Compilers and Compiler-based Tools for HPC

John Mellor-Crummey

Department of Computer Science Rice University

http://lacsi.rice.edu/review/2004/slides/Compilers-Tools.pdf



High Performance Computing



- Good scalability requires effective parallelizations
- Good performance requires efficient node programs
- Compiler technology can reduce the application development burden



Some ASC HPC Challenges for CS

 Understand interplay between ASC applications and architectures

-pinpoint performance bottlenecks to guide application tuning

-evaluate the promise of future architectures

- Support tailoring of applications to target architectures of ASC interest (current and emerging systems)
- Achieve high efficiency on a range of modern processors
- Make it easier to write high-performance programs for scalable parallel systems



Summary: Impact Now and in Future

Long Term Research Affecting Future HPC Systems

High-level data-parallel programming systems

Source-to-source transformation tools

Open source compilers

Compilation techniques for SPMD languages (UPC, CAF)

Performance diagnosis tools

Compilation for hybrid architectures

Fundamental compiler algorithms

Detailed performance modeling of applications

Hand application of aggressive transformations to important codes

Immediate Impact in Support of ASC Mission Goals



Outline

- Performance analysis
 - -HPCToolkit for analyzing node program performance
- Performance modeling

-understanding performance of node programs

- Compiler technology for parallel languages
 - -Co-array Fortran
 - -high-level data parallel languages (HPF)
- Open source platforms for delivering compiler technology
- Compiler technology for node programs



Performance Analysis

- Long-term compiler and architecture research requires detailed performance understanding
 - -identify sources of performance bottlenecks in complex applications
 - -discover automatic strategies for performance improvement
 - -understand the mismatch between application needs and architecture capabilities
- Short-term result: Programmer-accessible tools for understanding application performance



HPCToolkit

- Goal: Effective tools for performance analysis
 - -intuitive top-down user interface
 - -provide information crucial for analysis and tuning
- Platform, language and compiler independence
 - -emphasis on LANL ASC platforms
 - -multiple data sources enable cross-platform comparisons
 - -extract hierarchical program structure from binaries
 - handle multi-module, multi-language (F77, F9x, C, C++, ...)
- Eliminate manual labor from the analyze-tune-run cycle









launch unmodified, optimized application binaries
 collect statistical profiles of events of interest



-decode instructions and combine with profile data





-extract loop nesting information from executables





-synthesize new metrics by combining metrics -relate metrics, structure and program source





—support top-down analysis with interactive viewer —analyze results anytime, anywhere



