



Major Shared Resource Center
ERDC MSRC

DoD HPCMO Application Benchmarking and Profiling

Bill Ward
Computational Science and
Engineering Group
ERDC MSRC



Major Shared Resource Center
ERDC MSRC

TI-03 Application Benchmark Tests

Computational Science &
Engineering Group
ERDC MSRC



Major Shared Resource Center
ERDC
MSRC

Original Characterization of the DoD Workload

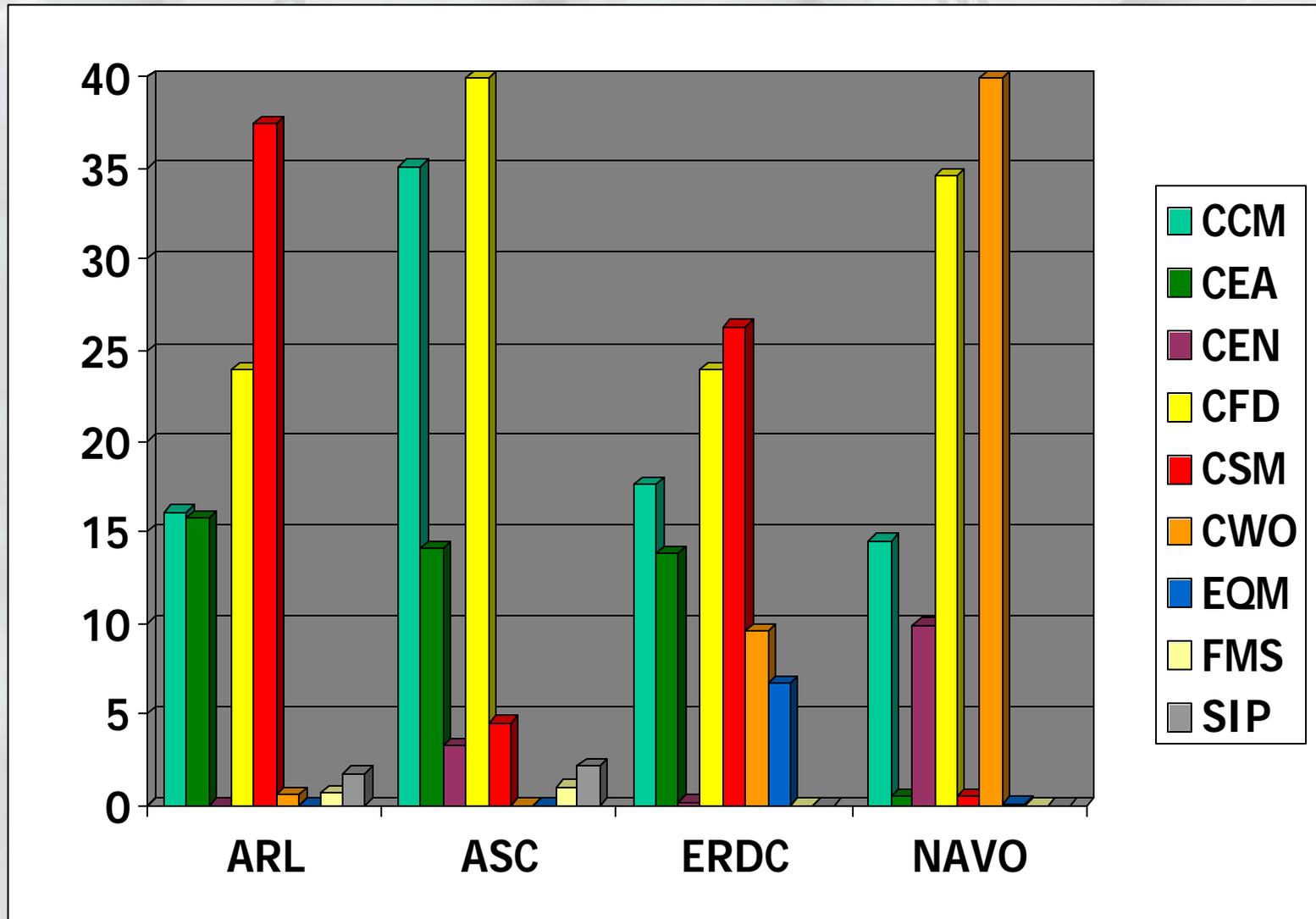
How to create a representative benchmark test package?

- Two stage survey: 32 responses
- Usage statistics across all MSRCs
- Combine the above information to make a list of possible applications to be used



Major Shared Resource Center
ERDC
MSRC

FY 2000 MSRC CTA Distribution





Major Shared Resource Center
ERDC

Selection of Codes

- Does the code use MPI or OpenMP?
- Does the code represent a CTA not already well represented in the test package?
- Does the code represent an MSRC not already well represented in the test package?
- Is the code readily obtainable?
- Is the code portable?



Major Shared Resource Center
ERDC VIGOR

Typical TI-xx Effort

- Coordinate with HPCMO On TI-XX's Benchmark Codes and Test Sets
- Generate a Complete Application Benchmark Package
- Respond to Vendor's Questions on Running the Benchmark Codes
- Evaluate the Vendor's Responses
- Benchmark Existing Government HPC Hardware for Comparison to the New Vendor Proposed Hardware



Structure of the Test Package

- Test package includes application tests (CS&E) and synthetic tests (Instrumental)
- Application tests include 6 codes
- 5 of the codes have a standard and a large test case (depending on the input)
- 1 code, Aero, is a serial vector code having only one test case
- Large test cases designed to run ~2 hours on 256 CPUs on NAVO POWER3
- Each test case has a baseline target for vendors to meet



Major Shared Resource Center
ERDC
VI
SRG

Benchmark Application Codes

Aero – Aeroelasticity CFD Code

(Fortran, Serial Vector, 15,000 Lines of Code)

Cobalt-60 – Turbulent Flow CFD Code

(Fortran, MPI, 19,000 Lines of Code)

CTH – Computational Structural Mechanics

(Fortran, MPI, 430,000 Lines of Code)

GAMESS – Molecular Dynamics Code

(Fortran, MPI, 330,000 Lines of Code)

LESlie3D – Large Eddy Simulation CFD Code

(Fortran, MPI, 7,000 Lines of Code)

NAMD – Molecular Dynamics Code

(C, MPI, 57,000 Line of Code)

NLOM – Ocean Circulation Modeling Code

(Fortran, MPI, 94,000 Lines of Code)



