Evaluation and Modeling of Program Execution Models

Stéphane Zuckerman

Computer Architecture & Parallel Systems Laboratory
Department of Electrical and Computer Engineering
University of Delaware

October 19th, 2011

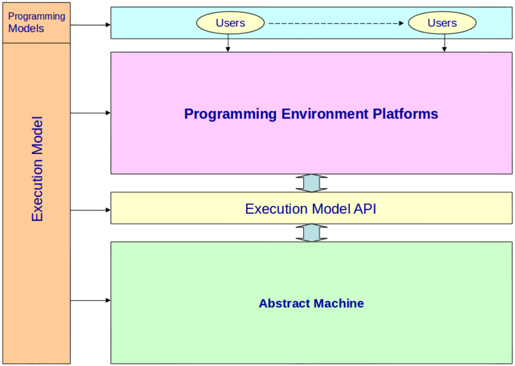
Outline

Looking at a PXM Abstract Machine again

Where to Implement a PXM?
- Full Hardware Implementation
- Full Software Implementation
- Hardware-Software Co-Design
- Timeline

Evaluating PXMs' Efficiency [4, 9]
- Analytical Models
- Micro-Benchmarking
- Application Benchmarking
- Evaluating Extensions to a given couple PXM-Abstract Machine

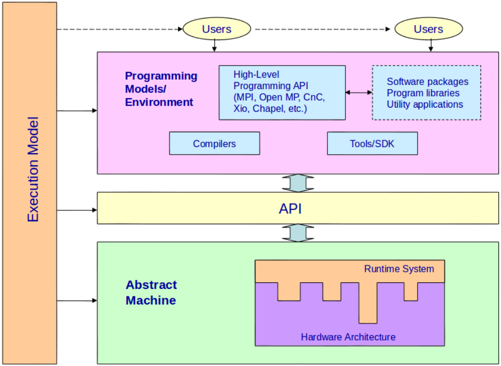
Relationship Between PXMs and Actual Computer Systems



Execution Model and Abstract Machines

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Relationship Between PXMs and Actual Computer Systems



Execution Model and Abstract Machines

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Outline

Looking at a PXM Abstract Machine again

Where to Implement a PXM?
- Full Hardware Implementation
- Full Software Implementation
- Hardware-Software Co-Design
- Timeline

Evaluating PXMs' Efficiency [4, 9]
- Analytical Models
- Micro-Benchmarking
- Application Benchmarking
- Evaluating Extensions to a given couple PXM-Abstract Machine

Full Hardware Implementation

Pros

- Would seem like the most efficient method: No additional software layer between the programmer and the hardware
- HW and abstract machines are a 1:1 match

Cons

- Any mistake in hardware is *costly*
  - Bug in the implementation
  - Conceptual mistake in the design
- Needs a “perfect” design beforehand
- Not always possible financially
- Makes the implementation of other PXMs potentially more difficult (not necessarily a weakness)

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Full Software Implementation

Pros

- Very flexible: any hardware architecture can be targeted
- Any oversight in the design of the PXM can be fixed relatively easily

Cons

- Some operations can be very slow if not implemented in hardware
- Can force the high-level programmers to know more about ”gory details” than they should in order to make programs run efficiently

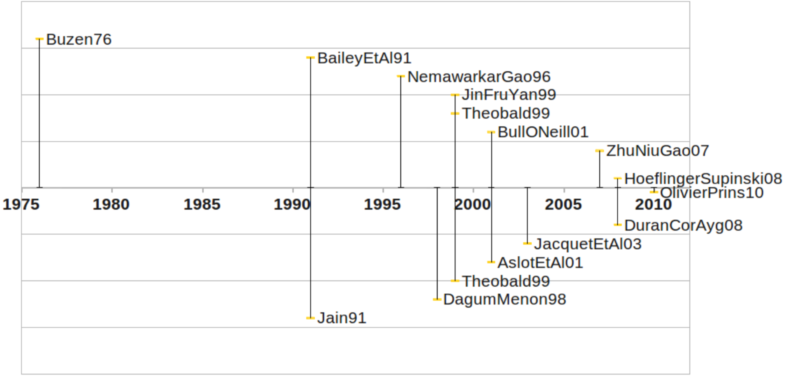
PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Hardware-Software Co-Design

- Trade-offs must be found (eg: atomic instructions to help build fast lock operations)
- Needs ways to model, measure and evaluate how well a given PXM and its associated abstract machine perform in order to decide what to implement in SW or HW.

