WRF performance on Intel® Processors

R. Dubtsov, A. Semenov, D. Shkurko
Intel Corp., pr. ak. Lavrentieva 6/1, Novosibirsk, Russia, 630090
{roman.s.dubtsov, alexander.l.semenov,dmitry.v.shkurko,}@intel.com

12-15 June 2007

Executive summary

- All existing WRF benchmarks have been ported to Intel platforms
- New configurations files for Intel platforms have been created for WRF 2.2 and WPS
- Intel tools (compilers, profilers, iMPI) have been successfully used to port and optimize WRF benchmarks
- Performance and scalability of WRF have been evaluated for wide spectrum of core numbers

Notations and abbreviations

- **Woodcrest processor** – Dual Core Intel® Xeon® Processor 5100 series
- **Clovertown processor** – Quad Core Intel® Xeon® Processor 5300 series
- **ppn** – processes per node (the number of cores used on a node)
- **MVAPICH** – an MPI implementation from the Ohio State University
- **OpenMPI** – an MPI implementation from Univ.-s Of Tennessee, Indiana, and Stuttgart, and Los Alamos National Laboratory

Workload overview

	Sim. time	Domain size	Input data size and format	Microphysics
CONUS 12km	48 hours	425x300	938M, NetCDF	WSM5
IVAN	2 days	425x300	673M, NetCDF	WMS3
CONUS 2.5km	9 hours	1501x1201	14G, RAW	WSM6

WRF Parallel Benchmarks Page,
http://www.mmm.ucar.edu/wrf/WG2/bench/wrf_benchmark_page.htm

System configurations

	CPUs	Memory	OS	Interconnect
Clovertown (32 nodes) 'Atlantis'	8x2.67ghz, 4Mb Clovertown	16Gb per node	RHEL4 (U3)	SDR Infiniband Mellanox HCAs
Woodcrest (256 nodes) 'Endeavor'	4x3.0ghz, 4Mb Woodcrest	8Gb per node	RHEL4 (U2)	
SGI Altix 4700 (256 cores) 'Alice'	1.6ghz Montecito	16Gb per memory node	SUSE10	NUMAlink

Intel® Fortran and C compilers v9.1 for Linux.
Intel® MPI 3.0, MVAPICH-0.9.x
Intel® MKL 9.1

Advantages of using Intel® Compilers

Efficiency:

Inherent ability to highly optimize codes for all Intel multi-core processors

Ease of Use:

Automatic optimization features make it easier to obtain highly optimized target code

OpenMP support simplifies threaded application development

Supports other Intel tools:

Code instrumentation for
- Cluster Tools and
- Threading Tools

Enhanced debug information for Intel debugger

Intel® compilers use:

- SSE/SSE2/SSE3/SSE4 instructions
- Advanced branch prediction
- Software pipelining for Intel® Itanium®
- Explicit speculation on Intel® Itanium®
- Advanced high-level optimizations
- Interprocedural optimizations
- Auto-vectorization for Intel 64
- OpenMP v2.5

Some useful Intel® Compiler Options

-O2/-O3 Turn optimizations for speed, including high-level optimization

-ip/ipo Enables single-/multi-file interprocedural optimizations

-x*X*/-mtune=… Tuning for particular processor
-xT Enables auto-vectorization with the use of specific Woodcrest instructions

-fno-alias, -safe-cray-ptr, –assume nodummy_aliases These allow for making different aliasing assumptions making more optimizations possible

-openmp Turns on OpenMP directives compilation into threaded code

-tcheck/-tcollect/-openmp_profile Turns on instrumentation for checking or profiling with other Intel tools like Intel Thread Checker, Intel Thread Profiler and Intel Trace Collector

Controlling FP calculations

High Level option: -fp-model <spec>

Accuracy increases →

fast=2 | fast=1 | precise/source | strict

← Speed increases

Lower Level options

Speeding up expensive operations

- **-ftz** denormals are treated as zero
- **-no-prec-{div,sqrt},-IPF_fp_relaxed** faster but less accurate division and/or sqrt

-fp-speculation fast|safe|off asks to speculate or not to speculate on FP operations