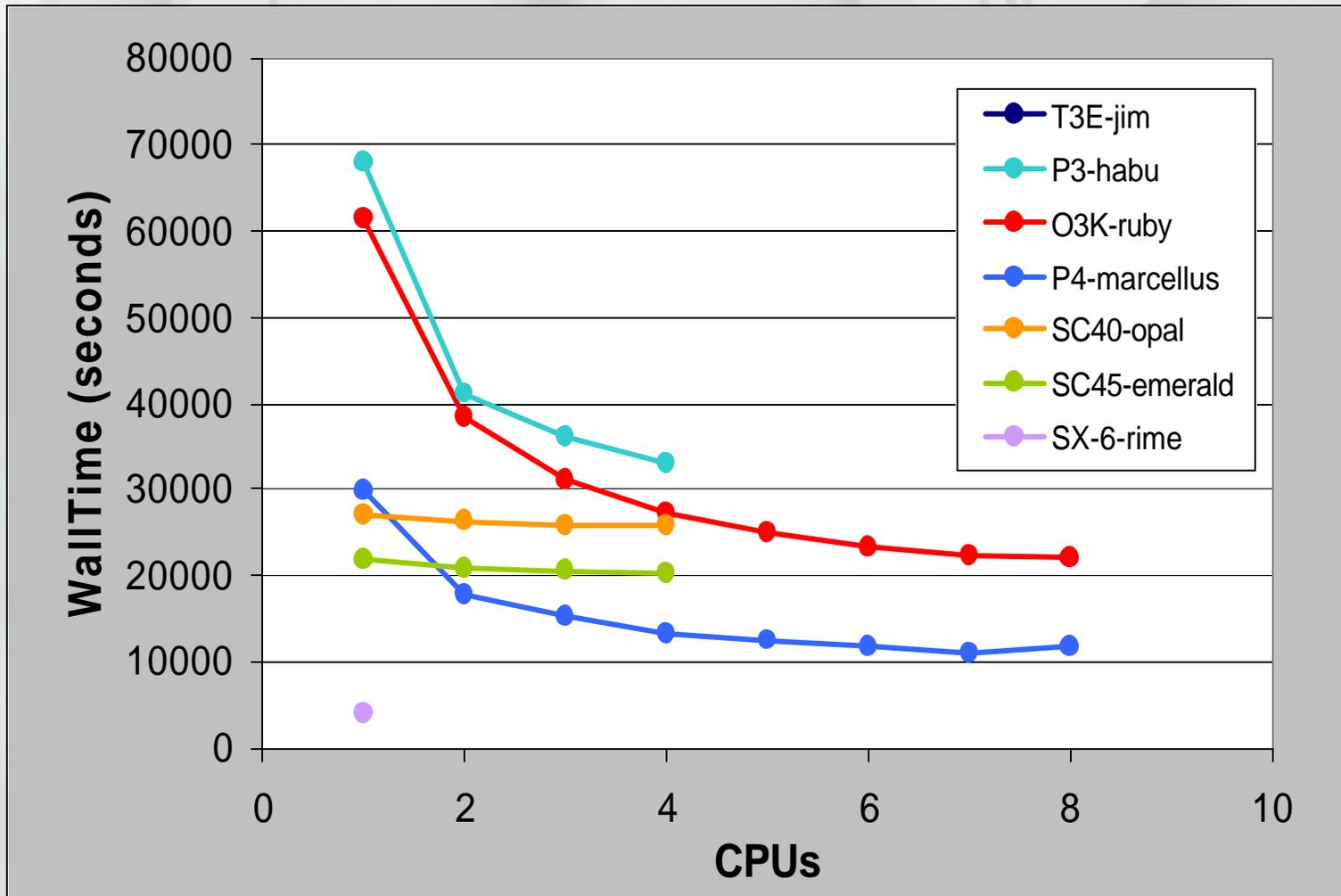
Tests Run on Govt. Systems

- Ranking (slowest to fastest)
 - T3E
 - 400 MHz O3K & 375 MHz POWER3 SP
 - 833 MHz Compaq SC
 - 1 GHz Compaq SC & 1.3 GHz POWER4 SP
- Some test cases don't scale well (problem too small), e.g., NAMD & NLOM



