



Experiences with First-Generation Itanium at NCSA

Rick Kufrin

Scientific Computing Division

National Center for Supercomputing Applications

University of Illinois at Urbana-Champaign

`rkufrin@ncsa.uiuc.edu`

3rd UKHEC Annual Seminar

10 December, 2002

Daresbury Laboratory

Daresbury, UK

Topics

- **History of NCSA HPC systems**
- **NCSA Linux clusters**
- **Application examples**
- **Performance measurements**
- **Performance tools**
- **Anticipating the future (!?!)**

Not only about Itanium, but also IA-64/Linux

NCSA Major System History

- **1985 - 1994:**
Cray X-MP/Y-MP/2 (**CTSS / UNICOS**)
- **1989 - 1997:** CM-2/CM-5 (**CM OS**)
- **1994 - present:**
SGI Power Challenge, Origin 2000 (**IRIX**)
- **1998:** IA-32 “SuperCluster” (**WinNT**)
- **2000 - present:** PIII cluster (**Linux**)
- **2002 - present:** Itanium cluster (**Linux**)
- **2003:** IBM p690 (**AIX**), Itanium 2 (**Linux**)

Current NCSA Linux Clusters

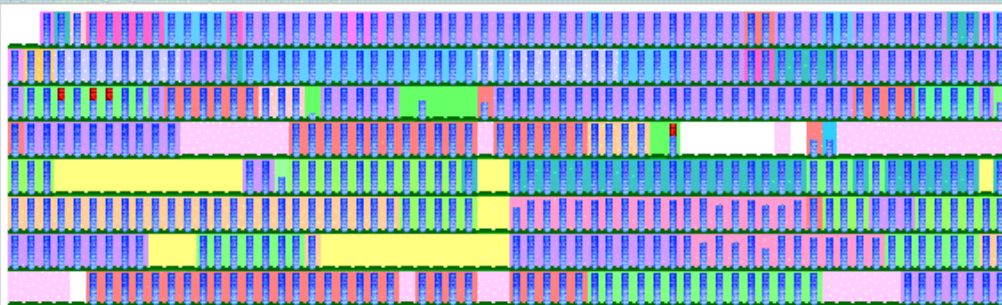
	Platinum (IA-32)	Titan (IA-64)
Processor	Intel 1GHz Pentium III	Intel 800MHz Itanium
Cache	256 KB L2 cache	4 MB L3 cache
Memory	1.5 GB Compute, 2GB Access	2 GB / Node
Peak Performance	1 Gflop / processor	3.2 Gflops / processor
Compute Nodes	516 (1032 processors)	160 (320 processors)
Access Nodes	4 (8 processors)	2 (4 processors)
Interconnects	Myrinet 2000, Gigabit Ethernet, 100Mbit Ethernet	Myrinet 2000, Gigabit Ethernet
Peak Performance	1 Tflops total	1 Tflops total
Operating System	Red Hat Linux 7.2 Linux 2.4.9 SMP	Red Hat Linux 7.1 Linux 2.4.16 SMP
Programming Model	MPI / VMI (Virtual Machine Interface)	MPI / VMI
Production Use	July 23 2001	April 15 2002

NCSA's Cluster Monitor

PLATINUM Cluster Monitor

Main
Hosts
Resources
Queues
Jobs
Alerts
Adm Notes
Help

Nodes: 516



Job	Owner	Job Name	Queue	State	Nodes	Time Used	% Time Allowed	Max Time Allowed
12876	bela	rand48	standard	RUNNING	32	18:44:56	104	24:00:00
12885	villa	lac-a	standard	RUNNING	128	05:37:44	31	24:00:00
12893	ruiqiao	ye40-init2	standard	RUNNING	1	19:34:01	82	23:55:00
12898	ruiqiao	ye41-init2	standard	RUNNING	1	19:27:13	81	23:55:00
12901	zong	sub0.73	standard	RUNNING	28	03:54:53	22	24:00:00
12902	zong	sub0.88	standard	RUNNING	28	01:16:29	7	24:00:00
12903	zong	sub1.0	standard	RUNNING	28	01:16:03	7	24:00:00
12905	ruiqiao	ye42-init2	standard	RUNNING	1	19:27:11	81	23:55:00
12906	ruiqiao	ye43-init2	standard	RUNNING	1	19:15:47	81	23:55:00