-IPF_fma use fused-multiply-add instructions on Intel® Itanium®

Lower Level options

-assume protect_parens forces handling of parenthesis according to Fortran standard

-fpe*N* Specifies the floating-point exception handling possibly impacting performance

Deprecated options

-mp,-mp1,-fltconsistency,-IPF-fltacc
Superseded by new options

WRF 2.2 on Intel platforms

WRF and **WPS** were ported to Intel64 (WDC/Generic) and IA64 (Montecito/Generic) architectures w/Intel 9.1 compiler

	w/o nesting	RSL	RSL_LITE
Serial	+		
OpenMP	+	+	
MPI		+	+
MPI+OpenMP		+	+

Supported MPI libraries:

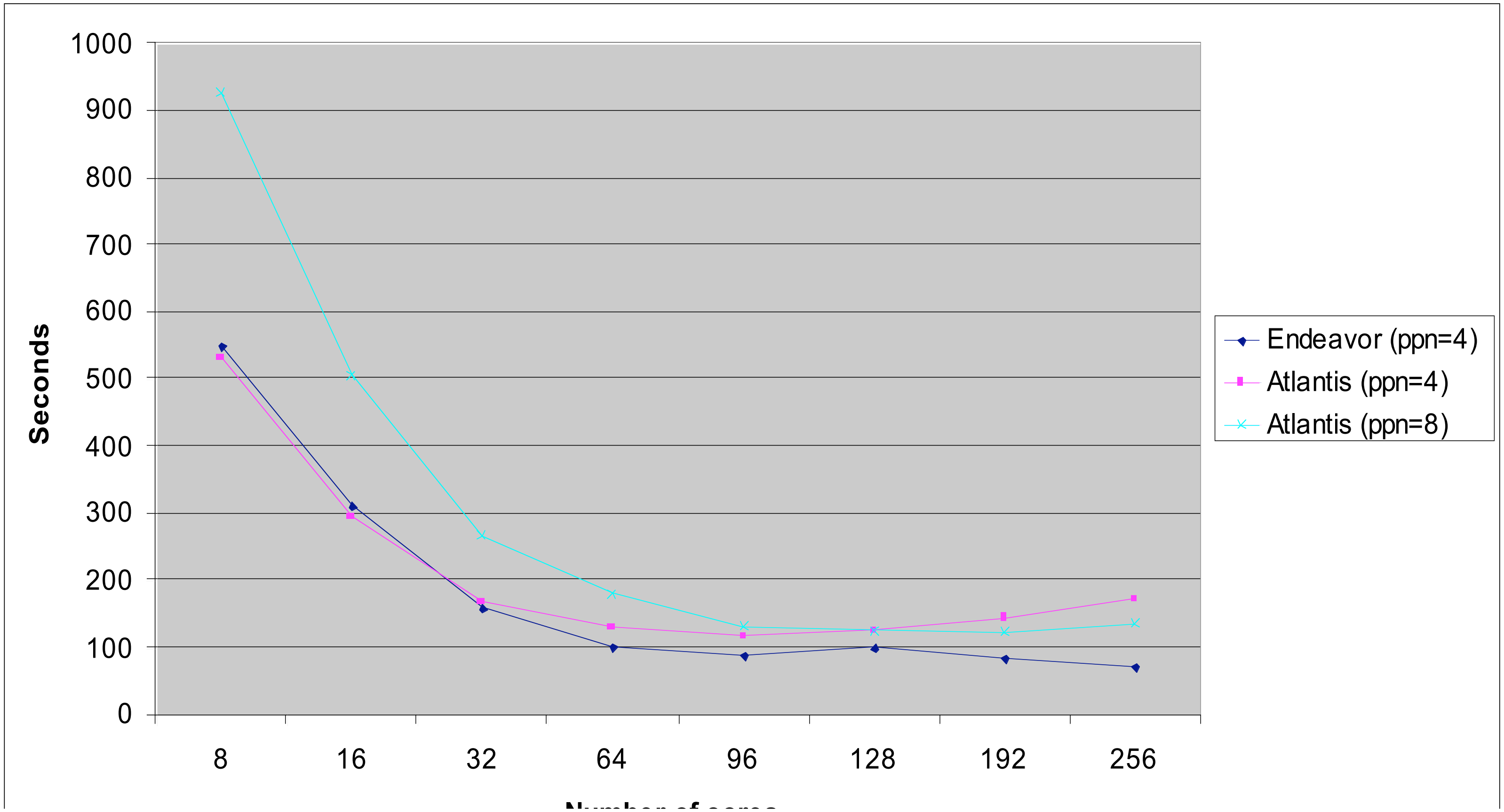
	IA64	Intel64
iMPI (2.0, 3.0)	+	+
MPICH-like	+	+
SGI MPT	+	