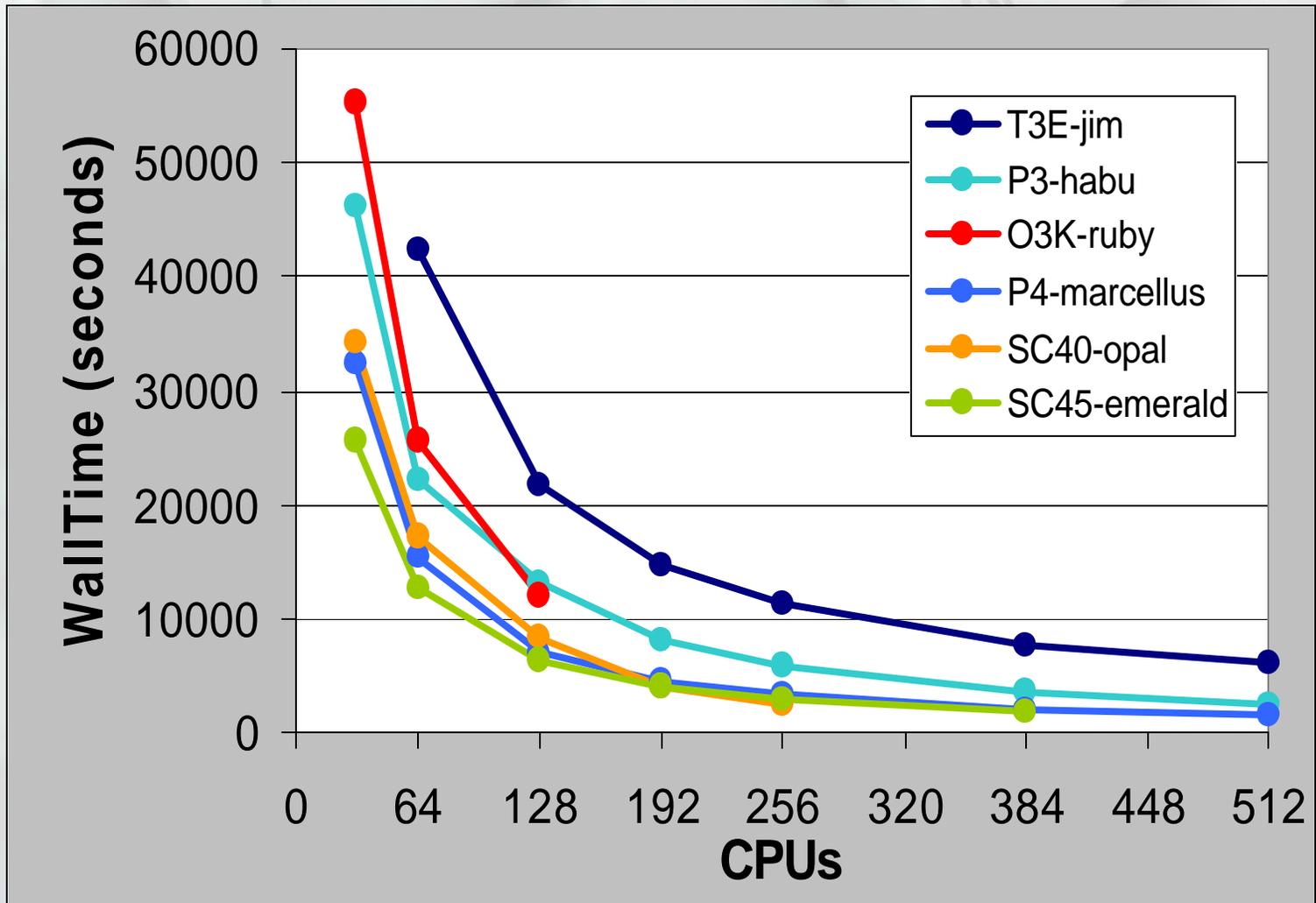
Major Shared Resource Center
ERDC VISRG

Aero Performance





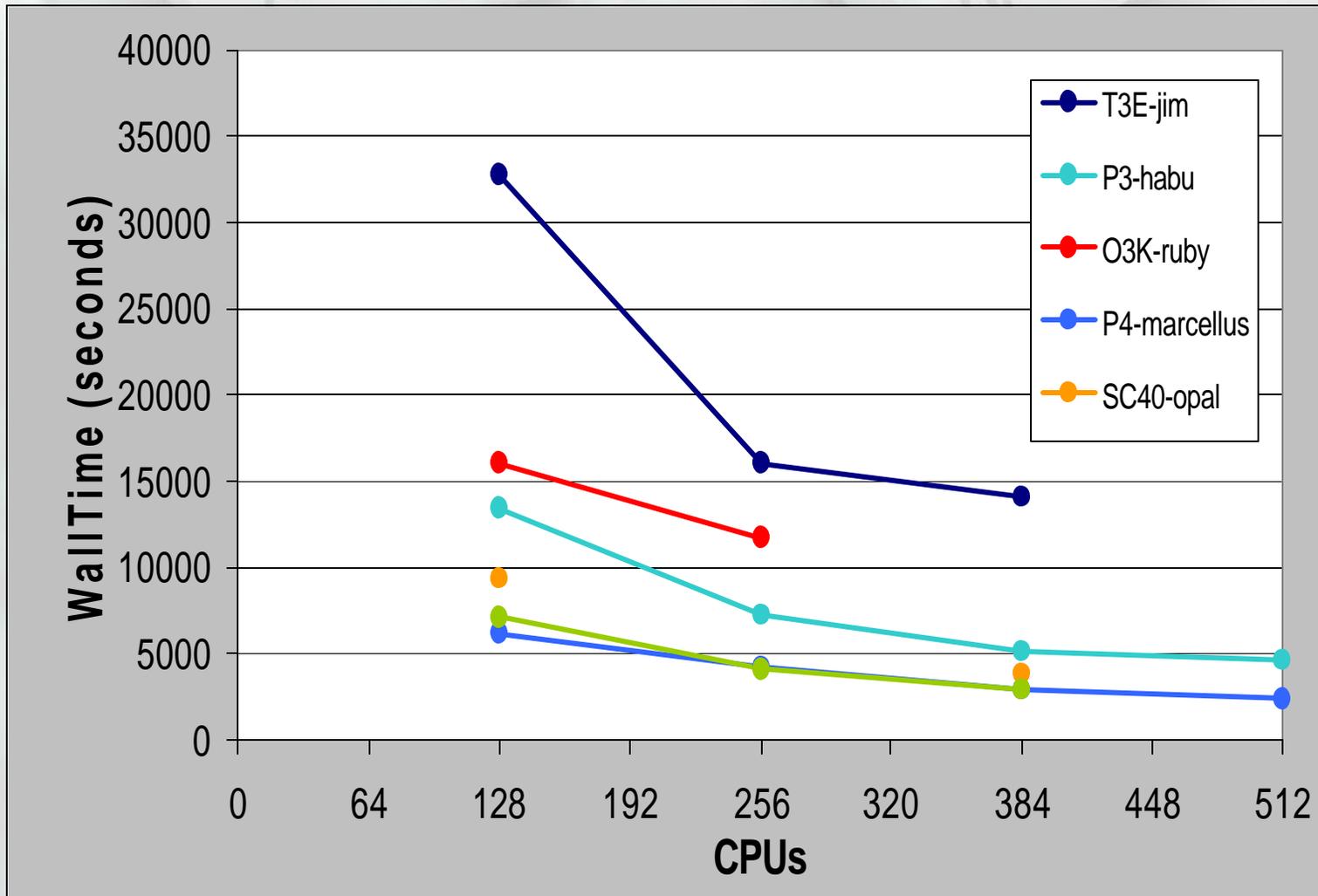
Cobalt-60 Performance





Major Shared Resource Center
ERDC
VISRG

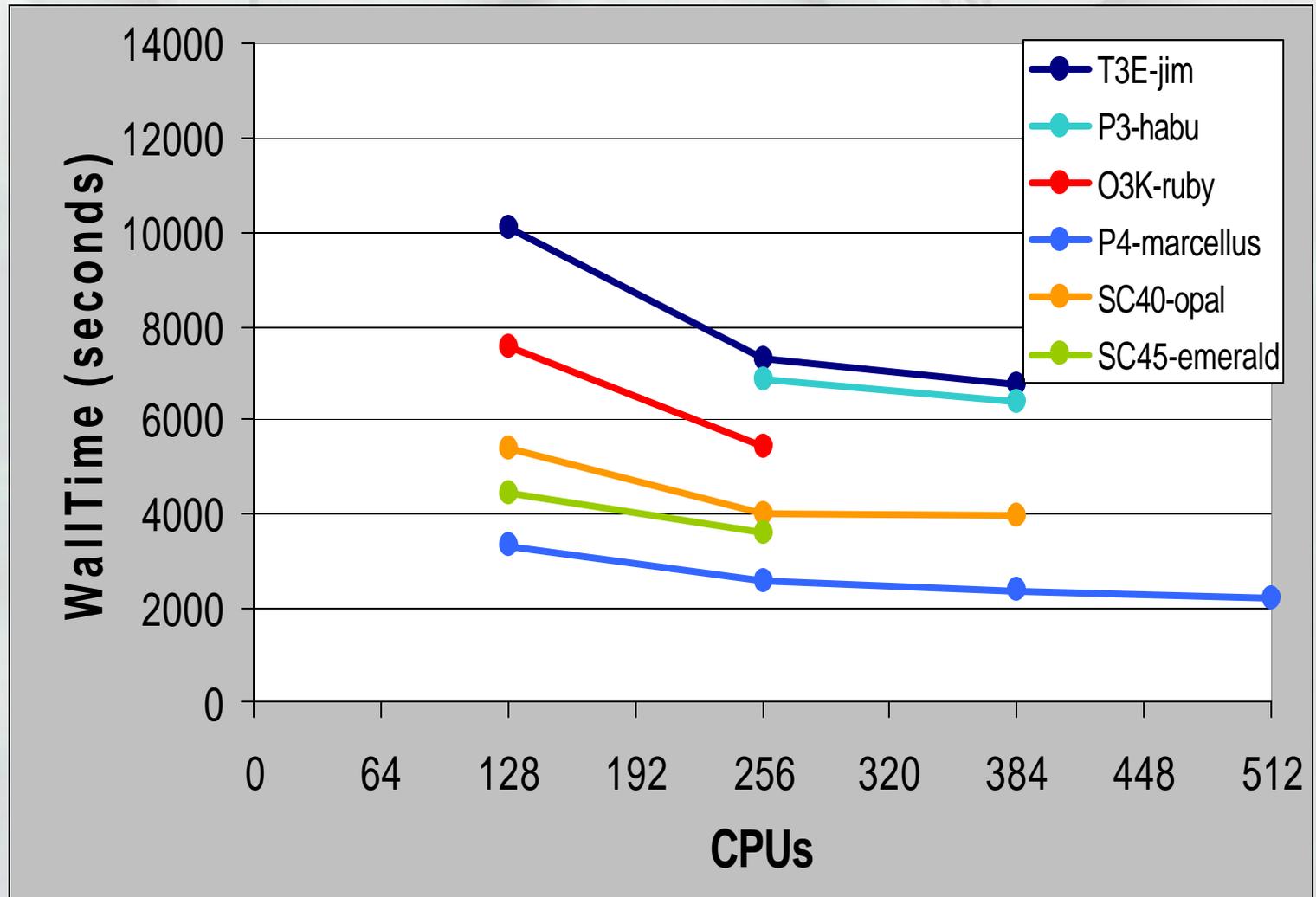
CTH Performance





Major Shared Resource Center
ERDC VISIRG

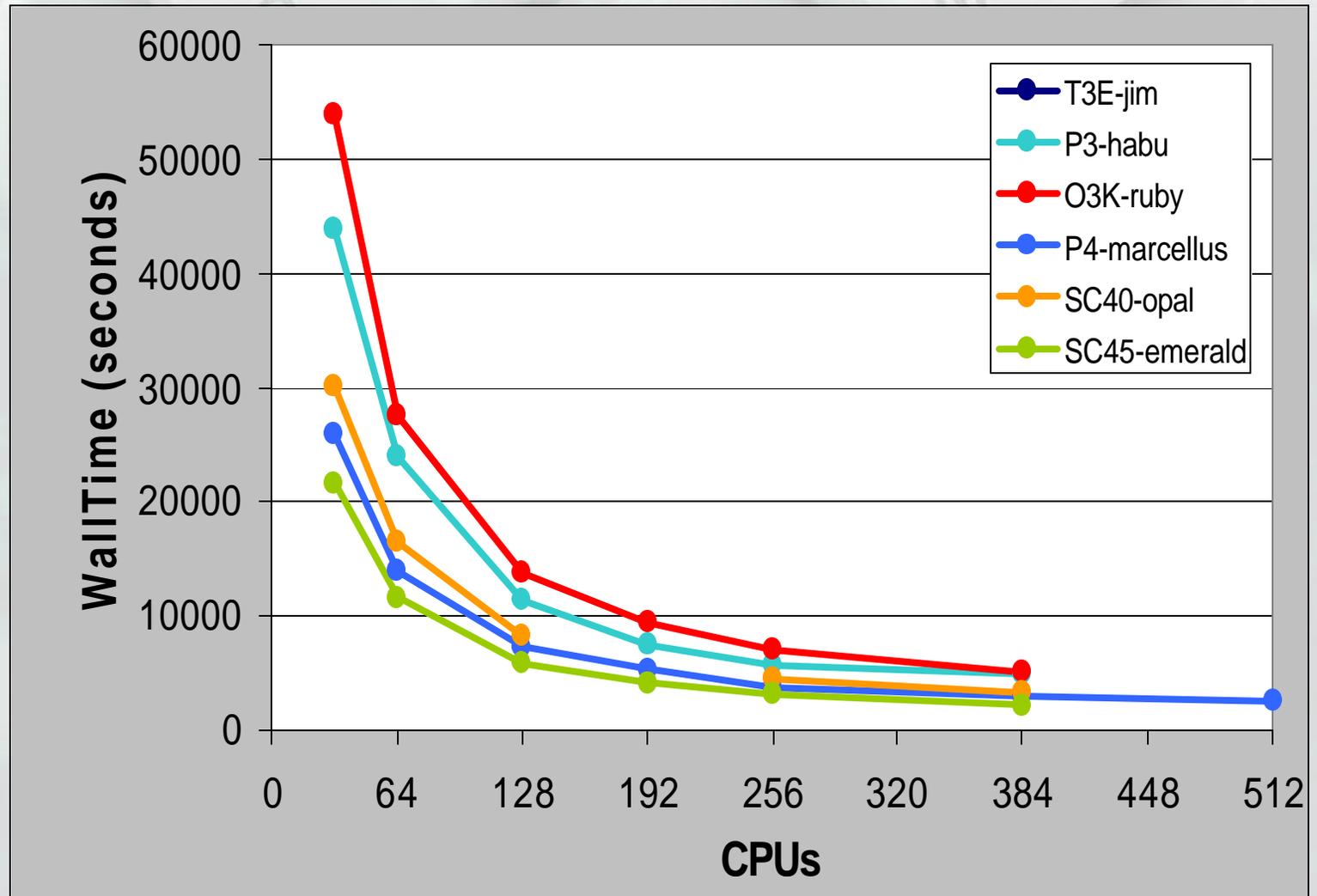
GAMESS Performance





Major Shared Resource Center
ERDC
VISRG

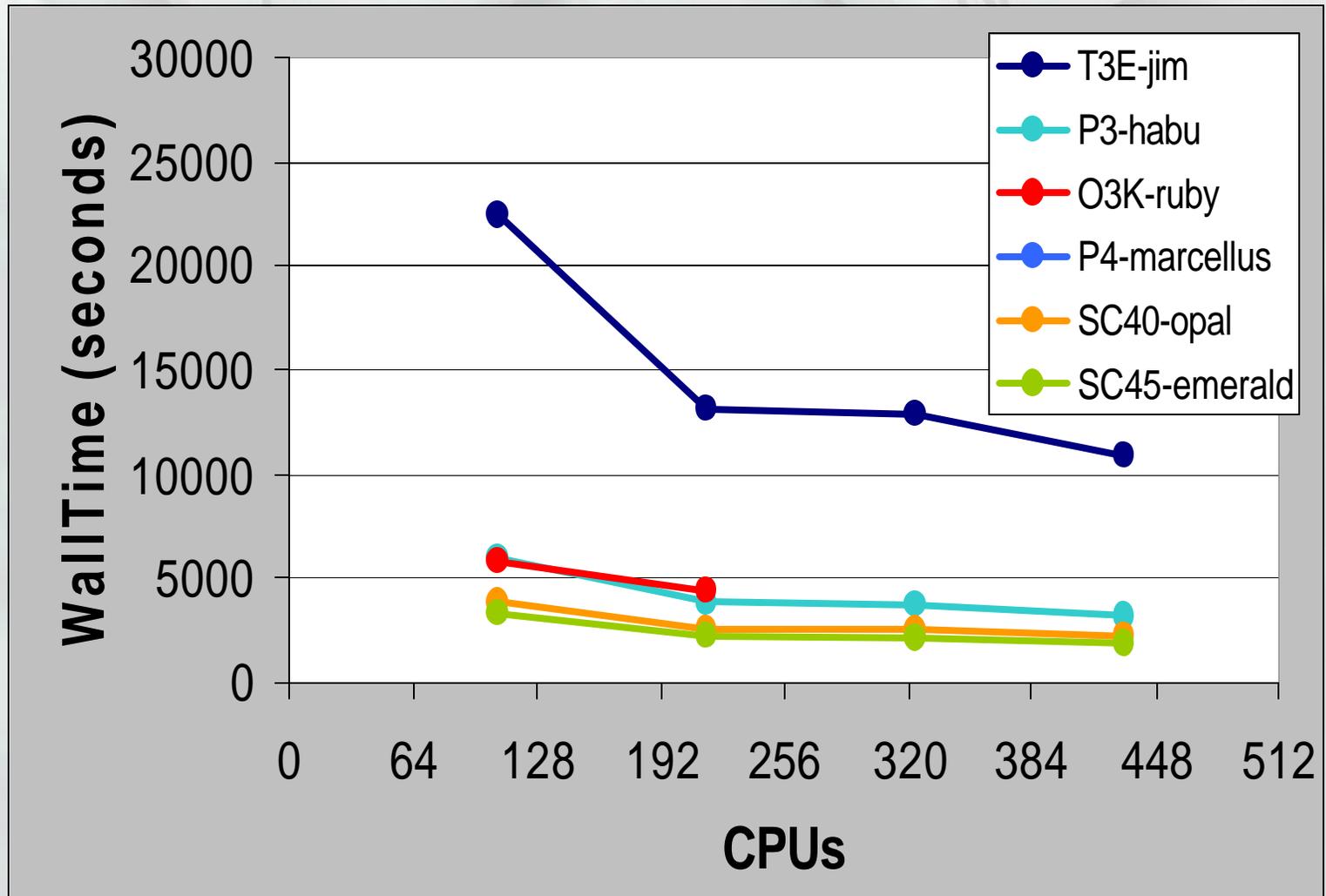
LESlie3D Performance





Major Shared Resource Center
ERDC VISION

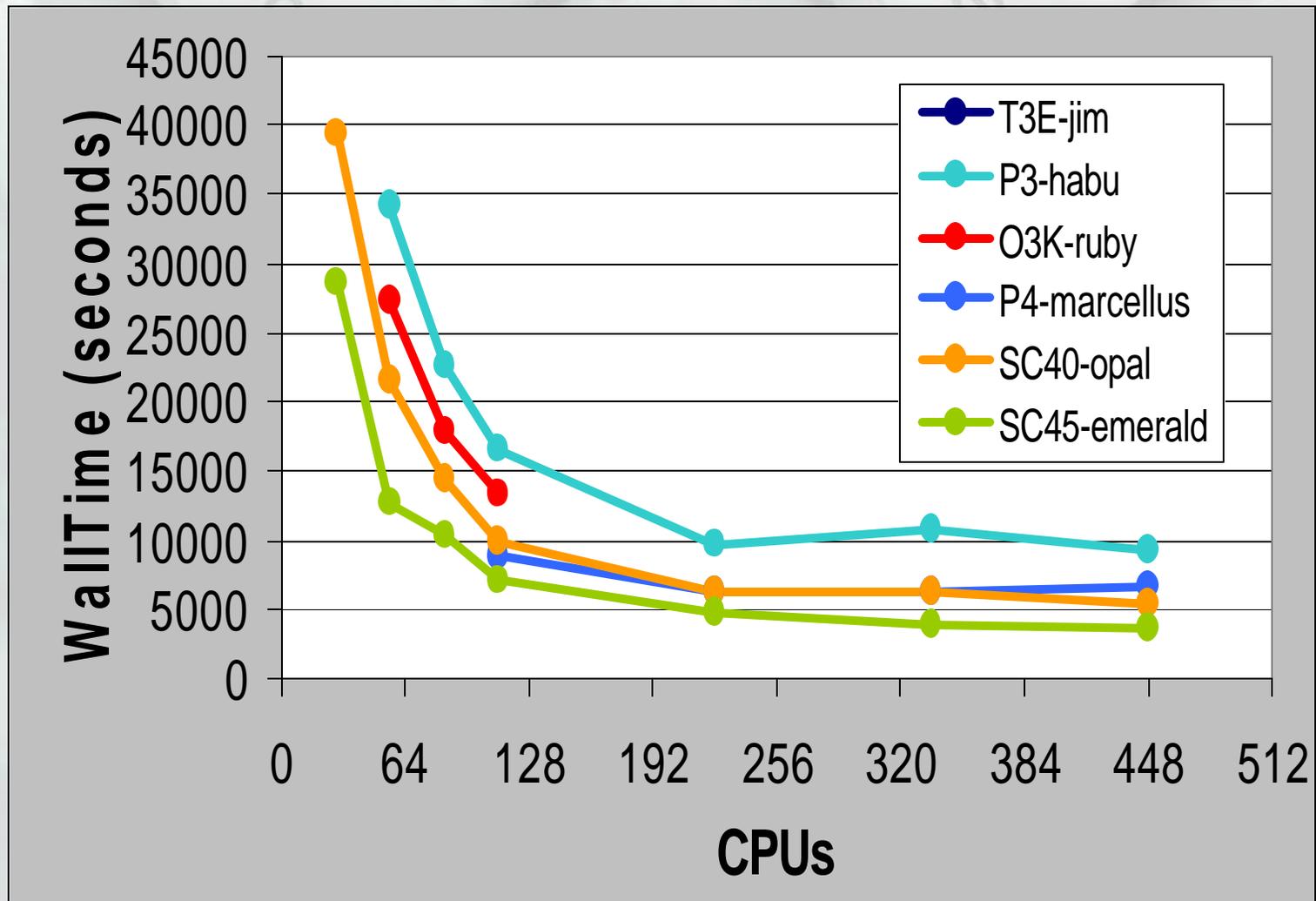
NAMD Performance





Major Shared Resource Center
ERDC VISOR

NLOM Performance



The Future

- New application codes
- Larger test sets (more CPUs)
- Synthetic applications
- Easier to run benchmark package
- Improved scoring methodology



Major Shared Resource Center
ERDC



Major Shared Resource Center
ERDC VISTAS

Recommended Profiling Tools for the HPCMO Benchmarking Effort

Shirley Moore, Henry Newman,
Allan Snavely, and Bill Ward

March, 2003

Purpose

- Describe types of profiling data to be collected
- Describe the available profiling tools
- Recommend a set of tools to be used to collect the desired data
- Present a plan for sharing the data-gathering load with application developers and users



Major Shared Resource Center
ERDC VISION



Goals of Gathering Profile Data

- Characterize application codes so as to be able to construct a representative set of benchmarks