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Timeline



PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Outline

Looking at a PXM Abstract Machine again

Where to Implement a PXM?
- Full Hardware Implementation
- Full Software Implementation
- Hardware-Software Co-Design
- Timeline

Evaluating PXMs' Efficiency [4, 9]
- Analytical Models
- Micro-Benchmarking
- Application Benchmarking
- Evaluating Extensions to a given couple PXM-Abstract Machine

Analytical Models

Description & Purpose

- Based on solid mathematical (often probabilistic/statistical) methods
- For specific features to evaluate
- Provide very useful trends for a given mechanism (when done right)
- Can give very accurate information on the behavior of a system (eg queueing networks)
- Shows its limits when trying to apply to a full system which implements the whole PXM (too many parameters)

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Micro-Benchmarking

- Made to evaluate the overhead induced by individual constructs of the PXM
- They only *verify* a given implementation is efficient, they do not *validate* the PXM does what it is intended to do
- Helps to predict the *minimal* overhead to expect when using the PXM

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Purpose of Application Benchmarking

- Must be representative of the kind of workload the PXM should process
- Helps determine how close (or far) the PXM is from fulfilling its goals – and how efficiently: programmability-wise, speed-wise, etc.

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

What to Measure

For parallel workloads

- Sequential execution (SE_{init}): provide a baseline
- Sequential execution programmed with the PXM (SE_{PXM}): measure the *global* overhead of the PXM
- Parallel execution programmed with the PXM (PE_{PXM})

Time Criterion Example

- SE_{init}/SE_{PXM} gives the global overhead of the given PXM
- SE_{init}/PE_{PXM} gives the *absolute* speedup of the PXM
- SE_{PXM}/PE_{PXM} gives the *relative* speedup of the PXM

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Evaluating Extensions to a given couple PXM-Abstract Machine

Motivation

- Current implementation may incur too much overhead for certain constructs
- Hardware is not necessarily available to test new ideas

Use of simulation

- Function-accurate
- Cycle-accurate
- Gate-accurate

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Evaluating Extensions to a given couple PXM-Abstract Machine

Motivation

- Current implementation may incur too much overhead for certain constructs
- Hardware is not necessarily available to test new ideas

Use of simulation

- Function-accurate
- Cycle-accurate
- Gate-accurate

PXM evaluation & modeling
S.Zuckerman
Looking at a PXM Abstract Machine again
Where to Implement a PXM?
Full Hardware Implementation
Full Software Implementation
Hardware-Software Co-Design
Timeline
Evaluating PXMs' Efficiency [4, 9]
Analytical Models
Micro-Benchmarking
Application Benchmarking
Evaluating Extensions to a given couple PXM-Abstract Machine

Case studies: OpenMP and EARTH

OpenMP

- Share-memory programming model
- One of the most popular (and available) programming models out there

EARTH

- Already seen before
- Hybrid Von Neumann – data flow model of computation
- Evaluated in multiple ways

Outline

The OpenMP Execution Model
- Evaluating OpenMP's efficiency
- Application Benchmarking with OpenMP
- Extending OpenMP

Evaluating EARTH
- Analytical Models for EARTH
- Evaluating EARTH on Off-the-Shelf Computers
- Other Ports of EARTH [15]
- Extending Hardware to be EARTH-compliant

Overview

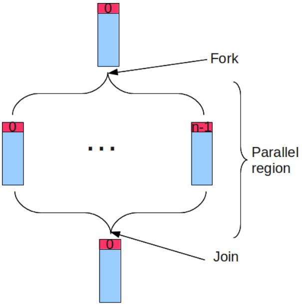
The OpenMP Programming Model [5]

- No specific abstract machine model (relies on Von Neumann's model for threads/processors)
- a language extension to Fortran, C, C++
- a library
- a runtime system

Originally, it was made to express data-parallel and SPMD programs easily.

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

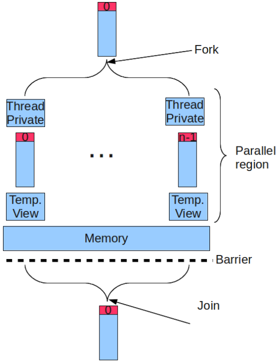
Threading Model: Fork-Join




```
#pragma omp parallel
{
  #pragma omp for
  for (int i = 0; i < M; ++i)
    for (int j = 0; j < N; ++j)
      for (int k = 0; k < K; ++k)
        C[i*N+j] = β * C[i*N+j] +
                   α * A[i*K+k]
                     * B[k*N+j];
}
```

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Memory Model [7] & Synchronization API



Reminder: this is not the complete description of the OpenMP model!

