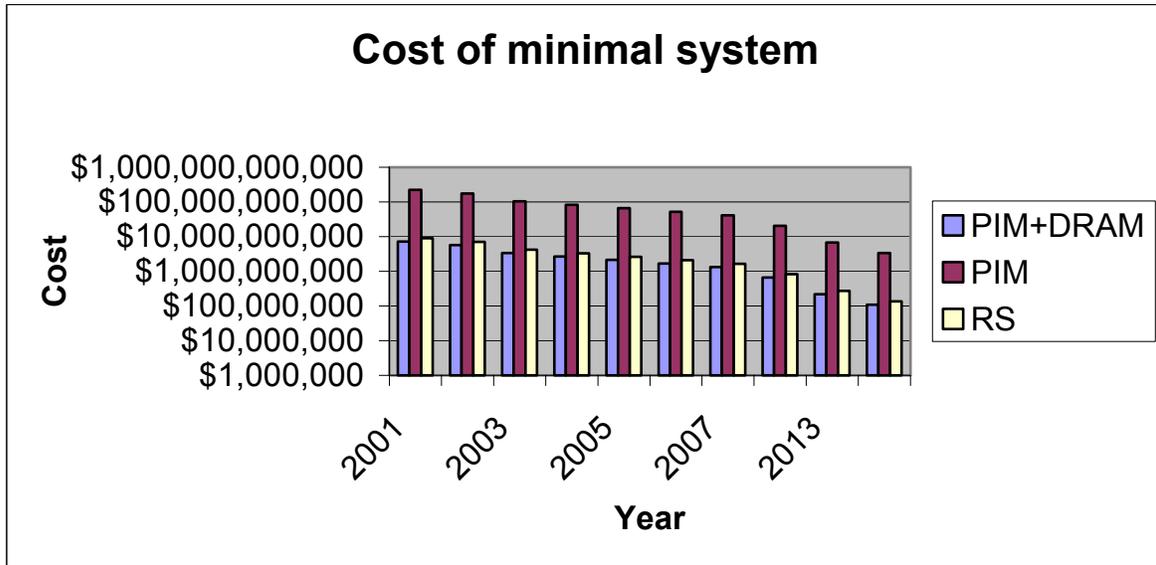


Figure 5. Chip cost of minimal systems to solve the problem.

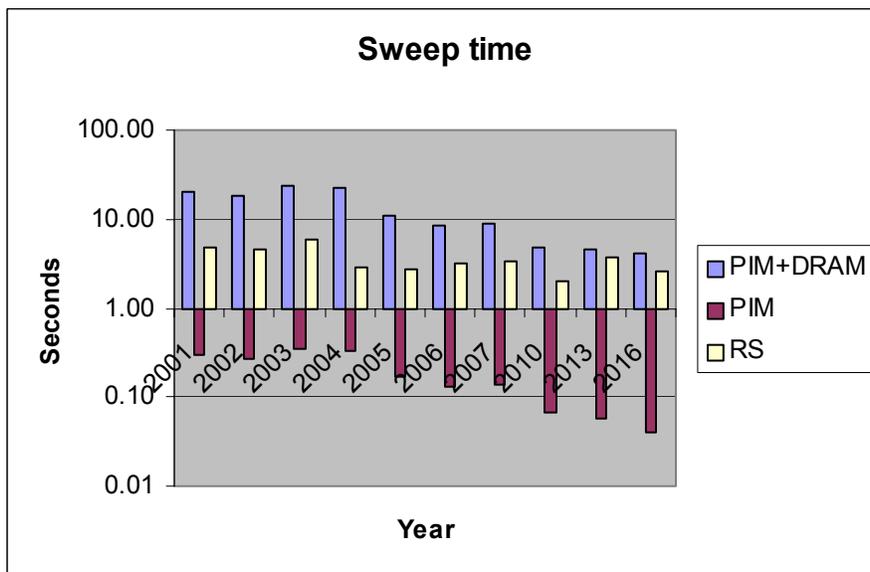


Figure 6. Sweep times.

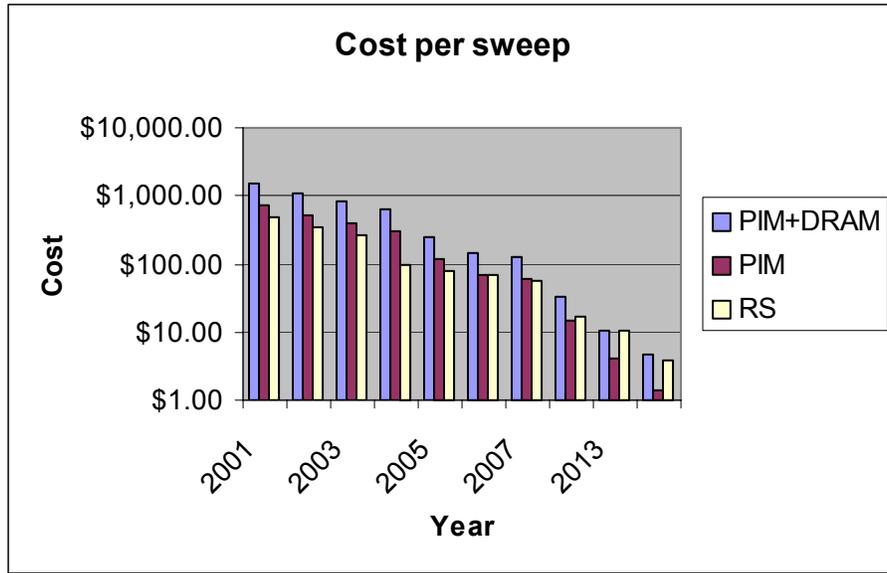


Figure 7. Cost per sweep.