- Provide relevant data for construction of synthetic benchmarks
- Optimize performance of benchmark codes
- Provide data to enable performance modeling and prediction



Data to Be Collected

- Three levels
 - Increasing levels involve more effort but allow for more detailed analysis.
- Gather data for each benchmark code for
 - at least two compiler optimization levels
 - three or four different CPU counts
 - two or more sets of input data
 - two or more different machine architectures



Level 1

- The following data for the entire run of the program
 - Wallclock time
 - Cycle count
 - Floating point operation counts by type
 - Memory operation counts by type
 - Branch counts
 - Cache (L1, L2, etc.) and TLB miss rates
 - Communication time broken down by MPI message type



Major Shared Resource Center
ERDC
VI
SRG

Level 1 (cont.)

- Easy to acquire
- Would give picture of workload with respect to broad characteristics but would not enable optimization and modeling



Level 2

- Same data as level 1
 - Level 2a: broken down by routine
 - Level 2b: broken down by basic block and loop
- Would provide a time-dependent profile of the application that would enable optimization and modeling to some degree
 - Could be used to construct “machine-dependent” performance models



Major Shared Resource Center
ERDC

Level 3

- Data required for “machine-independent” performance models
 - Memory access patterns
 - I/O profiles
 - Scaling profiles
 - Branch profiles
- Difficult to collect
- Collection will slow down execution significantly.



Available Profiling Tools

- Prof, gprof
- SGI perfex, ssrun
- Vprof
- PAPI
- Dynaprof
- HPM Toolkit
- PerfBench
- TAU
- Vampir
- SvPablo
- IBM Trace Libraries
- SIGMA
- MAPS
- MetaSim
- Dimemas
- MPIDtrace



Major Shared Resource Center
ERDC
VI
SRG

Tool Summaries

- See report for tables summarizing
 - General tool capabilities, ease of use, and status
 - Specific tool capabilities with respect to Level 1, 2, and 3 data



Recommendation 1

- Make tools for gathering Level 1 data available now (use tools that are already in use or already installed)
- Provide online form for developers/users to provide Level 1 data along with online help for using tools
- Reflect back report characterizing their code and making broad recommendations for machine(s) and tuning



Major Shared Resource Center
ERDC VISTAS

Recommendation 2

- Use existing tools to gather some Level 2 data
- Extend TAU to automatically gather all Level 2 information
- In return for Level 2 data, construct machine-dependent performance models of applications



Recommendation 3

- Have members of the benchmarking team use MAPS, MetaSim, and Dimemas to gather Level 3 data for selected codes.
- Use Level 3 data to construct detailed machine-independent performance models of these codes.