Directive/clause	**Effect**
`nowait`	Removes the implicit barrier of a given directive/clause
`flush(v1,v2,...)`	Forces the variables v_i to be written to (read from) memory (commits these variables from the temporary view to the shared memory).
`critical [(name)]`	Declares a given section of code is a critical section. Only one thread can go in at a time.
Library call	**Effect**
`omp_set_lock (omp_lock_t* lock)`	Tries to acquire lock *lock*
`omp_unset_lock (omp_lock_t* lock)`	Releases a lock *lock*

Table: Example of directives and library calls for synchronization in OpenMP

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Microbenchmarking: Using EPCC [3]

Description

- EPCC microbenchmarks (Edinburgh Parallel Computing Center) evaluate various overheads:
  - Scheduling policies (static, dynamic, guided)
  - Synchronization directives (barrier, single/master, atomic/critical)
  - Privatization directives (private, firstprivate, lastprivate, copyprivate, threadprivate)
- Provides a way to compare different implementations of OpenMP
  - same hardware platform (eg: gcc vs icc)
  - same compiler (eg Itanium2 vs Core 2 Quad)

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Experimental Testbed

Itanium2

- EPIC architecture (VLIW + superscalar)
- Mostly in-order (except for memory operations)
- All caches are private (16KB/256KB/12MB)
- Heat sink (Intel could never go beyond 1.6 GHz)
- Montecito and Montvale differ only w.r.t. the memory bus frequency (533MHz vs 667MHz).
- 2 types of nodes: UMA (Montecito) and NUMA (Montecito, Montvale)

Xeon Woodcrest

- Core 2 family (x86, out-of-order, superscalar, etc.)
- Private L1 cache: 32 KB
- Last level of cache (L2, 4MB) is shared between the 2 cores

Software

OS	Linux (kernel 2.6.18)
Compiler	ICC v10.0

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Example of Results with EPCC

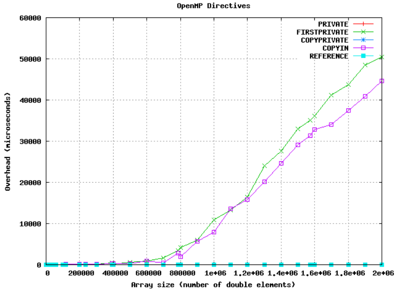


Figure: IA64

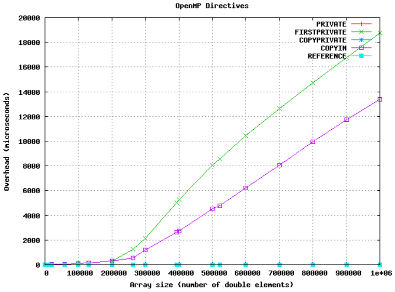


Figure: x86

`arraybench` results

Example of Results with EPCC

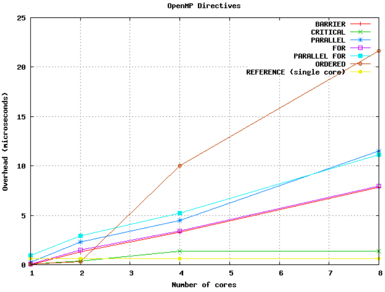


Figure: IA64

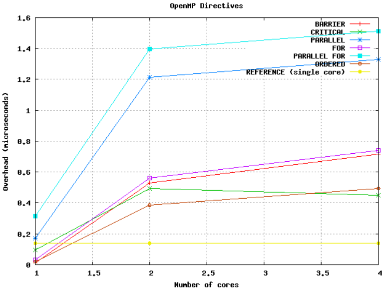


Figure: x86

`syncbench` results

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Example of Results with EPCC

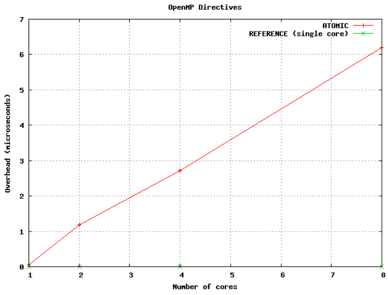


Figure: IA64

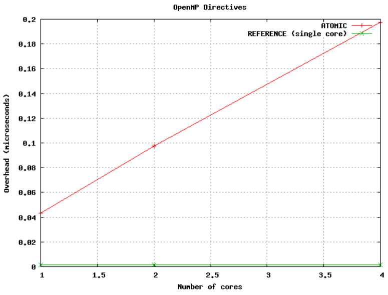


Figure: x86

`atomic` results

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Example of Results with EPCC

