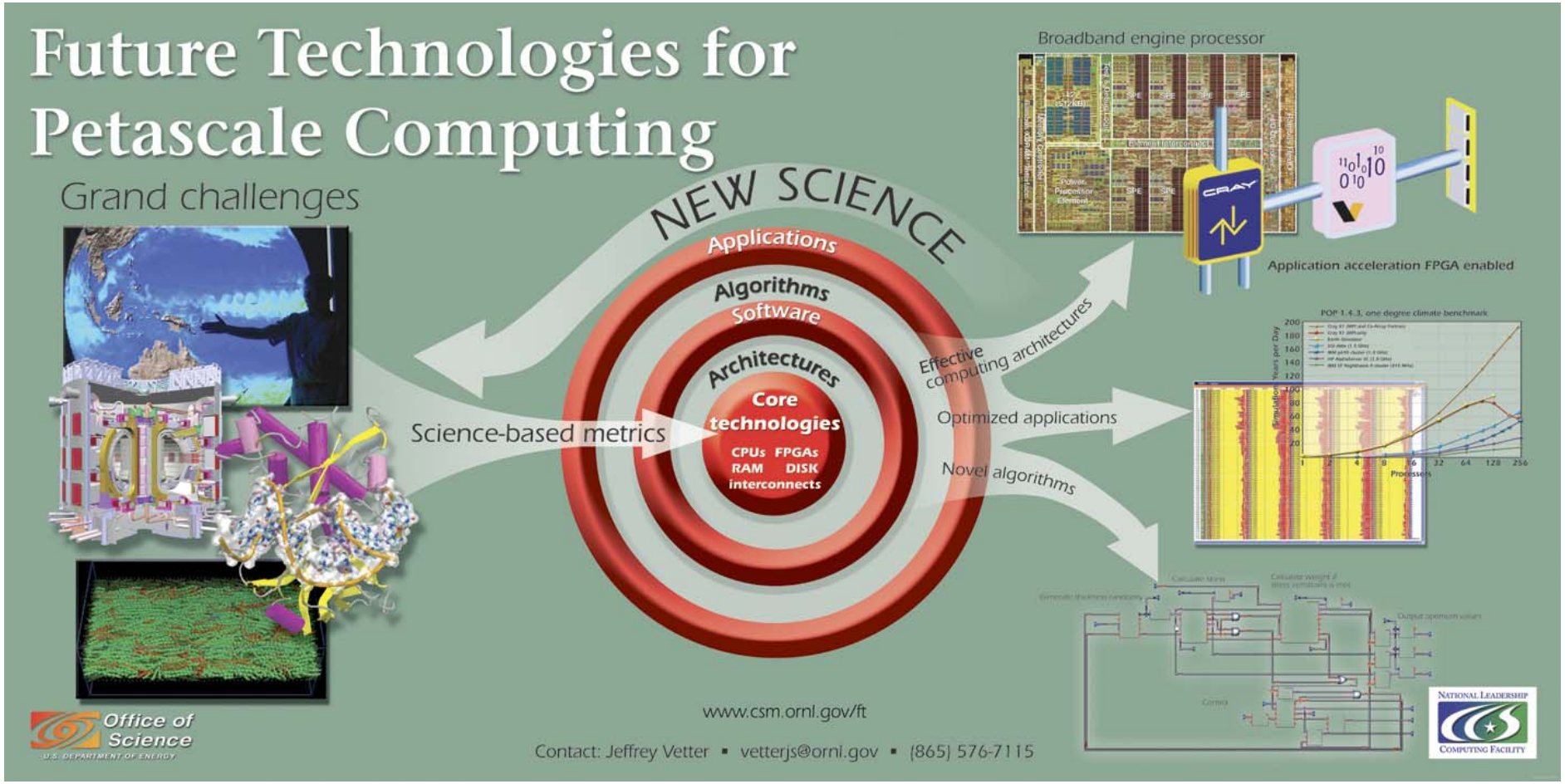
A Brief Overview of Activities in the Future Technologies Group at Oak Ridge National Laboratory

Jeff Kuehn

Future Technologies Group @ ORNL

kuehn@ornl.gov


Future Technologies for Petascale Computing
Grand challenges
NEW SCIENCE
Applications
Algorithms
Software
Architectures
Core technologies
CPUs FPGAs
RAM DISK
interconnects
Science-based metrics
Effective computing architectures
Optimized applications
Novel algorithms
Broadband engine processor
Application acceleration FPGA enabled
POP 1.4.3, one degree climate benchmark
Office of Science
U.S. DEPARTMENT OF ENERGY
www.csm.ornl.gov/ft
Contact: Jeffrey Vetter ▪ vetterjs@ornl.gov ▪ (865) 576-7115
NATIONAL LEADERSHIP COMPUTING FACILITY