The cost of a minimal PIM-only system to solve the radiation-transport problem is more than an order of magnitude larger than a PIM+DRAM or Red Storm-like system, but this is only to be expected from the differences in memory prices. The number of PIMs in a PIM system is equal to the number of DRAMs in a RS system. With 64 DRAMs per MPU and router, the overall chip cost per DRAM on a RS system is $5 + ((150 + 300) / 64) = 12.03$ dollars, so the PIM system price is about 25 times that of the RS. The sweep time on the PIM system is an order of magnitude lower than the RS, which makes the cost per sweep about the same, although the 2013 and 2016 years currently appear to be a win for PIMs.

Figure 8 and Figure 9 illustrate the fact that the equivalence of RS and PIM systems is very much a result of our assumptions about the costs of chips. If PIMs were to become a commodity, the price would decline. At \$30 per PIM, the order of magnitude decline in cost produces an order of magnitude improvement in cost per sweep. At \$10 per PIM, the system prices become nearly equivalent.

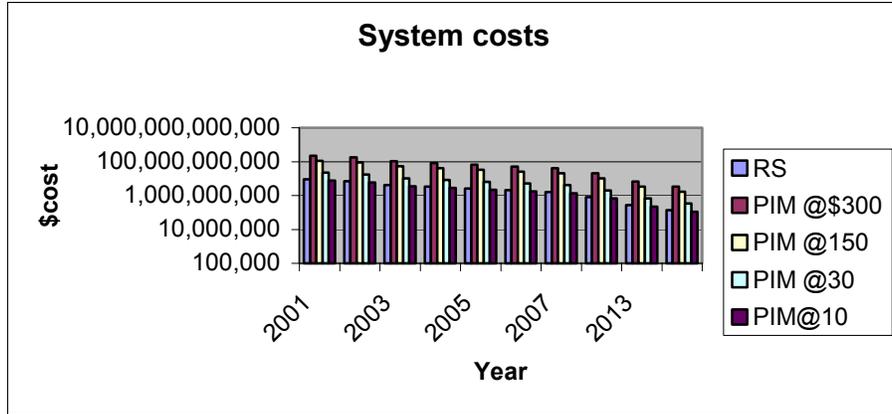


Figure 8. Cost of system with various PIM chip prices.

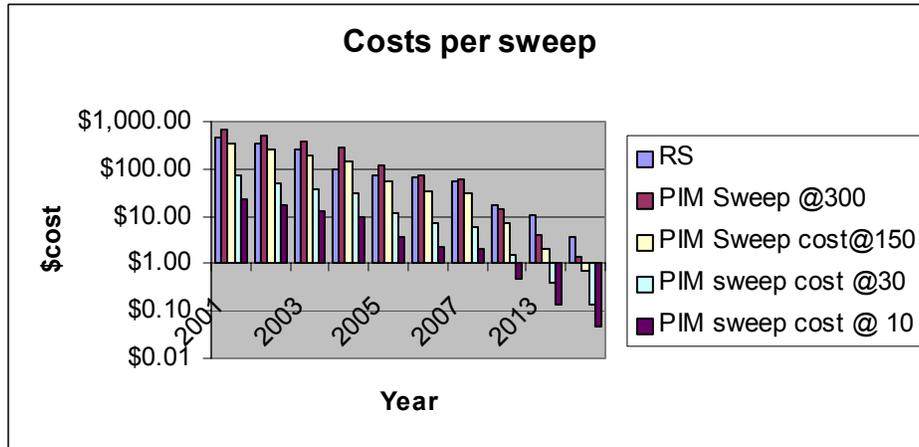


Figure 9. Cost per sweep with various PIM chip prices.

Conclusions

We have predicted system costs, speed, and cost per sweep of PIM, Red Storm-like, and PIM+DRAM systems over the next decade when applied to a large radiation transport problem. One argument to dismiss PIM-based systems out of hand is that their internal memories are too small and their bandwidth to external DRAM is too low. Arguing that the small internal PIM memory would force recoding the SWEEP3D family of algorithms is true, but the large problem size would force an equivalent recoding for RS-like systems. The sweep time and cost per sweep for a PIM+DRAM system is a bit worse than for the RS, which supports the argument that low bandwidth to off-chip DRAM will be the bane of PIM+DRAM systems.

A problem of the size studied here will certainly stress any hardware procurement budget in the near term. Assuming PIM chips cost about the same as routers, PIM systems can be expected to cost an order of magnitude more than Red Storm-like systems, but since they are an order of magnitude faster, the cost per sweep will be about the same (ignoring the cost of waiting for the answer). Significant declines in PIM prices would bring a PIM system's cost closer to a RS-like system's and give the PIM systems a significantly lower cost per sweep.