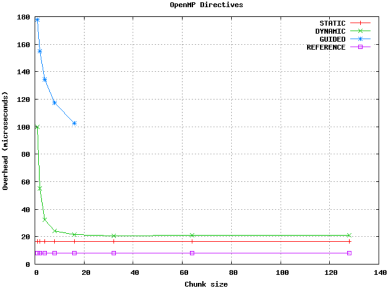


Figure: IA64

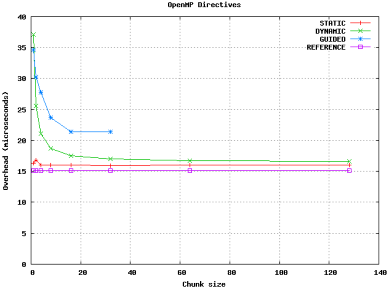


Figure: x86

`schedbench` results

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Application Benchmarking with OpenMP

Name	Description
BT	Simulated CFD: 3D Navier-Stoke equations. Alternating Direction Implicit (ADI) used to solve the finite difference solution to the problem.
SP	Simulated CFD: uses Beam-Warming approximate factorization to solve the finite difference problem.
LU	Simulated CFD: uses symmetric successive over-relaxation (SSOR) to solve a 3D Navier-Stoke equation system. Uses LU matrix decomposition kernels.
FT	3D Fast Fourier Transform (FFT). Based on spectral methods.
MG	3D scalar Poisson equation. solved with a V-cycle MultiGrid method.
CG	Conjugate Gradient used to compute the smallest eigenvalue of a large, sparse, unstructured matrix.
EP	Embarrasingly Parallel benchmark. Goal: provide reference point for all other benchmarks.

Table: NASA Advanced Supercomputing (NAS) Parallel Benchmarks [2, 10]

Name	Application
ammp	Chemistry/biology
applu	Fluid dynamics/ physics
apsi	Air pollution
art	Image recognition/ neural networks
facerec	Face recognition
fma3d	Crash simulation
gafort	Genetic algorithm
galgel	Fluid dynamics
equake	Earthquake modeling
mgrid	Multigrid solver
swim	Shallow water modeling
wupwise	Quantum chromodynamics (QCD)

Table: SPEComp benchmarks [1]

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Extending OpenMP

- Nested parallelism (OpenMP 2.0-2.5)
  - Not implemented in all OpenMP runtime systems yet (it is optional in the standard)
  - Can help handle “static” outer scheduling but “dynamic” inner scheduling
- Going beyond data/loop parallelism: tasks [6] (OpenMP 3.0)
  - Can “flatten” recursive calls
  - Created to handle pointer-chasing
  - For now, performance is rather poor [12]
- Loop coalescing directive (OpenMP 3.0)
- See `http://www.openmp.org`
- Mostly an “evolution” rather than a “revolution”

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Outline

The OpenMP Execution Model

Evaluating OpenMP's efficiency

Application Benchmarking with OpenMP

Extending OpenMP

Evaluating EARTH

Analytical Models for EARTH

Evaluating EARTH on Off-the-Shelf Computers

Other Ports of EARTH [15]

Extending Hardware to be EARTH-compliant

Analytical Models Applied to EARTH (and HTMT)

- Closed Queuing Network theory [11]: models EUs, SUs, output messages, input messages, under certain constraints
- Evaluation of the benefits of percolation [8]. The model predicts potential speedups going from 2 to 11 depending on memory behaviors of the programs, and how high memory latencies are.

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

EARTH-MANNA [13]

The MANNA supercomputer

- Made out of Intel i860 XP processors
  - RISC
  - clocked at 50MHz
  - 16KB L1 cache
- Each node embeds
  - 32MB
  - 2 processors
  - Cache coherence using MESI
  - Custom-designed *link* chip (memory-interconnect interface)
  - connected to other nodes through a 16 $\times$ 16 crossbar

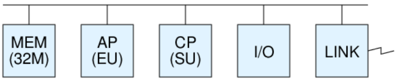
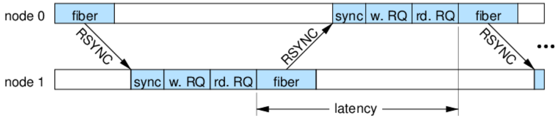


Figure: A MANNA node.