Recommendation 4

- Use access-controlled RIB (Repository in a Box) repository for
 - Benchmark codes
 - Machine information
 - Tool information
 - Schema for data to be collected
 - Case studies of how data benefited applications
 - Enter data manually from RIB maintainer interface or automatically from tools output via RIBAPI



Major Shared Resource Center
ERDC

Recommendation 5

- Create a “benchmark database”
 - Contains walltimes for all test cases
 - Contains metadata for all test cases (e.g., date, system, compiler options, software versions)
 - Contains profiling information



Major Shared Resource Center
ERDC

POWER3 Events (8 counters)

PM_0INST_CMPL
PM_0INST_DISP
PM_1MISS
PM_1WT_THRU_BUF_USED
PM_2CASTOUT_BF
PM_2MISS
PM_2WT_THRU_BUF_USED
PM_3CASTOUT_BF
PM_3MISS
PM_4CASTOUT_BUF
PM_4MISS
PM_6XX_RTRY_CHNG_TRTP
PM_6XXBUS_CMPL_LOAD
PM_ALIGN_INT
PM_BIU_ARI_RTRY
PM_BIU_LD_NORTRY
PM_BIU_LD_RTRY
PM_BIU_RETRY_DU_LOST_
RES
PM_BIU_ST_NORTRY
PM_BIU_ST_RTRY
PM_BIU_WT_ST_BF
PM_BR_CMPL
PM_BR_DISP
PM_BR_PRED
PM_BRQ_FULL_CYC
PM_BRU_IDLE

PM_BTAC_HITS
PM_BTAC_MISS
PM_BTC_BTL_BLK
PM_BURSTRD_L2ACC
PM_BURSTRD_L2MISS
PM_BURSTRD_MISS_L2_INT
PM_CBR_DISP
PM_CBR_RESOLV_DISP
PM_CHAIN_1_TO_8
PM_CHAIN_2_TO_1
PM_CHAIN_3_TO_2
PM_CHAIN_4_TO_3
PM_CHAIN_5_TO_4
PM_CHAIN_6_TO_5
PM_CHAIN_7_TO_6
PM_CHAIN_8_TO_7
PM_CI_ST_WT_CI_ST
PM_CMPLU_WT_LD
PM_CMPLU_WT_ST
PM_CORE_ST_N_COPYBACK
PM_CRB_BUSY_ENT
PM_CRLU_PROD_RES
PM_CYC
PM_CYC_1STBUF_OCCP
PM_DC_ALIAS_HIT
PM_DC_HIT_UNDER_MISS



Major Shared Resource Center
ERDC

More POWER3 Events

PM_DC_PREF_BF_INV
PM_DC_PREF_BLOCK_DEMAND_MISS
PM_DC_PREF_FILT_1STR
PM_DC_PREF_FILT_2STR
PM_DC_PREF_FILT_3STR
PM_DC_PREF_FILT_4STR
PM_DC_PREF_HIT
PM_DC_PREF_L2_INV
PM_DC_PREF_L2HIT
PM_DC_PREF_STREAM_ALLOC_BLK
PM_DC_PREF_USED
PM_DC_REQ_HIT_PREF_BUF
PM_DEM_FETCH_WT_PREF
PM_DISP_BF_EMPTY
PM_DSLB_MISS
PM_DU_ECAM_RCAM_OFFSET_HIT
PM_DU0_REQ_ST_ADDR_XTION
PM_DU1_REQ_ST_ADDR_XTION
PM_EE_OFF
PM_EE_OFF_EXT_INT
PM_EIEIO_WT_ST
PM_ENTRY_CMPLBF
PM_FETCH_CORR_AT_DISPATCH
PM_FPU_DENORM
PM_FPU_FADD_FMUL
PM_FPU_FCMP

PM_FPU_FDIV
PM_FPU_FEST
PM_FPU_FMA
PM_FPU_FPSCR
PM_FPU_FRSP_FCONV
PM_FPU_FSQRT
PM_FPU_IDLE
PM_FPU_IQ_FULL
PM_FPU_LD
PM_FPU_LD_ST_ISSUES
PM_FPU_SUCCESS_OOO_INST_SCHED
PM_FPU0_BUSY
PM_FPU0_CMPL
PM_FPU0_DENORM
PM_FPU0_FADD_FCMP_FMUL
PM_FPU0_FDIV
PM_FPU0_FEST
PM_FPU0_FMA
PM_FPU0_FMOV_FEST
PM_FPU0_FPSCR
PM_FPU0_FRSP_FCONV
PM_FPU0_FSQRT
PM_FPU1_BUSY
PM_FPU1_CMPL
PM_FPU1_DENORM



More POWER3 Events

PM_FPU1_IDLE
PM_FXU_IDLE
PM_FXU0_PROD_RESULT
PM_FXU1_IDLE
PM_FXU1_PROD_RESULT
PM_FXU2_BUSY
PM_FXU2_IDLE
PM_FXU2_PROD_RESULT
PM_GLOBAL_CANCEL_INST_DEL
PM_I_1_ST_TO_BUS
PM_IBUF_EMPTY
PM_IC_HIT
PM_IC_MISS
PM_IC_MISS_USED
PM_IC_PREF_USED
PM_INST_CMPL
PM_INST_DISP
PM_INTLEAVE_CONFL_STALLS
PM_IO_INTERPT
PM_L2ACC_BY_RWITM
PM_LD_CI
PM_LD_CMPL
PM_LD_CMPLBF_AT_GC
PM_LD_DISP
PM_LD_MISS_EXCEED_L2

PM_LD_MISS_EXCEED_NO_L2
PM_LD_MISS_L1
PM_LD_MISS_L2HIT
PM_LD_NEXT
PM_LD_WT_ADDR_CONF
PM_LD_WT_ST_CONF
PM_LINK_STACK_FULL
PM_LNK_REG_STACK_ERR
PM_LQ_FULL
PM_LSU_IDLE
PM_LSU_WT_SNOOP_BUSY
PM_LSU0_ISS_TAG_LD
PM_LSU0_ISS_TAG_ST
PM_LSU0_LD_DATA
PM_LSU1_IDLE
PM_LSU1_ISS_TAG_LD
PM_LSU1_ISS_TAG_ST
PM_MPRED_BR_CAUSED_GC
PM_PREF_MATCH_DEM_MISS
PM_RESRV_CMPL
PM_RESRV_RQ
PM_RETRY_BUS_OP
PM_SC_INST
PM_SNOOP
PM_SNOOP_E_TO_S



More POWER3 Events

PM_SNOOP_L1_M_TO_E_OR_S
PM_SNOOP_L2_E_OR_S_TO_I
PM_SNOOP_L2_M_TO_E_OR_S
PM_SNOOP_L2_M_TO_I
PM_SNOOP_L2ACC
PM_SNOOP_L2HIT
PM_SNOOP_PUSH_BUF
PM_SNOOP_PUSH_INT
PM_ST_CI_PREGATH
PM_ST_CMPL
PM_ST_CMPLBF_AT_GC
PM_ST_COND_FAIL
PM_ST_DISP
PM_ST_GATH_BYTES
PM_ST_GATH_DW
PM_ST_GATH_HW
PM_ST_GATH_WORD
PM_ST_HIT_L1
PM_ST_MISS_EXCEED_L2
PM_ST_MISS_EXCEED_NO_L2
PM_ST_MISS_L1
PM_ST_MISS_L2
PM_ST_MISS_L2_INT