Key User Concerns Today

- **As of Sept '02: ~100 projects, ~450 accounts**
 - large increase in recent requests for time on Itanium cluster (ALL available time is allocated)
- **New users encounter similar problems**
 - Batch system / Scheduling
 - Network issues (Myrinet / MPI)
 - Compiler issues
 - Porting (to 64-bit / EPIC)

Representative Applications

- **NAMD (Klaus Schulten)**
 - C++ / Charm++ molecular dynamics
- **GenIDLEST (Danesh Tafti)**
 - F77 / F90 / MPI turbulent flow
- **NCOMMAS (Lou Wicker / Bob Wilhelmson)**
 - F90 / OpenMP weather modeling
- **CACTUS (Ed Seidel)**
 - Multi-language PSE developed for relativity, extensible to other domains
- **PHASTA (Ken Jansen)**
 - Fluid flow/mass transport simulator

Compiler Support

- **Compiler is first exposure/experience**
 - not unusual to see reports of similar or worse initial performance vs. Pentium
 - C++ codes currently vary widely
- **Compiler-generated info/feedback can make a difference:**
 - knowledge of important compiler switches
 - profile-guided optimization
 - optimization reports
 - libraries (e.g., MKL)

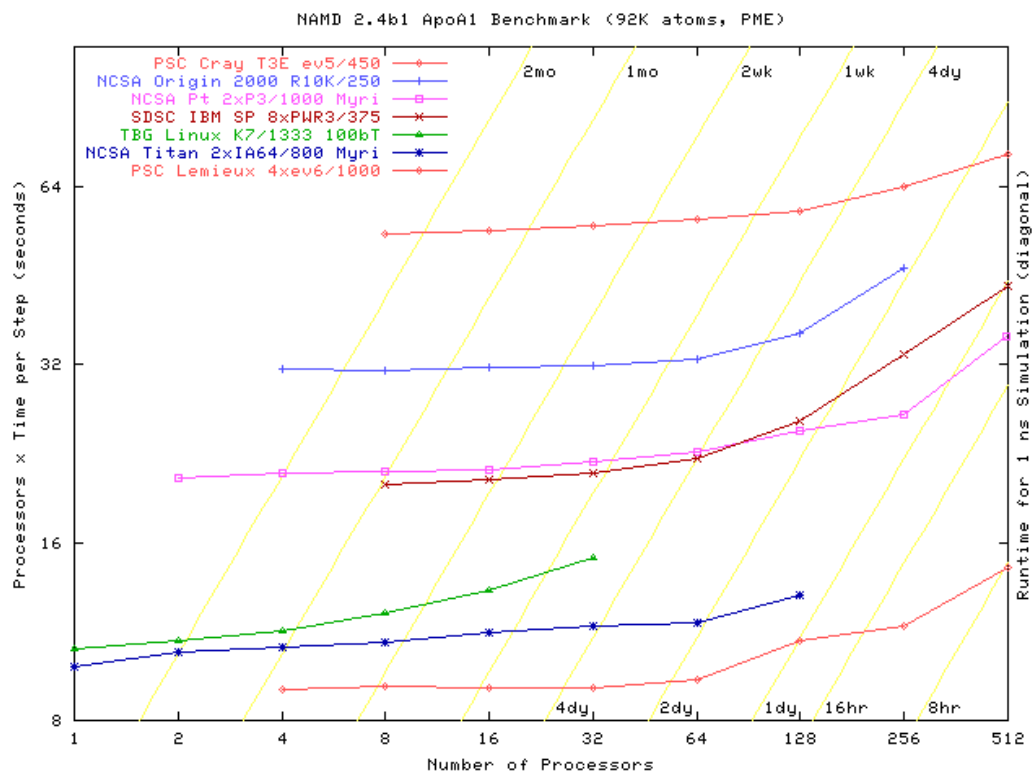
<http://www.intel.com/software/products/compilers/>