26 configurations have been created and verified

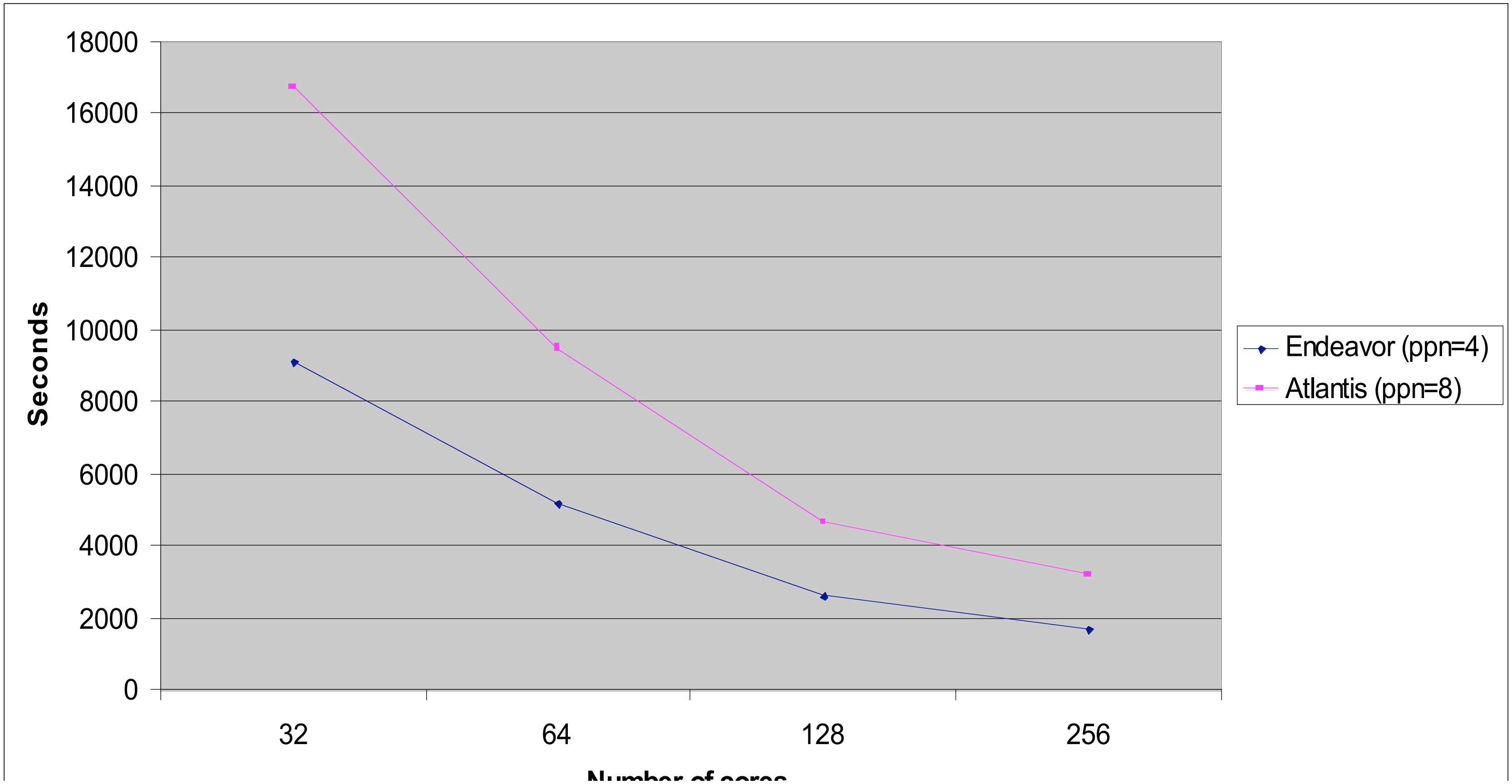
WRF 2.1.2 build configuration

- DM_PARALLEL with RSL_LITE
- MPI library: OpenMPI 1.2.1 / iMPI 3.0.043
- Compilers:
  - ifort 9.1.037
  - icc 9.1.043
- Compiler options: -O3 –xT –ip –no-prec-div –no-prec sqrt –fp model fast=2
- MKL 9.1