PM_STQ_FULL
PM_SYNC
PM_SYNC_CMPLBF_CYC
PM_SYNC_RERUN
PM_SYNCHRO_INST
PM_TAG_BURSTRD_L2ACC
PM_TAG_BURSTRD_L2MISS
PM_TAG_BURSTRD_MISS_L2_INT
PM_TAG_LD_DATA_RECV
PM_TAG_ST_CMPL
PM_TAG_ST_L2ACC
PM_TAG_ST_MISS_L2
PM_TAG_ST_MISS_L2_INT
PM_TB_BIT_TRANS
PM_TLB_MISS
PM_TLBSYNC_CMPLBF_CYC
PM_TLBSYNC_RERUN
PM_UNALIGNED_LD
PM_UNALIGNED_ST
PM_W_1_ST



MIPS R12K Events (2 counters)

Level 1 data cache misses (1)
Level 1 instruction cache misses (0)
Level 2 data cache misses (1)
Level 2 instruction cache misses (0)
Requests for exclusive access to shared cache (1)
Requests for cache line invalidation (0)
Requests for cache line intervention (0)
Total translation lookaside buffer misses (1)
Data prefetch cache misses (1)
Failed store conditional instructions (0)
Total store conditional instructions (1)
Conditional branch instructions (0)
Conditional branch instructions mispredicted (1)
Instructions issued (0)
Instructions completed (0)
Floating point instructions (1)
Load instructions (1)
Store instructions (1)
Total cycles (0)



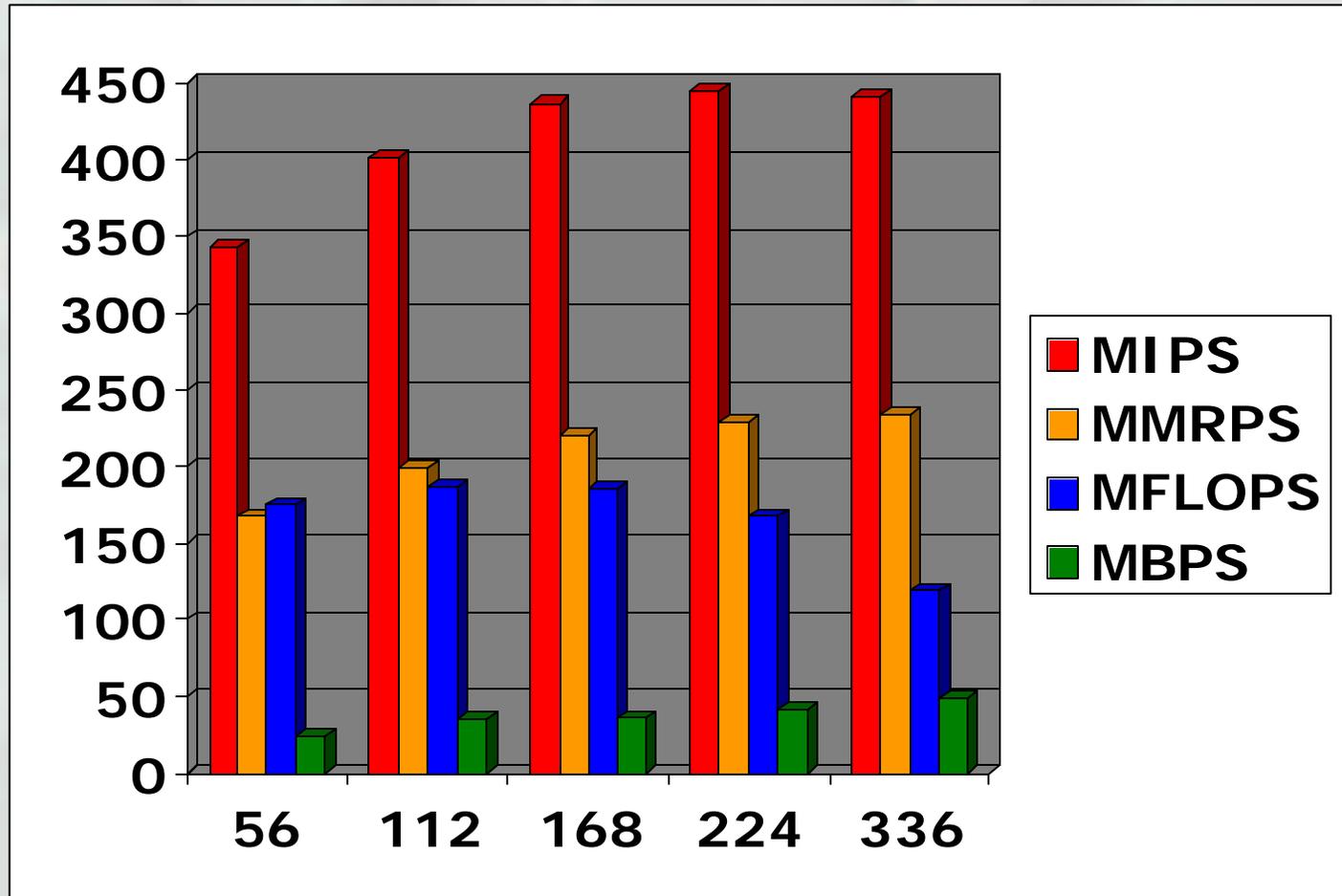
POWER3 Events Used to Profile NLOM

PM_EXEC_FMA	PM_FPU_EXE_FCMP
PM_FPU_EXEC_FPSCR	PM_FPU_FADD_FMUL
PM_FPU_FDIV	PM_FPU0_CMPL
PM_FPU1_CMPL	PM_INST_CMPL
PM_BR_CMPL	PM_BR_DISP
PM_CYC	PM_IC_HIT
PM_IC_MISS	PM_FPU_FDIV
PM_INST_DISP	PM_FRSP_FCONV_EXEC
PM_INST_CMPL	PM_LD_CMPL
PM_ST_CMPL	PM_ST_DISP
PM_ST_L1MISS	PM_FPU_LD
PM_LD_DISP	PM_LD_MISS_L1
PM_TLB_MISS	