“Out-Of-The-Box” Tools

- **time**
- **gprof**
 - initially non-functional, required glibc patches from early 2002
 - pthread support not functioning (MPI/VMI profiling affected)
- **strace**
- **/proc filesystem**
- **... and others**

NAMD (UIUC Theoretical Biophysics)

- Molecular dynamics (C++)
- Gordon Bell Award winner (SC '02)
- Highly-scalable, message-driven parallelism through Charm++ (UIUC Parallel Programming Laboratory)



NAMD, cont'd

- One of the first applications to run successfully on NCSA Itanium cluster
- Initial attempt at ecpc compiler build: performance on par with PIII 1GHz
 - Optimization report information indicated failure to software pipeline key innermost (non-bonded calculation) loops. “Could not estimate trip count”
 - Recompile with compiler feedback profiles doubled per-node performance
- Recent work by the NAMD developers has uncovered/reported a basic compiler flaw (`ostrstream`) => further perf. improvement

<http://www.ks.uiuc.edu/Research/namd>

Aside: C++ “Abstraction Penalty”

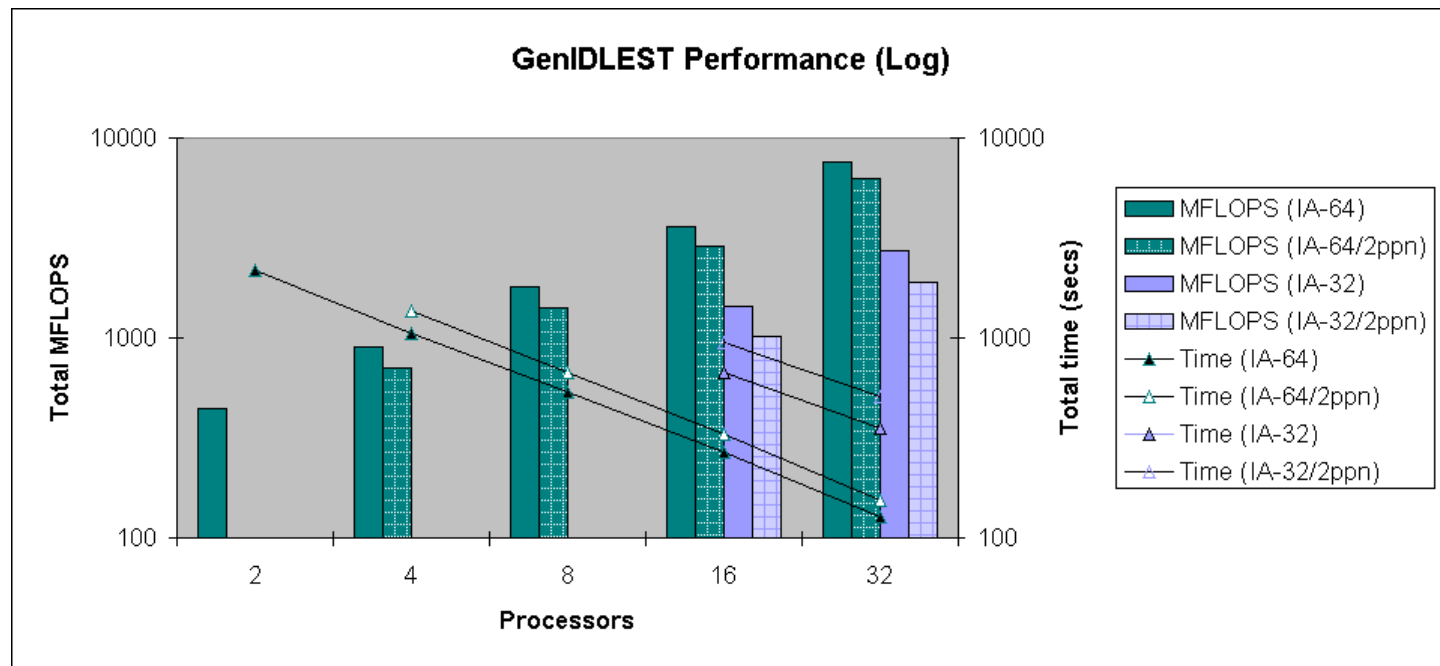
- 1 million 3-dimensional double precision elements
 - $A = B + 1.3 * C + 1.7 * D$