References

1. Bill Camp, personal communications.
2. International Technology Roadmap for Semiconductors, <http://public.itrs.net>.
3. Adolfo Hoisie, Olaf Lubeck, Harvey Wasserman, "*Performance and Scalability Analysis of Teraflop-Scale Parallel Architectures Using Multidimensional Wavefront Applications*," The International Journal of High Performance Computing Applications, Sage Science Press, Volume 14, Number 4, Winter 2000.

Distribution:

| | | | | | |
|------|------|--------------------------|------|------|------------------------------------|
| 1 MS | 9037 | J. C. Berry, 8945 | 1 MS | 0818 | P. Yarrington, 9230 |
| 1 | 9019 | S. C. Carpenter, 8945 | 1 | 0819 | R. M. Summers, 9231 |
| 1 | 9012 | J. A. Friesen, 8963 | 1 | 0820 | P. F. Chavez, 9232 |
| 1 | 9012 | S. C. Gray, 8949 | 1 | 0316 | S. S. Dosanjh, 9233 |
| 1 | 9011 | B. V. Hess, 8941 | 1 | 0316 | J. B. Aidun, 9235 |
| 1 | 9915 | M. L. Koszykowski, 8961 | 1 | 0813 | R. M. Cahoon, 9311 |
| 1 | 9019 | B. A. Maxwell, 8945 | 1 | 0801 | F. W. Mason, 9320 |
| 1 | 9012 | P. E. Nielan, 8964 | 1 | 0806 | C. Jones, 9322 |
| 1 | 9217 | S. W. Thomas, 8962 | 1 | 0822 | C. Pavlakos, 9326 |
| 1 | 0824 | A. C. Ratzel, 9110 | 1 | 0807 | J. P. Noe, 9328 |
| 1 | 0847 | H. S. Morgan, 9120 | 1 | 0805 | W.D. Swartz, 9329 |
| 1 | 0824 | J. L. Moya, 9130 | 1 | 0812 | M. R. Sjulín, 9330 |
| 1 | 0835 | J. M. McGlaun, 9140 | 1 | 0813 | A. Maese, 9333 |
| 1 | 0833 | B. J. Hunter, 9103 | 1 | 0812 | M. J. Benson, 9334 |
| 1 | 0834 | M. R. Prarie, 9112 | 1 | 0809 | G. E. Connor, 9335 |
| 1 | 0555 | M. S. Garrett, 9122 | 1 | 0806 | L. Stans, 9336 |
| 1 | 0821 | L. A. Gritzko, 9132 | 1 | 1110 | R. B. Brightwell, 9224 |
| 1 | 0835 | E. A. Boucheron, 9141 | 1 | 1110 | R. E. Riesen, 9223 |
| 1 | 0826 | S. N. Kempka, 9113 | 1 | 1110 | K. D. Underwood, 9223 |
| 1 | 0893 | J. Pott, 9123 | 1 | 1110 | E. P. DeBenedictis, 9223 |
| 1 | 1183 | M. W. Pilch, 9133 | 1 | 0321 | W. Camp, 9200 |
| 1 | 0835 | K. F. Alvin, 9142 | 1 | 0841 | T. Bickel, 9100 |
| 1 | 0834 | J. E. Johannes, 9114 | 1 | 9003 | K. Washington, 8900 |
| 1 | 0847 | J. M. Redmond, 9124 | 1 | 0801 | A. Hale, 9300 |
| 1 | 1135 | S. R. Heffelfinger, 9134 | 1 | 0139 | M. Vahle, 9900 |
| 1 | 0826 | J. D. Zepper, 9143 | 1 | 0134 | Ron Detry, 9700 |
| 1 | 0825 | B. Hassan, 9115 | | | |
| 1 | 0557 | T. J. Baca, 9125 | 1 | 9018 | Central Technical Files, 8945-1 |
| 1 | 0836 | E. S. Hertel, Jr., 9116 | | | |
| 1 | 0847 | R. A. May, 9126 | | | |
| 1 | 0836 | R. O. Griffith, 9117 | 2 | 0899 | Technical Library, 9616 |
| 1 | 0847 | J. Jung, 9127 | | | |
| 1 | 0321 | P. R. Graham, 9208 | | | |
| 1 | 0318 | J. E. Nelson, 9209 | | | |
| 1 | 0847 | S. A. Mitchell, 9211 | | | |
| 1 | 0310 | M. D. Rintoul, 9212 | | | |
| 1 | 1110 | D. E. Womble, 9214 | | | |
| 1 | 1111 | B. A. Hendrickson, 9215 | | | |
| 1 | 0310 | R. W. Leland, 9220 | | | |
| 1 | 1110 | N. D. Pundit, 9223 | | | |
| 1 | 1110 | D. W. Doerfler, 9224 | | | |
| 1 | 0847 | T. D. Blacker, 9226 | | | |
| 1 | 0822 | P. Heermann, 9227 | | | |