HPCToolkit on LANL's POP

00	0			D POP						
File										
state_	mod.f90									
Ă	417	mwifdens0t2 = mwifdp0	s0t2							
ă	418	mwifdens0t3 = mwifdp0s0t3 + p**2 * mwifdp2s0t3								1
ð	419	mwifdens0t4 = mwifdp0s0t4								
ð	420	mwjfdens1t0 = mwjfdp0	s1t0							
0	421	mwjfdens1t1 = mwjfdp0	s1t1							
0	422	mwjfdens1t3 = mwjfdp0	is1t3							
0	423	mwjfdensqt0 = mwjfdp0sqt0								
0	424	4 mwjfdensqt2 = mwjfdp0sqt2								
~	425									
Q.	426 WORK2 = mwjfdens0t0 + TQ * (mwjfdens0t1 + TQ * (mwjfdens0t2 + &									
9	427	TQ * (mwjfdens0t3 + mwjfdens0t4 * TQ))) + &								
8	428	428 SQ * (mwjfdens1t0 + TQ * (mwjfdens1t1 + TQ*TQ*mwjfdens1t3)+ &								
\boldsymbol{v}	429 SQR * (mwjfdensqt0 + TQ*TQ*mwjfdensqt2))									
8	430	DENOME - 41 (WORK)								
0	431 422	DENOMIN = CIYWORNZ			Δnn	∩tat	ed So	urc	ο Viev	N/
A	431 432 433	if (present/RHOOLIT)) the	2		Ann	otat	ed So	urc	e Viev	N
8	431 432 433 434	if (present(RHOOUT)) the	n NOMK		Ann	otat	ed So	ourc	e Viev	N
8	431 432 433 434	if (present(RHOOUT)) the RHOOUT = WORK1*DE	n NOMK		Anno	otat	ed So	ourco	e Viev	N
8	431 432 433 434 Scopes	if (present(RHOOUT)) the RHOOUT = WORK1*DE	n Nomk PAF	РІ_ТОТ ⊽	Ann PAPI_TO	otat	ed So	urco	e Viev	N B
B	431 432 433 434 5copes	if (present(RHOOUT)) the RHOOUT = WORK1*DE	n NOMK PAF	PI_TOT▼ 38e10	Ann <u>PAPI_TO</u> 3.79e10	otat	ed So PAPI_L2 3.33e08		e View	₩ B
Exp	431 432 433 434 5copes eriment Agg	if (present(RHOOUT)) the RHOOUT = WORK1*DE	n NOMK PAF 5.8	PI_TOT ▽ 38e10 95e09 15.2%	Ann PAPI_TO 3.79e10 5.20e09	otat T	ed So PAPI_L2 3.33e08 1.43e07	 4.3%	e Viev PAPI_TLI 4.44e07 3.60e06	N B 8.1%
Exp	431 432 433 434 Scopes eriment Agg state	if (present(RHOOUT)) the RHOOUT = WORK1*DE	n NOMK PAF 5.8 8.9	PI_TOT ▼ 38e10 95e09 15.2% 34e09 3.3%	Ann PAPI_TO 3.79e10 5.20e09 1.64e09	otat T 13.7% 4.3%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05	 4.3%	e View PAPI_TLI 4.44e07 3.60e06 8.19e05	N B 8.1% 1.8%
Exp	431 432 433 434 Scopes eriment Agg state	if (present(RHOOUT)) the RHOOUT = WORK1*DE	n NOMK PAF 5.8 8.9 1.9	PI_TOT ▼ 38e10 95e09 15.2% 34e09 3.3% 34e09 3.1%	Ann PAPI_TO 3.79e10 5.20e09 1.64e09 1.55e08	otat T 13.7% 4.3% 0.4%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05	4.3% 0.1%	PAPI_TLI 4.44e07 3.60e06 8.19e05 9.83e04	N B 8.1% 1.8% 0.2%
Exp	431 432 433 434 Scopes eriment Agg state 1 loop at st 1 loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 303	n NOMK PAF 5.8 8.9 1.9 1.8 1.5	PL_TOT ▼ 38e10 95e09 15.2% 34e09 3.3% 34e09 3.1% 38e09 2.7%	Anno PAPI_TO 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08	0tat T 13.7% 4.3% 0.4%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05	 4.3% 0.1%	PAPI_TLI 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05	8.1% 8.1% 1.8% 0.2% 0.4%
Exp e	431 432 433 434 Scopes eriment Agg state 1 loop at st 1 loop at st 1 loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 393 ate_mod.f90: 407	n NOMK PAF 5.8 8.9 1.9 1.8 1.5 0.0	PL_TOT ▼ 38e10 35e09 15.2% 34e09 3.3% 34e09 3.1% 38e09 2.7% 31e08 1.7%	Anno 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08	otat T 13.7% 4.3% 0.4% 0.5% 2.8%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05	 4.3% 0.1%	PAPI_TLI 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05	8.1% 1.8% 0.2% 0.4% 0.8%