	(MFLOPS)	1	2	penalty (%)	3	4	penalty (%)	5	6
modi4	CC 7.3.1.3m	5	13	58	2	8	70	26	26
	KCC 4.0	22	22	3	22	25	11	25	31
	GNU 2.95.3	11	14	19	13	16	24	19	21
platinum	Intel 6.0	14	23	40	21	50	58	50	59
	GNU 2.96	12	48	74	28	49	43	48	49
	PGI 3.3-2	4	8	54	4	21	83	50	50
titan	Intel 6.0	13	26	49	16	45	64	141	141
	GNU 2.96	5	12	58	10	21	53	21	18

- 1. $A = B + 1.3 * C + 1.7 * D$, 2. `for(int i=0; i<n; i++) A[i] = B[i] + 1.3*C[i] + 1.7*D[i];`
- 3. `a = b + 1.3*c + 1.7*d`, 4. `for(int i=0; i<ntot; i++) a[i] = b[i] + 1.3*c[i] + 1.7*d[i];`
- 5. C inline function of #4 with restrict keyword, 6. Fortran subroutine of #4

GenIDLEST (VA Tech ME)

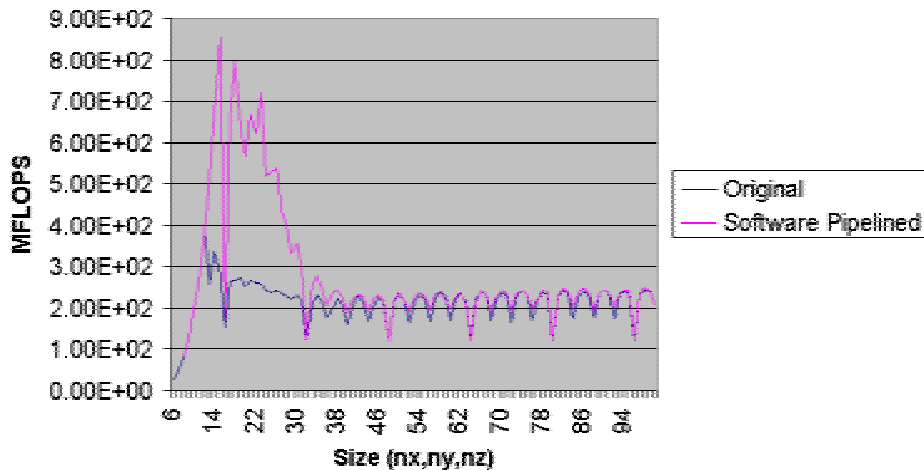
- **CFD - one of the more heavily-used applications on NCSA's Itanium cluster - turbulent flows in complex geometries**
- **Execution time dominated by sparse matrix-vector operations (stencils), linear solvers**