Our Team and Collaborators

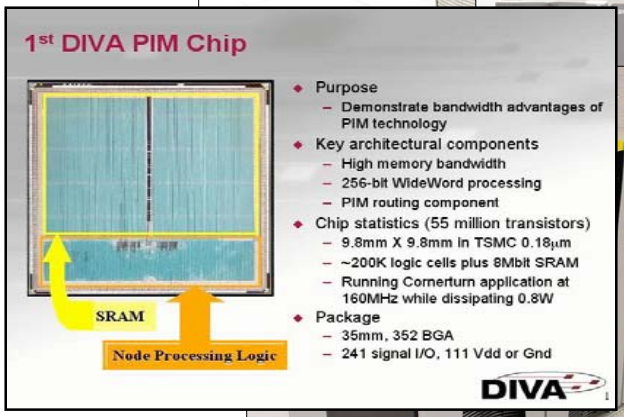
- Future Technologies Group
  - Sadaf Alam
  - Richard Barrett
  - Nikhil Bhatia
  - Jeff Kuehn
  - Collin McCurdy
  - Jeremy Meredith
  - Ken Roche
  - Philip Roth
  - Olaf Storaasli
  - Jeffrey Vetter
- Collaborators
  - Hong Ong
  - Melissa Smith
  - Pat Worley
  - Pratul Agarwal
  - SciDAC PERC Team
  - DARPA HPCS Team
  - DoD HPCMP
  - Georgia Tech CSE Dept, CERCS
  - Vendors
  - Many others…

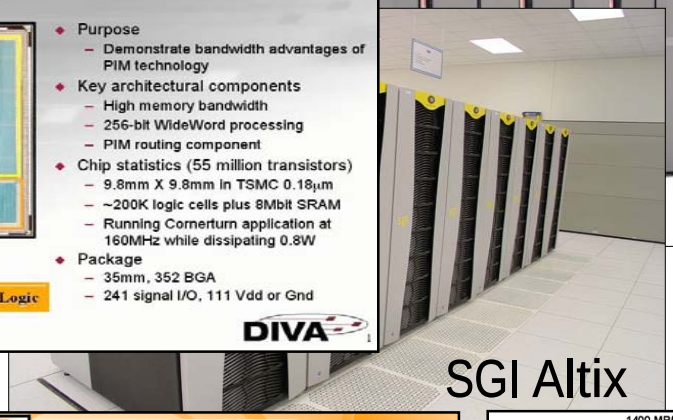
- Performance analysis, evaluation, and modeling of architectures in support of scientific computation
- Research and development of software and algorithms for HPC
- ExCL (Experimental Computing Lab) for examining new technologies
  - FPGAs
  - Array processors
  - Optical processors
  - Multicore processors
  - …

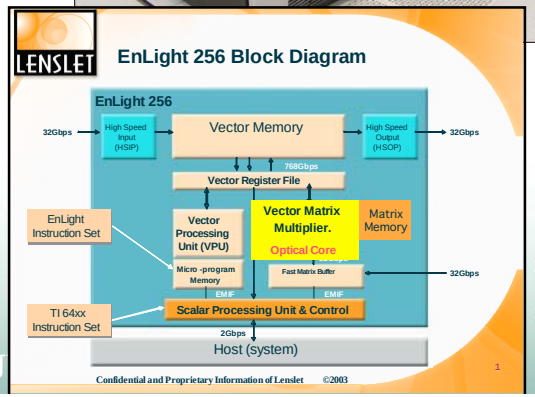
http://www.csm.ornl.gov/ft

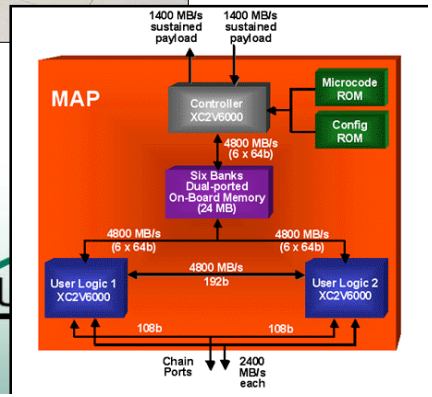
Investigating Diverse Architectures

- Cray X1e, XD1, XT3 (Red Storm)
- IBM BlueGene/L
- IBM SP3, p655
- SGI Altix
- Intel Itanium, Xeon
- IBM POWER5
- IBM Cell
- FPGAs
- Optical processors
- Processors-in-memory
- Multithreading
- Array processors, etc.