Exp • •	431 432 433 434 Scopes eriment Agg state 1 loop at st 1 loop at st 1 loop at st 1 loop at st 1 loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 393 ate_mod.f90: 407-409	n NOMK PAF 5.8 8.9 1.9 1.8 1.5 9.9	PL_TOT ▼ 38e10 35e09 15.2% 34e09 3.3% 34e09 3.1% 38e09 2.7% 31e08 1.7% 32e08 1.0%	Ann 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.05e09 4.20e02	otat T 13.7% 4.3% 0.4% 0.5% 2.8%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 1.97e05 6.05e06	 4.3% 0.1%	e View PAPI_TLI 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05	8.1% 1.8% 0.2% 0.4% 0.8%
Exp e o	431 432 433 434 Scopes eriment Agg state 1 loop at st 1 loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 393 ate_mod.f90: 407-409 ate_mod.f90: 364	n NOMK PAF 5.8 8.9 1.9 1.8 1.5 9.9 6.0	PL_TOT ▼ 38e10 35e09 15.2% 34e09 3.3% 34e09 3.1% 38e09 2.7% 31e08 1.7% 32e08 1.0%	Ann 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.82e08 4.20e08 4.16e02	otat T 13.7% 4.3% 0.4% 0.5% 2.8% 1.1%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 1.97e05 6.95e06	 4.3% 0.1%	e View PAPI_TLI 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05 3.60e05	8.1% 1.8% 0.2% 0.4% 0.8% 0.8%
Exp Co Co Co Co Co Co Co Co Co Co Co Co Co	431 432 433 434 Scopes eriment Agg state 100p at st 100p at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 393 ate_mod.f90: 407-409 ate_mod.f90: 364 ate_mod.f90: 367	n NOMK 5.8 8.9 1.9 1.8 1.5 9.9 6.0 5.2	PL_TOT ▼ 38e10 35e09 15.2% 34e09 3.3% 34e09 3.1% 38e09 2.7% 31e08 1.7% 32e08 1.0% 21e08 0.9%	Ann 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.05e09 4.20e08 4.16e08	Otat T 13.7% 4.3% 0.4% 0.5% 2.8% 1.1% 1.1%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 1.97e05 6.95e06 6.55e06	 4.3% 0.1% 0.1% 2.0%	e View <u>PAPI_TLI</u> 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05 3.60e05 2.95e05 2.95e05	8.1% 1.8% 0.2% 0.4% 0.8% 0.8% 0.8%
Exp 6 6 6 6 6 6	431 432 433 434 Scopes eriment Agg state 100p at st 100p at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 393 ate_mod.f90: 407-409 ate_mod.f90: 364 ate_mod.f90: 367 ate_mod.f90: 368	n NOMK 5.8 8.9 1.9 1.8 1.5 9.9 6.0 5.2 4.0	PL_TOT ▼ 38e10 35e09 15.2% 34e09 3.3% 34e09 3.1% 38e09 2.7% 31e08 1.7% 32e08 1.0% 21e08 0.9% 38e08 0.7%	Ann 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.05e09 4.20e08 4.16e08 4.29e08	Otat T 13.7% 4.3% 0.4% 0.4% 0.5% 2.8% 1.1% 1.1% 1.1%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 1.97e05 6.95e06 6.55e06 6.55e04	 4.3% 0.1% 0.1% 2.0% 0.0%	e View <u>PAPI_TLI</u> 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05 2.95e05 1.64e05 1.64e05	 8.1% 1.8% 0.2% 0.4% 0.8% 0.7% 0.4% 0.7% 0.4%
Exp 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	431 432 433 434 Scopes eriment Agg state ↑ loop at st ↑ loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 393 ate_mod.f90: 407-409 ate_mod.f90: 364 ate_mod.f90: 367 ate_mod.f90: 434	n NOMK 5.8 8.9 1.9 1.8 1.5 9.9 6.0 5.2 4.0 3.7	PI_TOT ▼ 38e10 35e09 15.2% 34e09 3.3% 34e09 3.1% 36e09 2.7% 31e08 1.7% 32e08 1.0% 21e08 0.9% 38e08 0.7% 37e08 0.6%	PAPI_TO 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.05e09 4.20e08 4.16e08 4.29e08 2.95e08	Otat T 13.7% 4.3% 0.4% 0.5% 2.8% 1.1% 1.1% 0.8%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 1.97e05 6.95e06 6.55e04 3.28e04	4.3% 0.1% 0.1% 2.0% 0.0%	e View <u>PAPI_TLI</u> 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05 2.95e05 1.64e05 2.62e05	 8.1% 1.8% 0.2% 0.4% 0.8% 0.7% 0.4% 0.6%