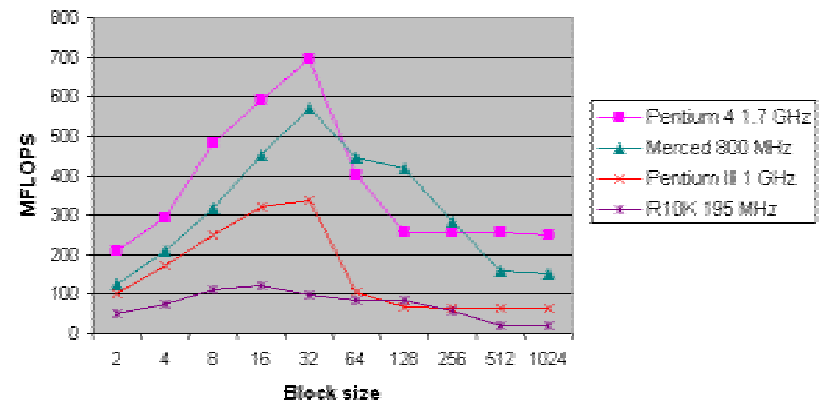
GenIDLEST, cont'd

The ASPCG kernel is a good overall indicator of “full-blown” code performance. Bandwidth limitations apparent.

GenIDLEST Matvec Cache Threshing



Single-processor ASPCG Kernel (D. Tafti)



Sparse matrix-vector multiply (19-pt stencil) also a key indicator. Here the need for array padding and software pipelining can be seen.

NCOMMAS (SSL, UIUC)

- **SSL cloud dynamics/model**
- **Joint optimization work by atmospheric researchers, Rice University, Univ. of Minnesota (P. Woodward)**
- **Initial port => poor performance, *excessive* system time provides a clue: denormalized number kernel traps. “-ftz” yielded *90% reduction in runtime.***
- **Collaboration with Rice focuses on compiler efforts & processing for cache optimization**

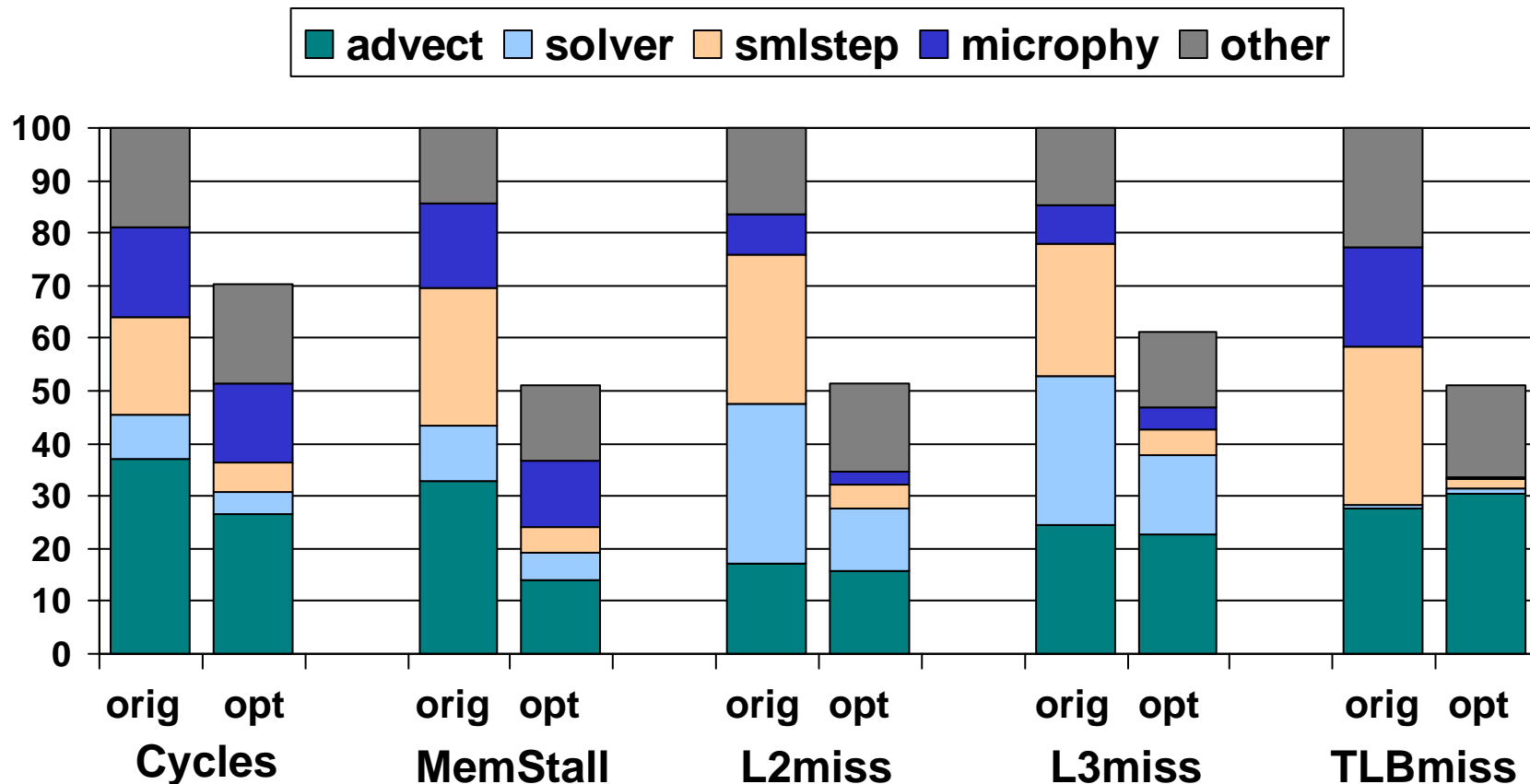
<http://www.nssl.noaa.gov/~wicker/commas.html>