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Microbenchmark Example: ping-pong



Parameter	Dual-processor	Single-processor
Latency (ns)	4091	2450
Latency (cycles)	204.5	122.5
Bandwidth (MB/s)	42.0	28.8
Bandwidth (% of peak)	83.9	57.5

Table: Latency and Bandwidth on EARTH-MANNA

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Microbenchmarks: Operation Latencies

Operation	Dual-processor nodes				Single-processor nodes			
	Sequential		Pipelined		Sequential		Pipelined	
	Loc.	Rem.	Loc.	Rem.	Loc.	Rem.	Loc.	Rem.
(r)sync	2327	3982	841	994	1000	2290	380	668
(r)spawn	2252	4266	N/A	N/A	920	2500	N/A	N/A
get_sync	2824	6968	1137	1880	1440	4666	700	1502
data_(r)sync	2767	6667	1060	1814	1280	4340	560	1200
invoke (1 arg)	5011	9011	3188	2794	2300	5360	1611	2165
invoke (5 args)	6217	10240	3879	2984	2460	5640	1768	2231
invoke (9 args)	6826	10727	4260	3504	3060	6500	2368	3165
invoke (18 args)	8192	12552	5529	4456	3220	7620	2528	3537

Table: EARTH Operation Latencies (nsec.) on EARTH-MANNA

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Microbenchmarks: EU Costs of EARTH Operations

Operation	Dual-processor nodes		Single-processor nodes	
	Local	Remote	Local	Remote
(r)sync	504	504	300	588
(r)spawn	721	580	323	640
end_fiber	530	N/A	441	N/A
incr_(r)sync	561	554	300	620
data_(r)sync	580	606	480	660
get_sync	580	620	620	700
invoke (1 arg)	760	620	479	806
end_procedure (1 arg)	794	N/A	760	N/A
invoke (5 args)	1039	907	599	936
end_procedure (5 args)	1203	N/A	800	N/A
invoke (9 args)	1223	1210	960	1406
end_procedure (9 args)	1372	N/A	1040	N/A
invoke (18 args)	1766	1512	1099	1670
end_procedure (18 args)	1728	N/A	1060	N/A

Table: EARTH-MANNA-D: Cost of forming a request message and writing it to the EQ in memory; for EARTH-MANNA-S: Cost of stopping and performing the entire operation (if local) or forming a request message and writing it to the link chip (if remote)

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Application Benchmarking: Sequential Timings

Benchmark	Input	T_{seq} (sec.)	Description
FFT	2^{16}	0.866	Regular; frequent data moves
Fibonacci	30	0.969	Recursive; high overheads
Matrix multiply	512×512	36.6	Regular, data-parallel
N-Queens-P	12 queens	17.2	Fully para. recursive enumeration
N-Queens-T	12 queens	”	Partially sequentialized
Paraffins	$N = 23$	3.69	Recursive enumeration
Povray	shapes $(256)^2$	69.4	Task-parallel
Protein folding	$3 \times 3 \times 3$	7.43	Recursive search
SLT-2D	80×80	2.60	Regular, data-parallel
Tomcatv	$N = 257$	48.6	Regular, data-parallel, barrier
TSP	10 cities	38.2	Recursive search

Table: Benchmarks and Sequential Performance

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Metrics to Measure EARTH-MANNA's Performance

The USE factor

$USE = T_{seq}/T_1$, with

- T_{seq}: best "pure" sequential execution time
- T_1: execution time using EARTH (Threaded-C program) with a single thread

Parallel Performance Metrics

- Relative speedup on k nodes: $R_k = T_1/T_k$
- Absolute speedup on k nodes: $A_k = T_{seq}/T_k$
- Relationship between R_k and A_k: $A_k = USE \times R_k$

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Application Benchmarking: Uni-Node Support Efficiencies aka USE Factor

Benchmark	USE factor (%)	
	Dual-processor	Single-processor
FFT	59.8	75.6
Fibonacci	7.55	13.9
Matrix multiply	99.9	100.3
N-Queens-P	52.5	67.0
N-Queens-T	98.8	99.3
Paraffins	91.4	99.4
Povray	94.0	100.0
Protein folding	95.0	98.8
SLT-2D	88.5	99.9
Tomcatv	95.0	100.0
TSP	98.9	99.6

Table: Uni-Node Support Efficiencies on EARTH-MANNA

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Application Benchmarking: Relative Speedups

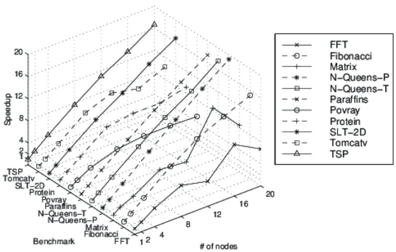


Figure: Single-processor

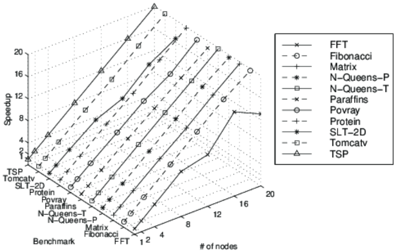


Figure: Dual-processor

PXM evaluation & modeling

S.Zuckerman

The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP

Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Application Benchmarking: Absolute Speedups