CONUS 12km

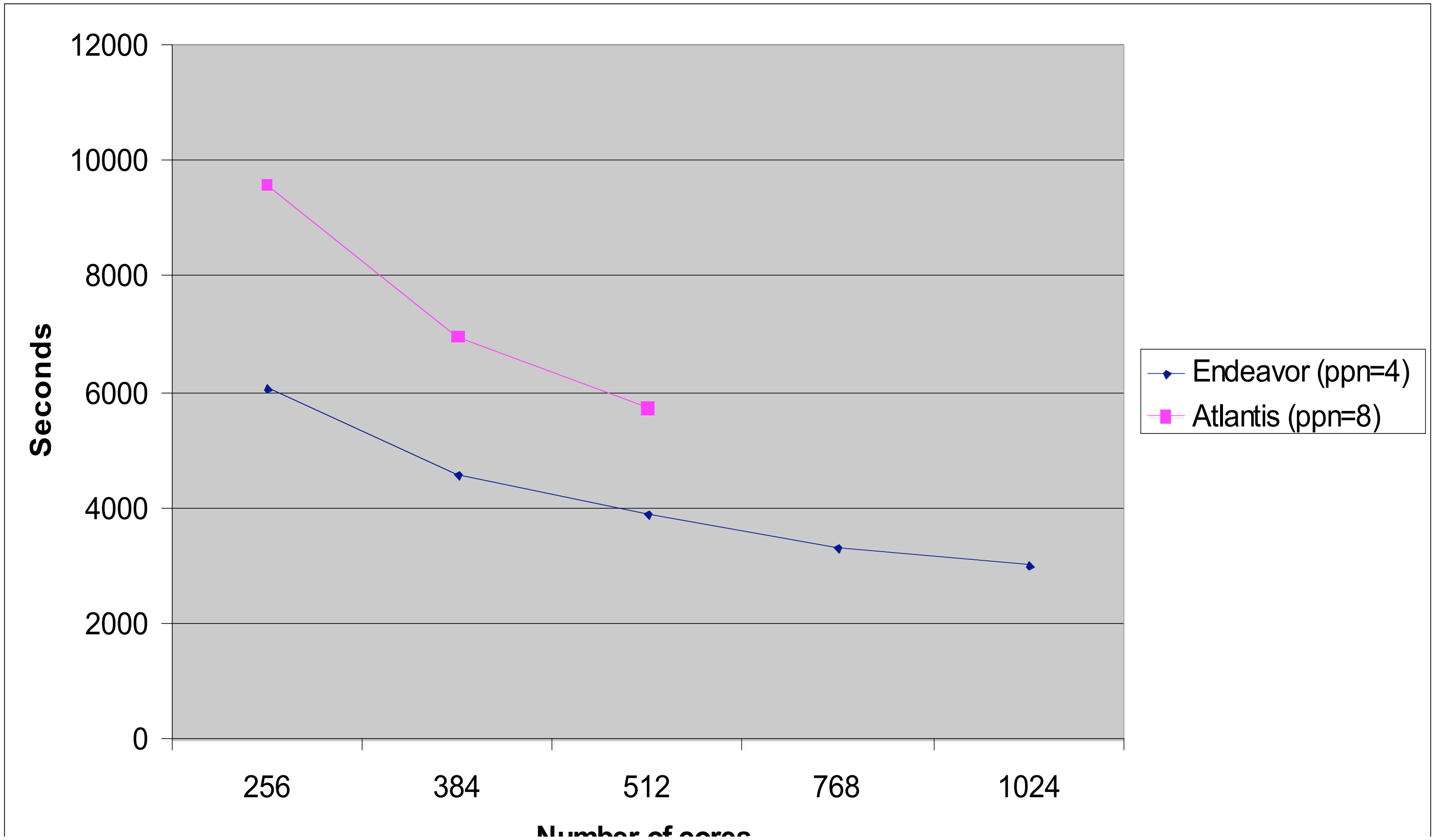

1000
900
800
700
600
500
400
300
200
100
0
Seconds
8
16
32
64
96
128
192
256
Endeavor (ppn=4)
Atlantis (ppn=4)
Atlantis (ppn=8)