Rice / HiPerSoft Compiler Support

- **Multi-level loop fusion and blocking**
 - Improve cache reuses by reducing reuse distance
- **Time skewing**
 - Exploit temporal reuse across time steps
- **Unroll & Jam**
 - Exploit temporal reuse carried by outer loops
 - Improve register reuse
- **Array contraction for fused, blocked, unroll & jammed codes**
 - Reduce data footprints
 - Improve cache reuses
- **Guard-free Core Code Generation**
 - Generate efficient computation core with no if-conditions

NCOMMAS Optimizations

Data: 81x81x36 Time steps: 1800
compiled w/ efc -O3



CACTUS (Max Planck Institute)

- **Open Source PSE (Problem Solving Environment) for scientists and engineers**
- **Very full-featured, modular, parallel, Grid-enabled**
- **Large, active development group
Originated in academia (gravitation / relativity);**

<http://www.cactuscode.org/>

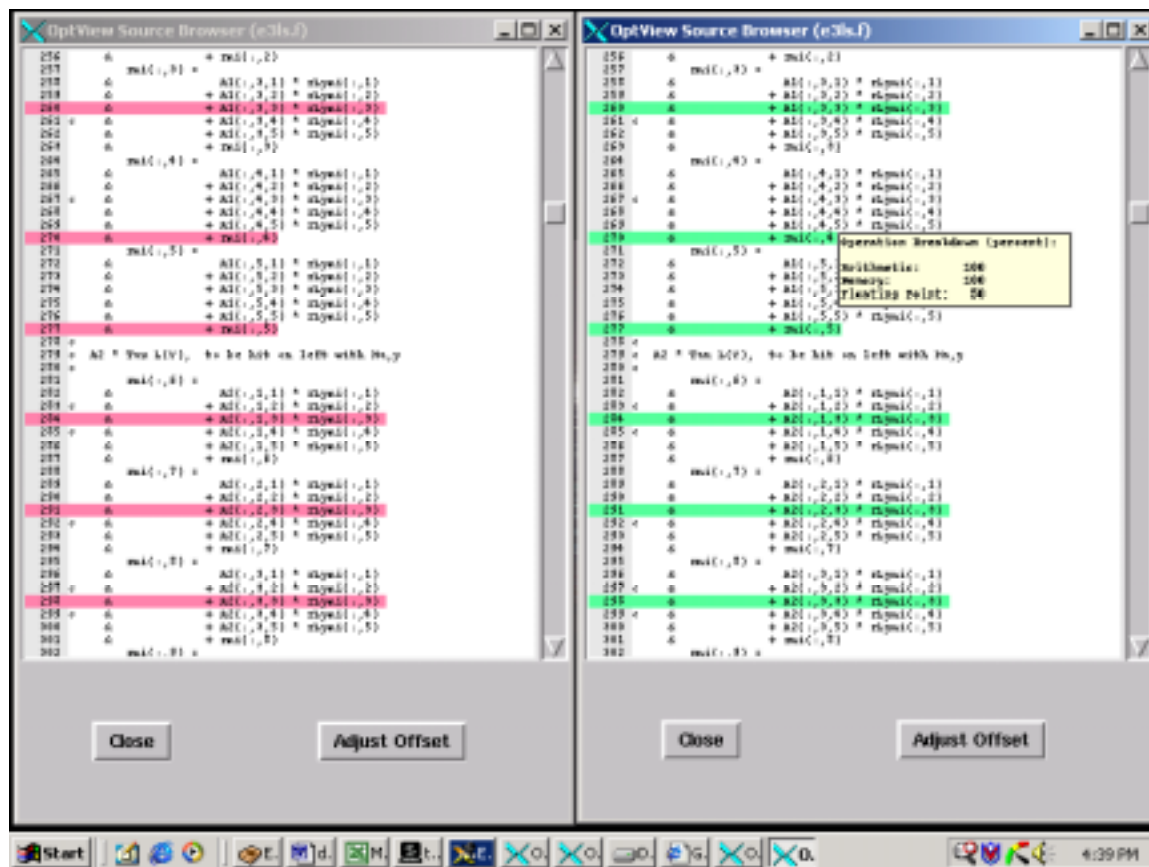
CACTUS, cont'd

- Initial performance of simplified kernels very promising, however actual code (more complex) not the same story => 0.5 % of peak
- Current version of code now at 4% of peak, but much improvement remains to be realized
- *Dependent on compiler maturity: key problem is extremely complex loops that exceed compiler internal optimization limits. Problem is being addressed by Intel.*

PHASTA (Rensselaer Polytechnic)

- Initial difficulties with compiler support for F90 modules (fixed in efc 7.0)