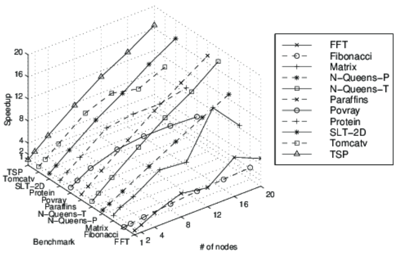


Figure: Single-processor

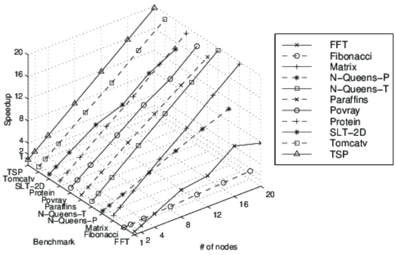


Figure: Dual-processor

Other Ports of EARTH

EARTH on IBM SP2

- Implied changes to Threaded-C (32 bit address space not enough to address more than 4GB)
- Compilation chain changed due to different ISA

EARTH-Beowulf

- Network-of-Workstations
- Fast Ethernet (100Base-T)
- 60-node machine running Povray (presented at CalTech in 1998)
- Inter-node communications pass through TCP/IP

Clusters of SMP Workstations

- 4-way UltraSPARC-II machines
- Shared memory (local crossbar)
- Myrinet network interconnect
- Reuses EARTH-Beowulf implementation
- Handles multiple EUs

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Other Ports of EARTH

EARTH on IBM SP2

- Implied changes to Threaded-C (32 bit address space not enough to address more than 4GB)
- Compilation chain changed due to different ISA

EARTH-Beowulf

- Network-of-Workstations
- Fast Ethernet (100Base-T)
- 60-node machine running Povray (presented at CalTech in 1998)
- Inter-node communications pass through TCP/IP

Clusters of SMP Workstations

- 4-way UltraSPARC-II machines
- Shared memory (local crossbar)
- Myrinet network interconnect
- Reuses EARTH-Beowulf implementation
- Handles multiple EUs

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Other Ports of EARTH

EARTH on IBM SP2

- Implied changes to Threaded-C (32 bit address space not enough to address more than 4GB)
- Compilation chain changed due to different ISA

EARTH-Beowulf

- Network-of-Workstations
- Fast Ethernet (100Base-T)
- 60-node machine running Povray (presented at CalTech in 1998)
- Inter-node communications pass through TCP/IP

Clusters of SMP Workstations

- 4-way UltraSPARC-II machines
- Shared memory (local crossbar)
- Myrinet network interconnect
- Reuses EARTH-Beowulf implementation
- Handles multiple EUs

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Extending Hardware to be EARTH-compliant

Why Extend EARTH?

- EARTH was designed to run on off-the-shelf multiprocessor computers
- What if a specialized computer was built for EARTH?
- Use of SEMi [14]: Simulator of EARTH-MANNA on i860 (single-threaded, cycle-accurate to some degree)
- Speed ratio: $\approx 300 - 500$ times slower than reality (which is not bad!)

Additional Hardware Features

- Extension of the machine from 20 to 120 nodes
- Modification of the i860:
  - Models changes to the network topology ($n \times n$ network of routers)
  - Parameterized caches and memory delays
  - Added scoreboard logic (instead of locking the whole functional unit)
  - Non-blocking on-chip L1 cache
  - Added an L2 cache
  - Added in-order, multiple instruction issue (instead of the limited VLIW capabilities of the i860)

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Extending Hardware to be EARTH-compliant

Why Extend EARTH?

- EARTH was designed to run on off-the-shelf multiprocessor computers
- What if a specialized computer was built for EARTH?
- Use of SEMi [14]: Simulator of EARTH-MANNA on i860 (single-threaded, cycle-accurate to some degree)
- Speed ratio: $\approx 300 - 500$ times slower than reality (which is not bad!)

Additional Hardware Features

- Extension of the machine from 20 to 120 nodes
- Modification of the i860:
  - Models changes to the network topology ($n \times n$ network of routers)
  - Parameterized caches and memory delays
  - Added scoreboard logic (instead of locking the whole functional unit)
  - Non-blocking on-chip L1 cache
  - Added an L2 cache
  - Added in-order, multiple instruction issue (instead of the limited VLIW capabilities of the i860)

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Results after simulation: USE Factor