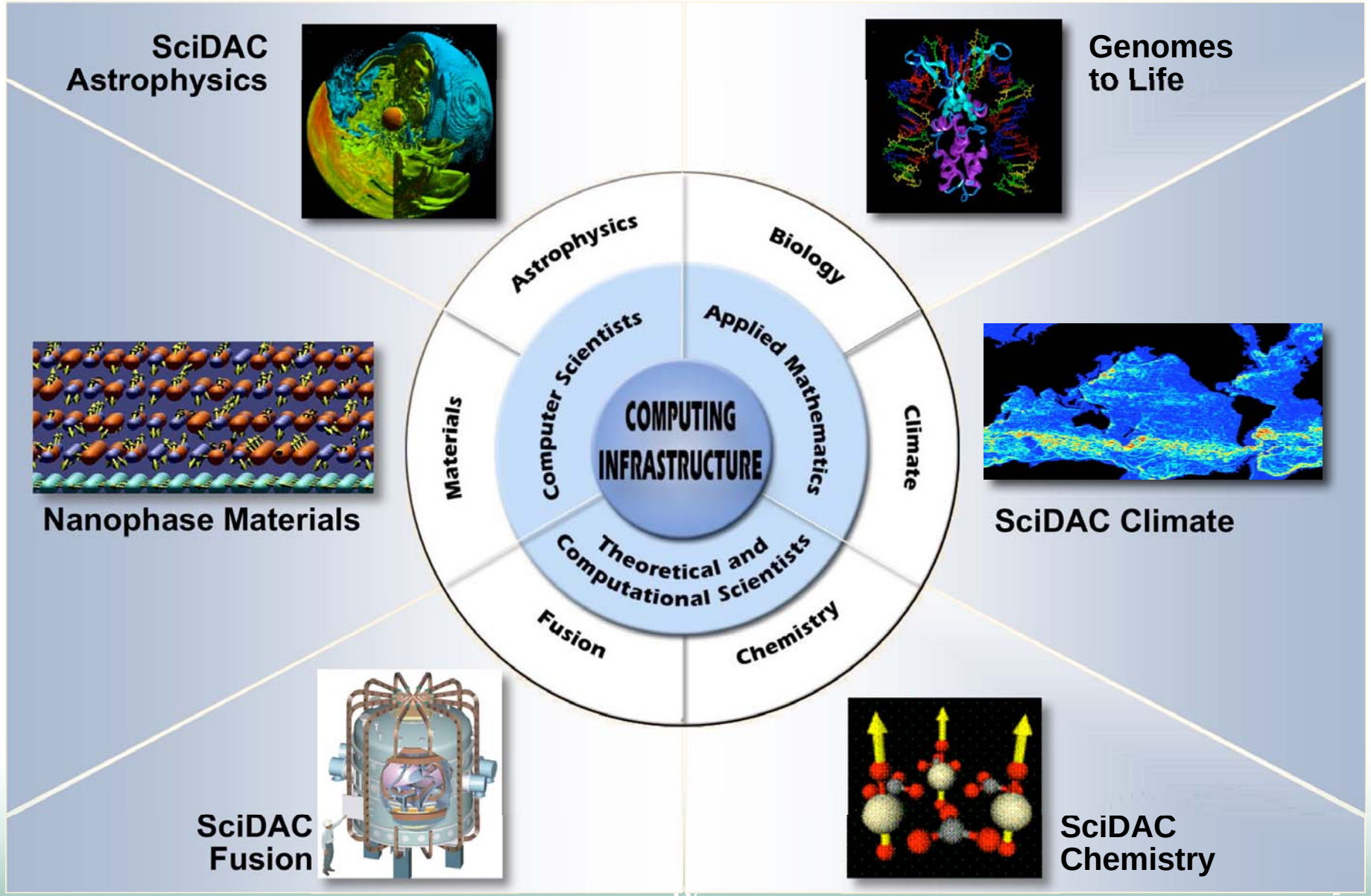






- This diversity makes the problem much more interesting!