Exp 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	431 432 433 434 Scopes eriment Agg state ↑ loop at st ↑ loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 393 ate_mod.f90: 407-409 ate_mod.f90: 364 ate_mod.f90: 368 ate_mod.f90: 434	n NOMK 5.8 8.9 1.9 1.8 1.5 9.9 6.0 5.2 4.0 3.7 3.7	PI_TOT ▼ 38e10 95e09 15.2% 94e09 3.3% 34e09 3.1% 36e09 2.7% 91e08 1.7% 92e08 1.0% 21e08 0.9% 98e08 0.7% 77e08 0.6%	PAPI_TO 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.05e09 4.20e08 4.16e08 4.16e08 4.29e08 2.95e08 4.13e08	otat T 13.7% 4.3% 0.4% 0.5% 2.8% 1.1% 1.1% 0.8% 1.1%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 1.97e05 6.95e06 6.55e04 3.28e04	4.3% 0.1% 0.1% 2.0% 0.0%	e View <u>PAPI_TLI</u> 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05 2.95e05 1.64e05 2.62e05 1.97e05	 8.1% 1.8% 0.2% 0.4% 0.8% 0.7% 0.4% 0.6% 0.4%
Exp	431 432 433 434 Scopes eriment Agg state ↑ loop at st ↑ loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 431 ate_mod.f90: 407-409 ate_mod.f90: 364 ate_mod.f90: 367 ate_mod.f90: 368 ate_mod.f90: 434	n NOMK 5.8 8.9 1.9 1.8 1.5 9.9 6.0 5.2 4.0 3.7 3.7 1.6	PI_TOT ▼ 38e10 95e09 15.2% 94e09 3.3% 34e09 3.1% 38e09 2.7% 91e08 1.7% 92e08 1.0% 21e08 0.9% 98e08 0.7% 77e08 0.6% 73e08 0.6% 59e08 0.3%	PAPI_TO 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.05e09 4.20e08 4.16e08 4.29e08 2.95e08 4.13e08 1.77e08	otat T 13.7% 4.3% 0.4% 0.5% 2.8% 1.1% 1.1% 0.8% 1.1% 0.5%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 1.97e05 6.95e06 6.55e06 6.55e04 3.28e04 Metr	4.3% 0.1% 0.1% 2.0% 0.0% 0.0%	e View 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05 3.60e05 2.95e05 1.64e05 2.95e05 1.64e05 2.62e05 1.97e05 3.28e04	 B 8.1% 1.8% 0.2% 0.4% 0.8% 0.7% 0.4% 0.6% 0.4% 0.1%
Exp 0 0 0 0 0 0 0 0 0 0 0 0 0	431 432 433 434 Scopes eriment Agg state ↑ loop at st ↑ loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 431 ate_mod.f90: 407-409 ate_mod.f90: 364 ate_mod.f90: 367 ate_mod.f90: 368 ate_mod.f90: 434 avigation 77	n NOMK 5.8 8.9 1.9 1.8 1.5 9.9 6.0 5.2 4.0 3.7 3.7 1.6 3.8	PI_TOT ▼ 38e10 95e09 15.2% 94e09 3.3% 34e09 3.1% 38e09 2.7% 91e08 1.7% 92e08 1.0% 21e08 0.9% 98e08 0.7% 77e08 0.6% 73e08 0.6% 39e08 0.3% 81e07 0.1%	PAPI_TO 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.05e09 4.20e08 4.16e08 4.29e08 2.95e08 4.13e08 1.77e08 2.06e06	Otat T 13.7% 4.3% 0.4% 0.5% 2.8% 1.1% 1.1% 0.8% 1.1% 0.5% 0.0%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 1.97e05 6.95e06 6.55e04 3.28e04 Metr	4.3% 0.1% 0.1% 2.0% 0.0% 0.0% 0.0%	e View <u>PAPI_TLI</u> 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05 2.95e05 1.64e05 2.95e05 1.64e05 2.62e05 1.97e05 3.28e04	8.1% 1.8% 0.2% 0.4% 0.8% 0.7% 0.4% 0.6% 0.4% 0.1%
Exp 0 0 0 0 0 0 0 0 0 0 0 0 0	431 432 433 434 Scopes eriment Agg state ↑ loop at st ↑ loop at st	if (present(RHOOUT)) the RHOOUT = WORK1*DE regate Metrics ate_mod.f90: 426-429 ate_mod.f90: 431 ate_mod.f90: 431 ate_mod.f90: 407-409 ate_mod.f90: 367 ate_mod.f90: 367 ate_mod.f90: 368 ate_mod.f90: 434 avigation 77 00: 200	n NOMK 5.8 8.9 1.9 1.8 1.5 9.9 6.0 5.2 4.0 3.7 3.7 1.6 3.8 2.7	PI_TOT ▼ 38e10 55e09 15.2% 34e09 3.3% 34e09 3.1% 38e09 2.7% 91e08 1.7% 92e08 1.0% 21e08 0.9% 98e08 0.7% 77e08 0.6% 39e08 0.3% 31e07 0.1% 71e07 0.0%	PAPI_TO 3.79e10 5.20e09 1.64e09 1.55e08 1.82e08 1.05e09 4.20e08 4.16e08 4.29e08 2.95e08 4.13e08 1.77e08 2.06e06 7.74e05	Otat T 13.7% 4.3% 0.4% 0.5% 2.8% 1.1% 1.1% 0.8% 1.1% 0.5% 0.0%	ed So PAPI_L2 3.33e08 1.43e07 2.95e05 6.95e06 6.55e06 6.55e04 3.28e04 Metr	4.3% 0.1% 0.1% 2.0% 0.0% 0.0% 0.0%	e View PAPI_TLI 4.44e07 3.60e06 8.19e05 9.83e04 1.64e05 3.60e05 3.60e05 3.60e05 2.95e05 1.64e05 2.95e05 1.64e05 2.62e05 1.97e05 3.28e04 3.28e04	8.1% 1.8% 0.2% 0.4% 0.8% 0.7% 0.4% 0.6% 0.1%