Benchmark	Input	T_{seq} (sec)	USE factor (%)	
			Dual-processor	Single-processor
Fibonacci	15	0.000831	8.6	15.7
	20	0.00801	7.7	14.1
	25	0.0875	7.6	13.9
	30	0.969	7.6	13.9
N-Queens-P	8	0.0223	39.9	51.7
	10	0.541	46.8	56.1
	12	17.3	53.9	65.6
N-Queens-T	8	0.0223	68.5	78.5
	10	0.541	93.1	95.3
	12	17.3	99.1	99.3
Paraffins	18	0.0394	82.1	97.6
	20	0.228	85.4	101.4
	23	3.69	84.7	100.6
Tomcatv	33	0.721	89.3	92.2
	65	2.94	91.4	93.7
	129	12.0	93.2	95.6
	257	48.7	93.7	96.5

Table: Uni-Node Support Efficiencies on SEMi Simulation of EARTH-MANNA

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Results after simulation: Fibonnaci

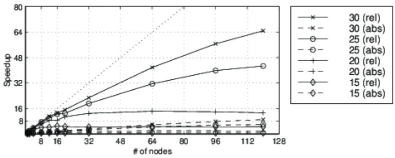


Figure: EARTH-MANNA-S

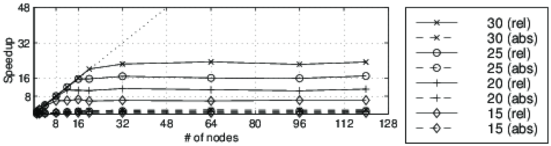


Figure: EARTH-MANNA-D

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Results after simulation: N-Queens-P

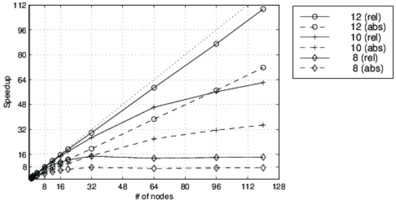


Figure: EARTH-MANNA-S

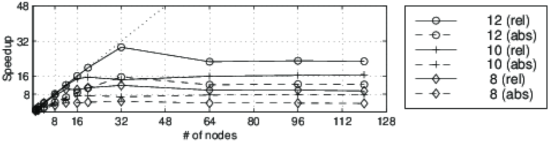


Figure: EARTH-MANNA-D

PXM evaluation & modeling

S.Zuckerman

The OpenMP Execution Model

Evaluating OpenMP's efficiency

Application Benchmarking with OpenMP

Extending OpenMP

Evaluating EARTH

Analytical Models for EARTH

Evaluating EARTH on Off-the-Shelf Computers

Other Ports of EARTH [15]

Extending Hardware to be EARTH-compliant

Results after simulation: N-Queens-T

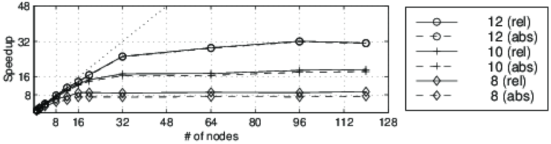


Figure: EARTH-MANNA-S

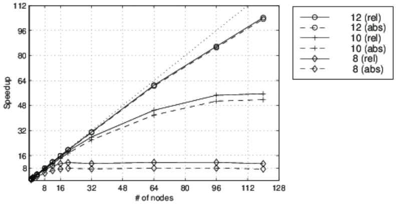


Figure: EARTH-MANNA-D

PXM evaluation & modeling
S.Zuckerman
The OpenMP Execution Model
Evaluating OpenMP's efficiency
Application Benchmarking with OpenMP
Extending OpenMP
Evaluating EARTH
Analytical Models for EARTH
Evaluating EARTH on Off-the-Shelf Computers
Other Ports of EARTH [15]
Extending Hardware to be EARTH-compliant