IVAN



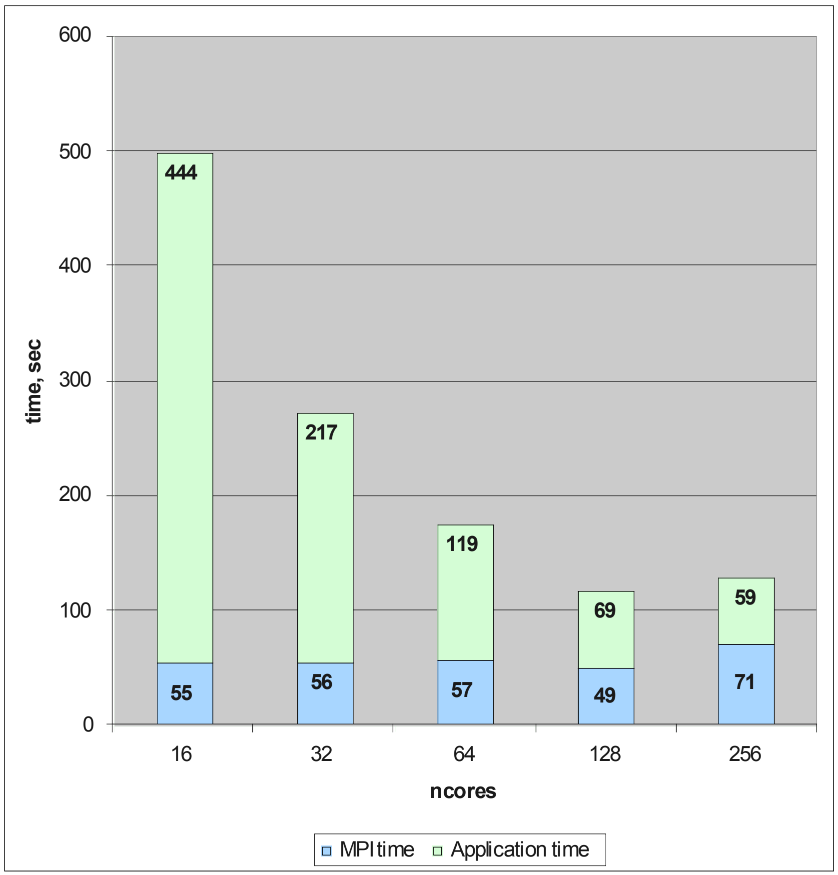
Seconds
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
32
64
128
256
Endeavor (ppn=4)
Atlantis (ppn=8)