HPCToolkit ASC Impact

- HPCToolkit installed and used on ASC systems at LANL
 - -past: Nirvana/Blue Mountain (MIPS+Irix)
 - -today: Q (Alpha+Tru64)
 - -emerging: Lightning (Opteron+Linux)
- Conducted performance tuning workshops at LANL —SAGE, FLAG, Truchas, MCNP, Blanca
- Used to pinpoint and tune ASC applications
 - -assisted in analysis and tuning SAGE (factor of 2 improvement)
 - —used to help gain additional factor of 2 on SAGE smvp performance
 - —used to help gain a factor of 26 in Blanca AMR setup model code
- Used to assess FLAG for ASC burn code review in 2003

Nov 2004: HPC community tutorial at SCO4



Performance Modeling

- LANL PAL modeling: scalable models of application performance
- The "missing link": detailed models of node performance
- Rice modeling: understand interplay between node program and microprocessor architecture
 - -measure application-specific factors
 - static analysis
 - dynamic analysis
 - -construct models of computation and memory hierarchy performance
 - instruction dependences
 - memory hierarchy miss rates and latencies
 - $-\operatorname{identify}$ impediments and enablers for high performance

Marin & Mellor-Crummey: SIGMETRICS '04



Analysis and Modeling Toolkit





17

Memory Behavior: Sweep3D



Predicted from optimized SPARC binaries!

LACS

Execution Behavior: Sweep3D



Predicted from optimized SPARC binaries!