ORNL has Major Efforts Focusing on Grand Challenge Scientific Applications



Recent and Ongoing Evaluations

- Cray X1
  - P.A. Agarwal, R.A. Alexander et al., "Cray X1 Evaluation Status Report," ORNL, Oak Ridge, TN, Technical Report ORNL/TM-2004/13, 2004.
  - T.H. Dunigan, Jr., M.R. Fahey et al., "Early Evaluation of the Cray X1," Proc. ACM/IEEE Conference High Performance Networking and Computing (SC03), 2003.
  - T.H. Dunigan, Jr., J.S. Vetter et al., "Performance Evaluation of the Cray X1 Distributed Shared Memory Architecture," IEEE Micro, 25(1):30-40, 2005.
- Cray XD1
  - M.R. Fahey, S.R. Alam et al., "Early Evaluation of the Cray XD1," Proc. Cray User Group Meeting, 2005, pp. 12.
- Cray XT3
  - J. S. Vetter, S. R. Alam et al., "Early Evaluation of the Cray XT3 at ORNL," Proc. Cray User Group Meeting, 2005, pp. 12.
- SGI Altix
  - T.H. Dunigan, Jr., J.S. Vetter, and P.H. Worley, "Performance Evaluation of the SGI Altix 3700," Proc. International Conf. Parallel Processing (ICPP), 2005.
- SRC
  - M.C. Smith, J.S. Vetter, and X. Liang, "Accelerating Scientific Applications with the SRC-6 Reconfigurable Computer: Methodologies and Analysis," Proc. Reconfigurable Architectures Workshop (RAW), 2005.

- Underway
  - XD1 FPGAs
  - ClearSpeed
  - EnLight
  - Multicore processors: AMD, Intel
  - IBM BlueGene/L

End

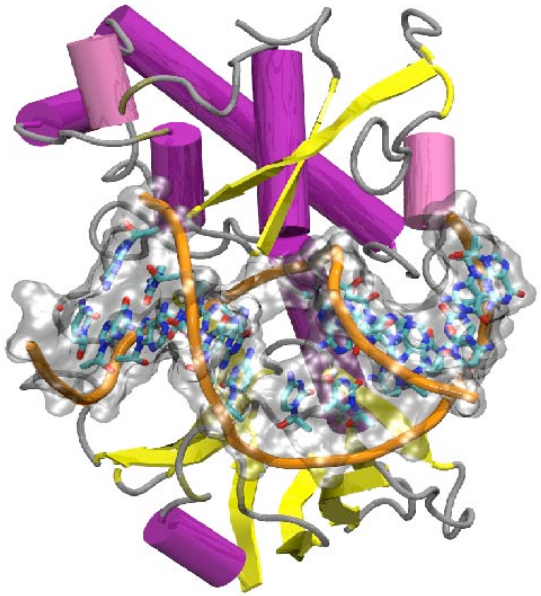
End

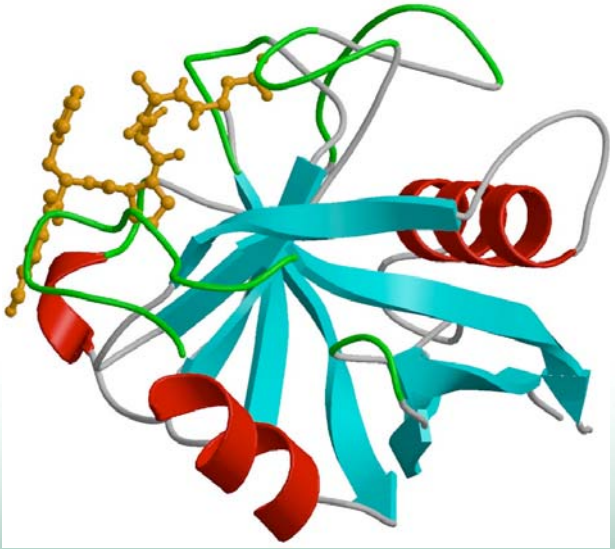
- CASE STUDIES:
  - Life Sciences
  - Computer Sciences

Case Study of a Life Sciences Application

Computational Biology using Molecular Modeling

- The structure, dynamics and function of biomolecular complexes are inter-related
- Various aspects of biomolecules structure and function span multiple scales of time and length
- Wide community of biologist are interested in the multi-scale modeling of biomolecules
- *Multi-scale modeling of a real system may require 1 peta-flop/s for an entire year!*
- Scaling of existing software packages and algorithms is limited





Joint work between Sadaf Alam and Comp Biologist Pratul Agarwal at ORNL.