Major Shared Resource Center
ERDC VISRC

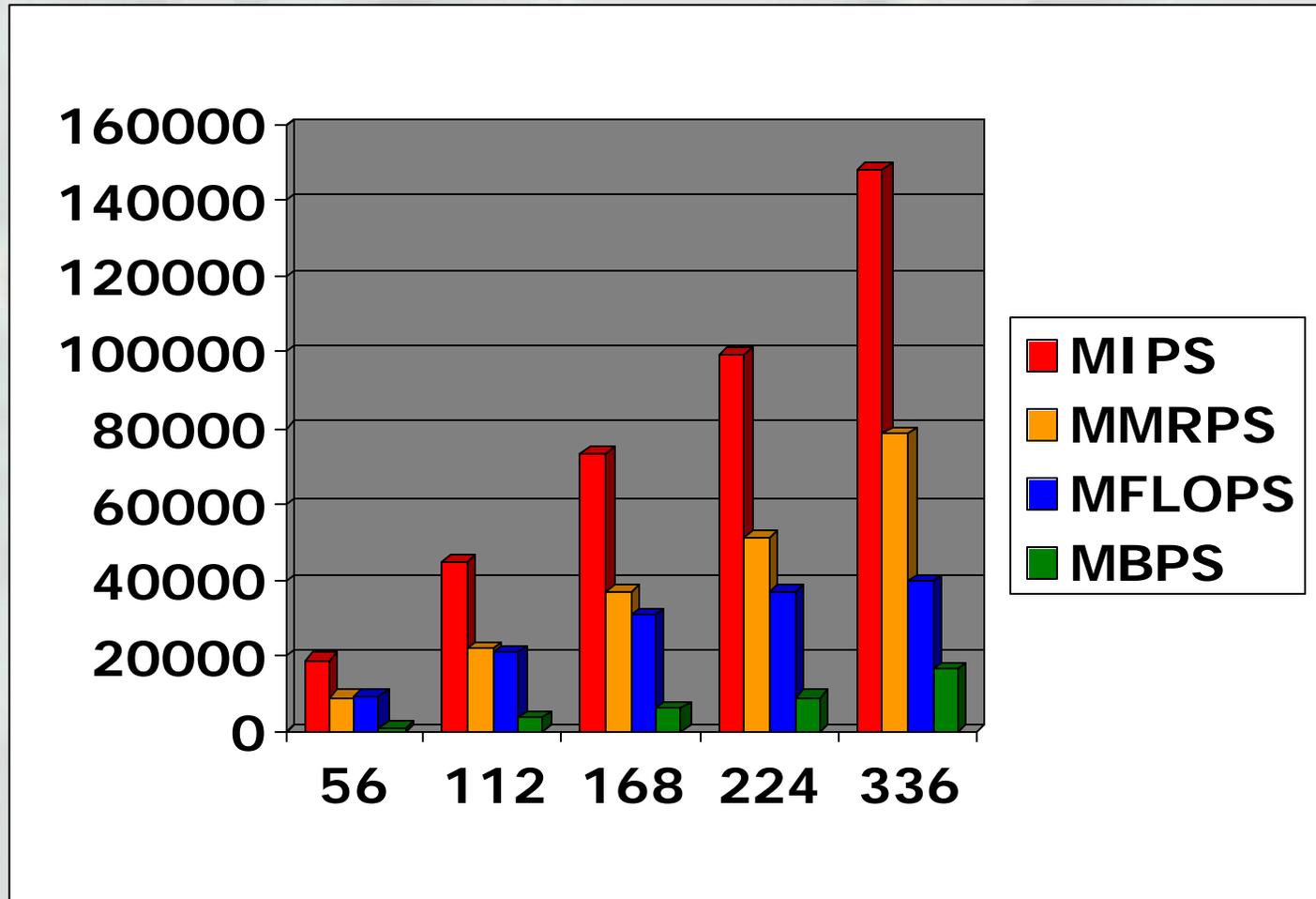
NLOM Profiling (Average per Process)





Major Shared Resource Center
ERDC VISIT

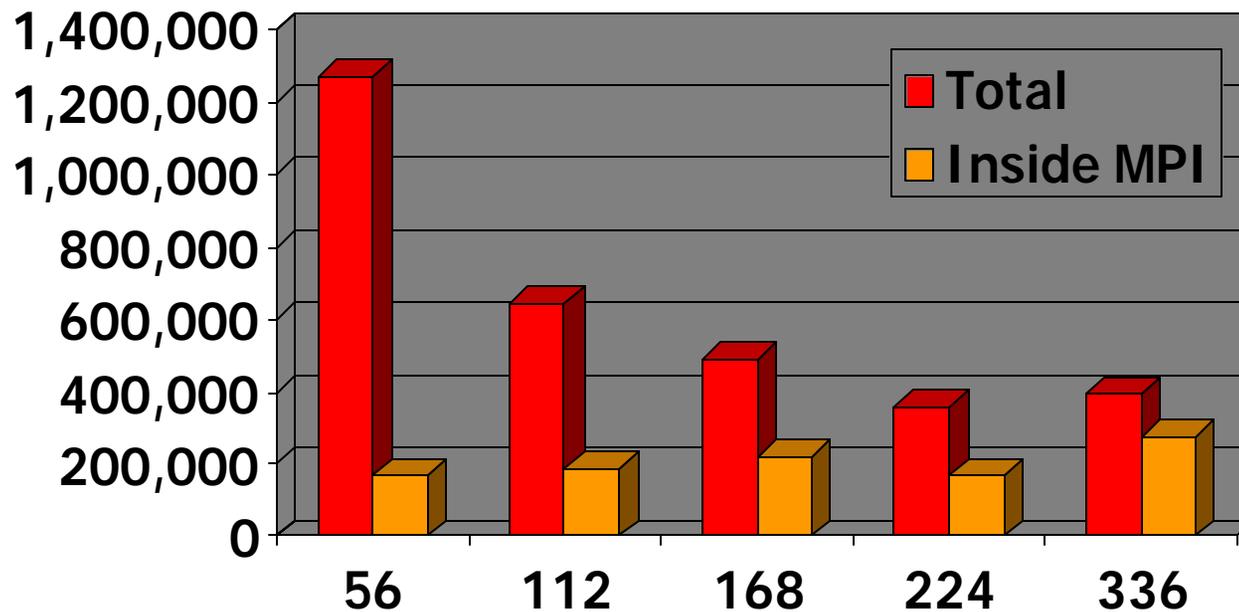
NLOM Profiling (Overall)





NLOM Profiling (MPI)

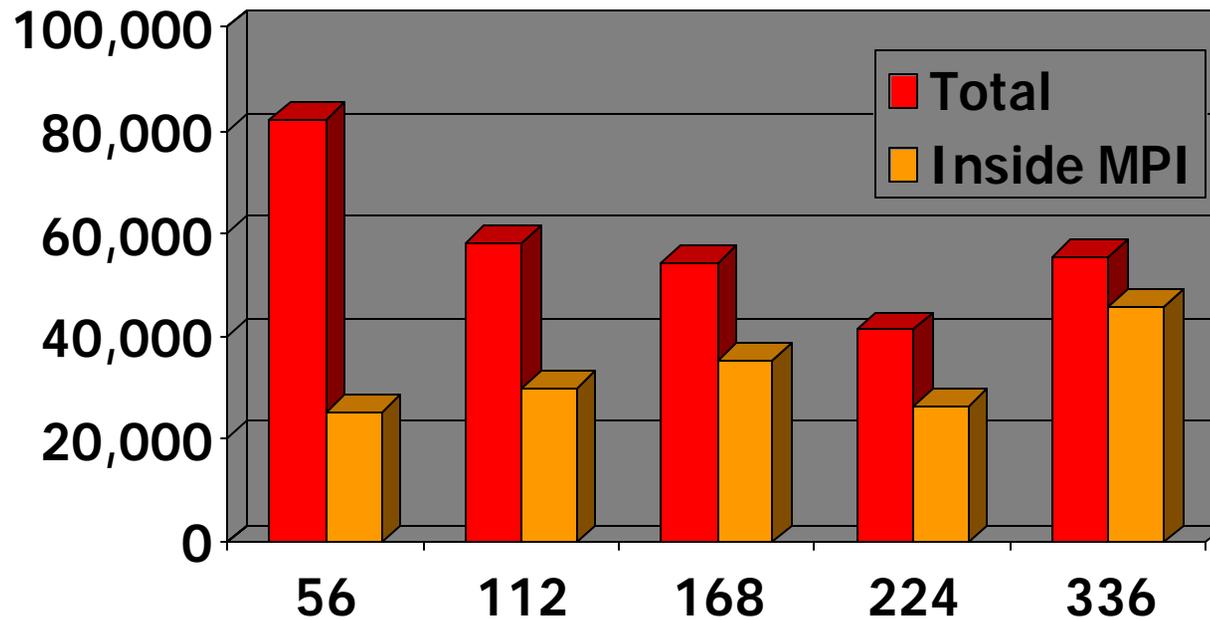
Average Cycles per Process (millions)





NLOM Profiling (MPI)

Average Branches per Process (millions)





NLOM Profiling (MPI)

Average Loads per Process (millions)