Compiler Technology for Parallel Languages

- Today: Computing on fragmented address spaces with MPI
- Enormous burden on application developer
 - -Choose granularity of parallelism
 - -Partition application data structures and computation
 - -Add data movement and synchronization
 - -Manage storage for non-local data
 - -Developer responsible for all optimization of communication
 - latency tolerance: overlapping communication with computation
- Implications
 - -Granularity choices are hard-coded into program
 - —Hard to tailor for different architectures, e.g. vector vs. clusters



Higher-level Programming Models

Simplify programming of parallel systems

- Provide abstraction of global address space (GAS)
 avoid tedious management of partitioned address spaces
- Separate concerns

 $-algorithm\ specification\ vs.\ partitioning,\ mapping\ and\ synchronization$

- Support flexible mappings of data and computation to "nodes" — high level management of locality
- Avoid target-dependent optimization by programmers
- Make applications more malleable
- Enhance performance portability



Compilers for High-Level Parallel Programming

- Near term
 - —compiler technology for SPMD global address space languages results:
 - multiplatform Co-array Fortran compiler: (Alpha, Itanium, MIPS, Pentium) × (Quadrics, Myrinet, shared memory)
 - prototype compiler delivers performance comparable with MPI
 - -Previous work on OpenMP compiler and tools
- Longer term
 - -compiler technology for high-level data parallel languages, e.g. HPF
 - -results: push the envelope for data parallel languages
 - new data and computation partitionings,
 - compiler optimizations for communication and computation
 - deliver performance competitive with hand-coded MPI

Two best paper awards in International Conferences



HPF vs MPI Efficiency for NAS SP (162³ size)

Efficiency SP class 'C'





IMPACT-3D

HPF application: Simulate 3D Rayleigh-Taylor instabilities in plasma using TVD

- Problem size: 1024 x 1024 x 2048
- Compiled with HPF/ES compiler +45 HPF directives -7.3 TFLOPS on 2048 ES processors ~ 45% peak
- Compiled with dHPF on PSC's Lemieux (Alpha+Quadrics)

# procs	relative speedup	GFLOPS	% peak
128	1.0	46.4	18.1
256	1.94	89.9	17.6
512	3.78	175.5	17.4
1024	7.58	352.0	17.2



1334 lines

Open Source Compilers

Critical resources in scientific computing

- Big picture issues
 - -GCC is a 1980s design and is showing its age
 - -what will replace GCC?
 - -will that compiler produce good code for scientific applications?
- Open64/ORC is a strong candidate
 - -Well done suite of optimizations, including backend components
 - -Full-blown dependence analyzer
 - -Somewhat lacking in support for retargeting

Led SCO4 Workshop on Open Source Open64

- LLVM is another candidate
 - -Newer compiler with fewer implemented optimizations
 - -Good architecture; strong support for retargeting
 - -Support for runtime reoptimization



Improving Backend Optimization in LLVM

- Register Allocation
 - implemented two coloring allocators for LLVM, tested them across the range of supported architectures
 - -both improve code performance with respect to the old allocator
 - -we will distribute one or both of these allocators (legal issues)
 - —both allocators move across architectures with almost no changes
- Instruction Selection
 - —we intend to build an aggressive scheduler for LLVM
 - coupled with a new register allocator, should make LLVM competitive



Compiler Optimization Research

- High Level
 - Scalarization
 - Asymptotically optimal scalarization of F90 array assignment
 - Blocking
 - Automating blocking of LU and QR factorization
 - -Array contraction
- Low Level
 - -Removing redundant memory operations
 - Discover and remove redundant pointer-based memory operations (up to 40% of all loads; 16% on average)
 - -Object-to-object vectorizer for Pentium
 - Vectorize Pentium object code for SSE: 4x for some loops
 - -Fast techniques for copy coalescing
 - New technique for modeling interferences
 - Speeds up copy coalescing and live-range identification



Knowledge Transfer

- Visitors at LANL
 - -2 LANL Rice summer interns
 - -Extended visits (Fowler at LANL 6 weeks)
- Joint workshops with LANL
 - -2 performance tuning; 1 performance modeling
- Publications
 - topics: parallel compiler and runtime technology, improving memory hierarchy performance, algorithm implementation studies, performance analysis & tools, performance modeling, compiler algorithms
 - -3 best paper awards (IPDPS 01, IPDPS 02, PACT 02)
 - -14 journal articles
 - -52 conference and workshop papers
 - -3 technical reports
- 5 Rice PhD graduates (2 to national labs, 3 to academia)