Results after simulation: Paraffins

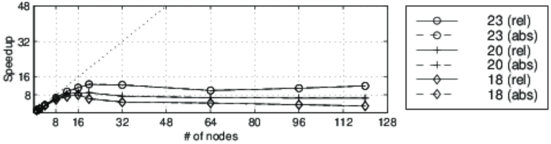


Figure: EARTH-MANNA-S

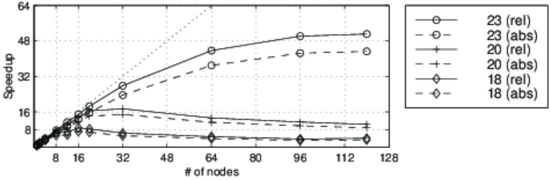


Figure: EARTH-MANNA-D

Results after simulation: Tomcatv

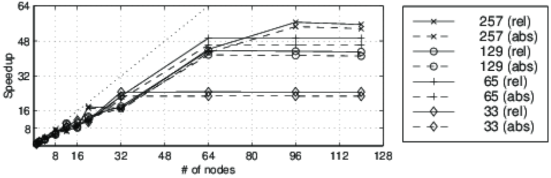


Figure: EARTH-MANNA-S

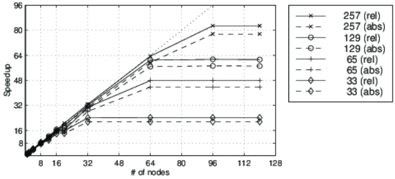


Figure: EARTH-MANNA-D

PXM evaluation & modeling

S.Zuckerman

The OpenMP Execution Model

Evaluating OpenMP's efficiency

Application Benchmarking with OpenMP

Extending OpenMP

Evaluating EARTH

Analytical Models for EARTH

Evaluating EARTH on Off-the-Shelf Computers

Other Ports of EARTH [15]

Extending Hardware to be EARTH-compliant

What to Take Home (for now!)

Decide What to Model

- Communication?
- Context-switch?
- Latency vs throughput
- etc.

Decide How to Model

- Analytical
- Real measurements on (imperfect) hardware
- Simulation of enhancements to make to the HW

Define a Set of Benchmarks

- Microbenchmarks: must evaluate (*verify*) the quality of the PXM implementation
- Application benchmarks: must be representative (*validate*) of the workloads the PXM is supposed to help process

Bibliography I

V. Aslot, M. Domeika, R. Eigenmann, G. Gaertner, W. Jones, and B. Parady.
Specomp: A new benchmark suite for measuring parallel computer performance.
In R. Eigenmann and M. Voss, editors, *OpenMP Shared Memory Parallel Programming*, volume 2104 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin / Heidelberg, 2001.
10.1007/3-540-44587-0_1.

D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, L. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga.
The nas parallel benchmarks–summary and preliminary results.
In *Proceedings of the 1991 ACM/IEEE conference on Supercomputing*, Supercomputing '91, pages 158–165, New York, NY, USA, 1991. ACM.

J. M. Bull and D. O'Neill.
A microbenchmark suite for openmp 2.0.
SIGARCH Comput. Archit. News, 29:41–48, December 2001.

Bibliography II

J. P. Buzen.
Fundamental laws of computer system performance.
In *Proceedings of the 1976 ACM SIGMETRICS conference on Computer performance modeling measurement and evaluation*, SIGMETRICS '76, pages 200–210, New York, NY, USA, 1976. ACM.

L. Dagum and R. Menon.
Openmp: an industry standard api for shared-memory programming.
Computational Science Engineering, IEEE, 5(1):46 –55, jan-mar 1998.

A. Duran, J. Corbalán, and E. Ayguadé.
Evaluation of openmp task scheduling strategies.
In *Proceedings of the 4th international conference on OpenMP in a new era of parallelism*, IWOMP'08, pages 100–110, Berlin, Heidelberg, 2008. Springer-Verlag.

J. Hoeflinger and B. de Supinski.
The openmp memory model.
In M. Mueller, B. Chapman, B. de Supinski, A. Malony, and M. Voss, editors, *OpenMP Shared Memory Parallel Programming*, volume 4315 of *Lecture Notes in Computer Science*, pages 167–177. Springer Berlin / Heidelberg, 2008.
10.1007/978-3-540-68555-5_14.

Bibliography III

A. Jacquet, V. Janot, C. Leung, G. R. Gao, R. Govindarajan, and T. L. Sterling.
An executable analytical performance evaluation approach for early performance prediction.
Parallel and Distributed Processing Symposium, International, 0:268a, 2003.

R. Jain.
The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.
Wiley professional computing. Wiley, 1991.

H. Jin, M. Frumkin, and J. Yan.
The openmp implementation of nas parallel benchmarks and its performance.
Technical report, 1999.

S. Nemawarkar and G. Gao.
Measurement and modeling of earth-manna multithreaded architecture.
In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 1996. MASCOTS '96., Proceedings of the Fourth International Workshop on*, pages 109 –114, feb 1996.

Bibliography IV

S. Olivier and J. Prins.
Comparison of openmp 3.0 and other task parallel frameworks on unbalanced task graphs.
International Journal of Parallel Programming, 38:341–360, 2010.
10.1007/s10766-010-0140-7.

K. J. Theobald.
EARTH: An Efficient Architecture for Running Threads.
PhD thesis, University of Delaware, 1999.

K. J. Theobald.
Semi: A simulator for earth, manna, and i860.
Technical report, University of Delaware, March 1999.

W. Zhu, Y. Niu, and G. R. Gao.
Performance portability on earth: a case study across several parallel architectures.
Cluster Computing, 10:115–126, June 2007.