Once the code successfully compiled, software pipelining issues next (-O2 found to be better): 20% overall improvement



Performance Tools On NCSA IA-64

- **Perfmon: Hardware counters (HWPC)**
- **PAPI: HWPC, wallclock, sysinfo**
- **SvPablo: Timers, HWPC**
- **HPM Toolkit: HWPC**
- **VProf: HWPC**
- **PerfSuite: HWPC, compiler reports, MPI**

Perfmon (H-P)

- **Strong support for accessing native IA-64 counters**
- **Sample command line tool `pfmon` provided**
- **Kernel incompatibility from 2.4.18 up**
 - Very new (last week) `pfmon` test version announced by author
- **Not a 1st choice for the general user**
 - Wrapper tool `ps_run` written locally for ease of use, derived metrics (invoked on that basis)

PAPI (ICL / UTK)

- **Foundation for several of the tools used in this talk and analysis work**
- **Usage and awareness increasing**
 - tool / algorithm development
 - benchmarking and system evaluation
- **Active user community, support, development**
- **Installed kernel version *critical***

<http://icl.cs.utk.edu/projects/papi/>

HPM Toolkit (IBM ACTC)

- **Command-line, API, and viz tool for performance counter access**
 - Itanium support as of summer 2002
 - API (PAPI-based) and viz tool operational, command-line utility planned
- **Rich set of derived metrics, mpx-aware**
- **Also available for x86 (Linux) and Power3/Power4 (AIX)**

<http://www.alphaworks.ibm.com/tech/hpmtoolkit/>

VProf (Sandia)