CONUS 2.5km

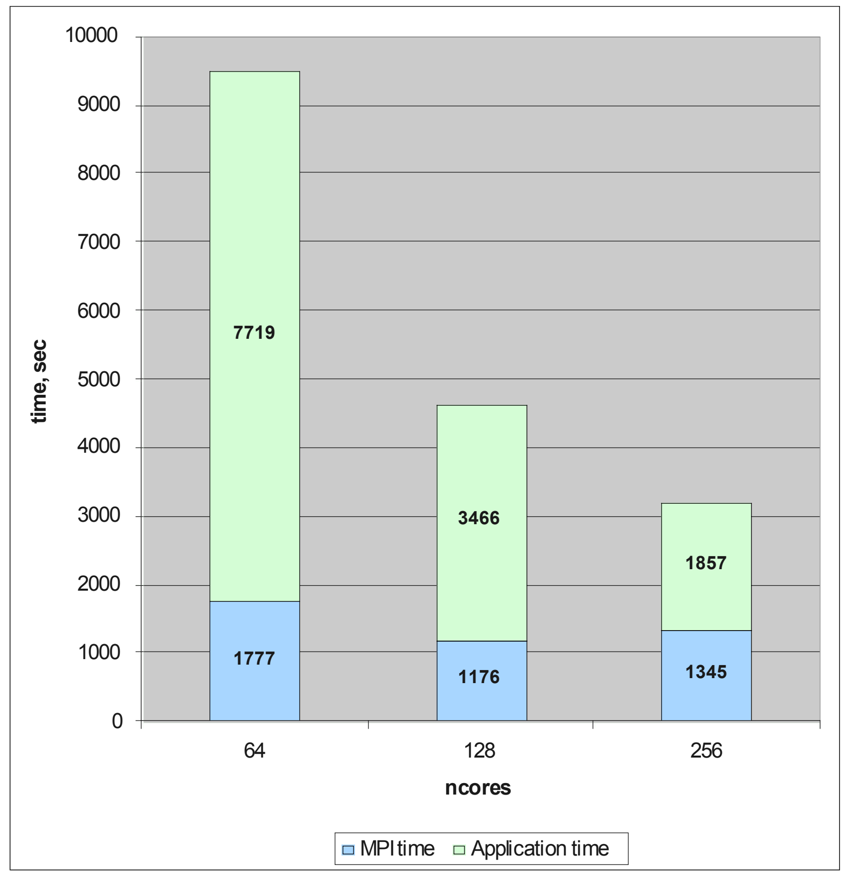

12000
10000
8000
6000
4000
2000
0
Seconds
256
384
512
768
1024
Endeavor (ppn=4)
Atlantis (ppn=8)

CONUS 12km/IVAN MPI Profiles
(Atlantis)

CONUS 12km



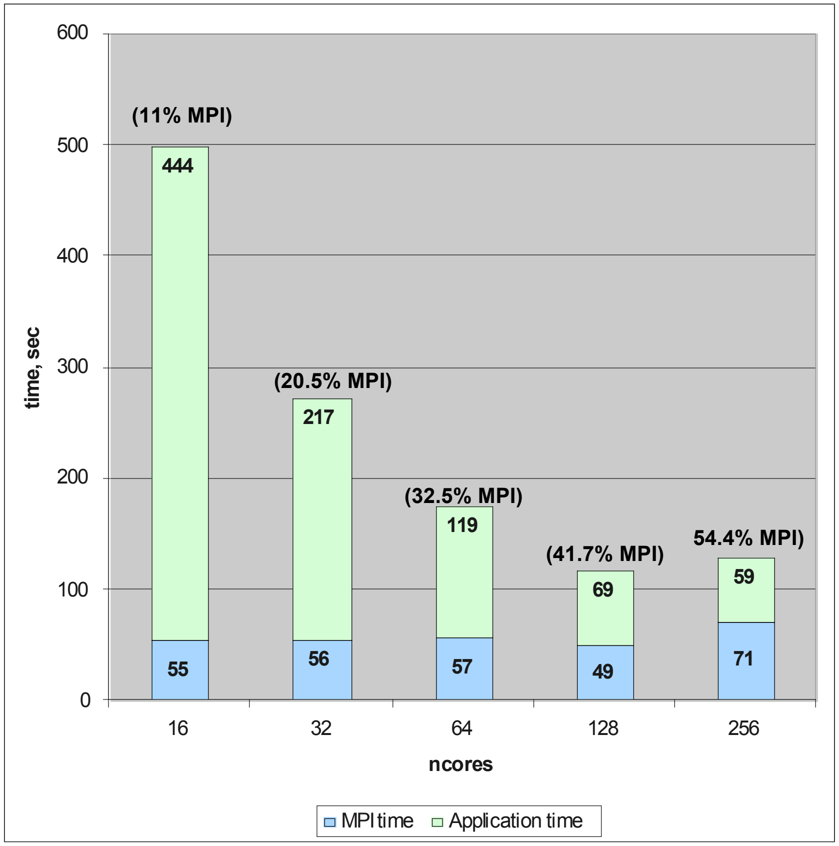
IVAN



CONUS 12km MPI Profiles

(Atlantis)

ppn = 8



ppn = 4