- **Distributed version not yet w/official support for IA-64/Linux**
 - x86 provides profil(), perfctr, and PAPI access
- **PAPI updates to IA-64 substrate enabled VProf profiling w/ PAPI_profil()**
- **Kernel support @ NCSA initially limited to soft interrupts thru PAPI**
- **Very new, still investigating / testing**
 - NCSA mods to VProf implemented

<http://aros.ca.sandia.gov/~cljanss/perf/vprof/>

OptView (NCSA)

The screenshot shows the OptView application window with a menu bar (File, Options, Help) and a toolbar. The main content is divided into three sections: Source, Functions, and Reports.

Source

File	Directory	Loop Reports
ComputeNonbondedBase.h	src	42
ComputeNonbondedUtil.C	src	46
xlocale	/usr/local/intel/compiler60/ia64/incl	1
xlocnum	/usr/local/intel/compiler60/ia64/incl	7

Functions

Function	Loop Reports	Pipelined
_ZN20ComputeNonbondedUtil19su	46	46

Reports

Line	Function	Pipeline Stages
72	_ZN20ComputeNonbondedUtil19su	2
72	_ZN20ComputeNonbondedUtil19su	10
73	_ZN20ComputeNonbondedUtil19su	2
73	_ZN20ComputeNonbondedUtil19su	10
74	_ZN20ComputeNonbondedUtil19su	2
74	_ZN20ComputeNonbondedUtil19su	10
75	_ZN20ComputeNonbondedUtil19su	2
75	_ZN20ComputeNonbondedUtil19su	10

The screenshot shows the OptView Directory Summary window for the directory /u/ncsa/rkufrin/apps/CN2/cn2. It displays a table of optimization reports with columns for Report File, Source File, Reports, SWP, and SWP.

Report File	Source File	Reports	SWP	SWP
att_order.opt	att_order.c	5	2	3
cn.c.opt	cn.c	13	6	7
debug.opt				
example.opt	example.c	2	0	2
execute.opt	execute.c	11	2	9
filter.opt	filter.c	5	0	5
heap.opt	heap.c	7	0	7
interact.opt	interact.c	2	0	2
interact_utils.opt	interact_utils.c	12	3	9
list.opt	list.c	1	0	1
main.opt				
names.opt	names.c	7	1	6
peccles.opt	peccles.c	10	7	3
print_gen_thing.opt	print_gen_thing.c	7	0	7
quickfit.opt	quickfit.c	3	2	1
robin.opt	robin.c	1	0	1
rule_reader.opt				
specialise.opt	specialise.c	14	3	11
test.opt				
trace.opt	trace.c	8	0	8

Graphical interface to Intel compiler optimization reports

<http://perfsuite.ncsa.uiuc.edu/>

Looking Toward The Future

- **NCSA moved to the IA-64 processor to gain experience with the architecture**
- **Intent: in-house and early user exposure**
- **Demand has turned out higher than expected and researchers are eager to work for more performance**
- **NCSA's Itanium cluster rapidly reached the 90% utilization mark after production**
- **Itanium 2 / TeraGrid right around the corner - early results & non-disclosure experiences very promising**

Conclusion

- Many users still getting apps running
- Collaborations / “expeditions” are the focus - bringing teams together
- Systems upgrades slow but important
- Awareness of and exposure to tools
 - Web-based tool advisor developed
 - Linux Clusters Institute applications track
- ***Compiler maturation and next-generation improvements are key***

Thanks To...

- **Rob Fowler, John Mellor-Crummey & Guohua Jin (Rice University / HiPerSoft)**
- **Jim Phillips (UIUC Theoretical Biophysics)**
- **Ed Seidel and the CACTUS team (Max Planck Institute)**
- **Danesh Tafti (Virginia Tech)**
- **John Towns (NCSA SCD)**
- **Greg Bauer, Wai-Yip Kwok, John Towns (NCSA / SCD)**