Computer Simulations
(Molecular Dynamics)

⇨ Mathematical (potential) function

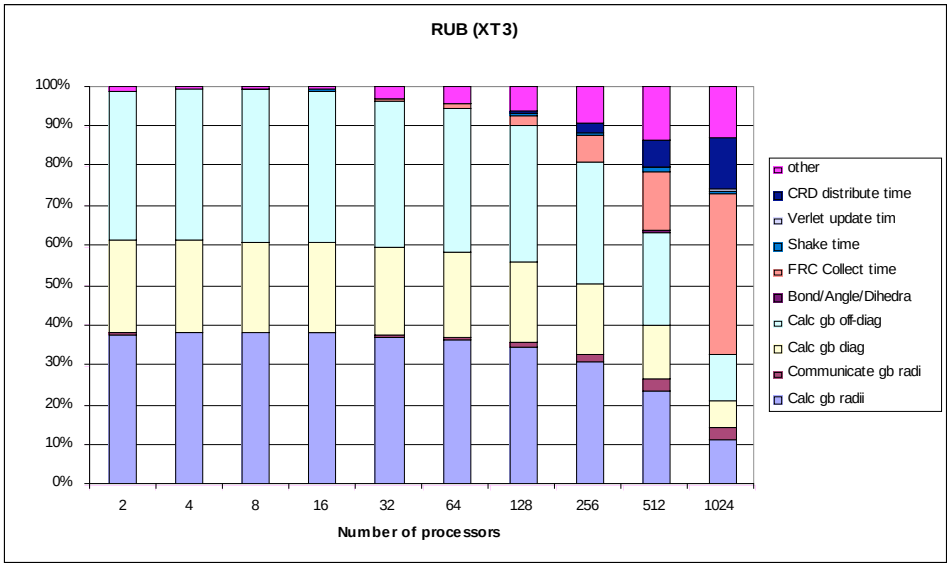
$$V(r^n) = \sum_{bonds} K_l (l - l_{eq})^2 + \sum_{angles} K_\theta (\theta - \theta_{eq})^2 + \sum_{torsions} \frac{V_n}{2}(1 + \cos[n\phi - \gamma]) + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\left[\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right] + \frac{q_i q_j}{\varepsilon r_{ij}} \right)$$

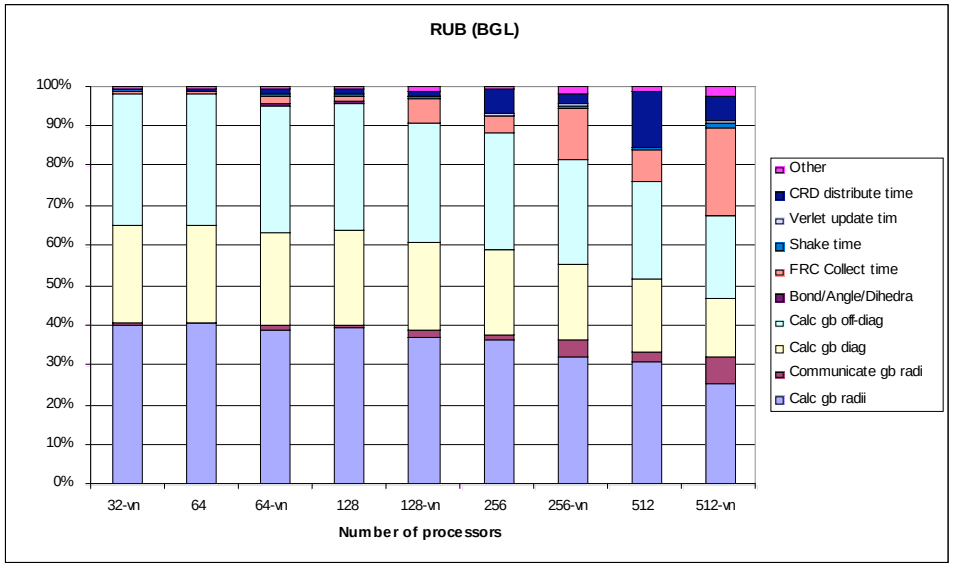
- Bond stretching, angle bending, angle torsion and the non-bond term
- Degree of freedom = 3N-6, where N=number of atoms
- Number of points to sample=M^{3N-6}, M >> 10
- Packages: Amber, GROMACS, GAMESS, LAMMPS, NAMD

AMBER Performance Analysis

- ORNL Computational biologists were using AMBER for their simulations, but its scalability was limited to about 128 processors
- Used several tools to study AMBER's performance
  - MPIP, PAPI, Xprofiler, GPROF
- Modified communication operations to improve scaling
- Identified computational kernels for acceleration with FPGAs

AMBER Profiling on Cray XT3 and IBM BlueGene/L





XT3

Bottlenecks: Distribute, Collect and I/O times

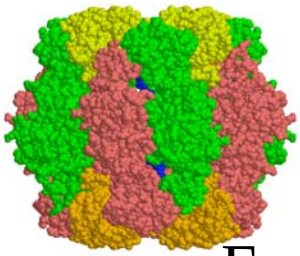
Expected to improve significantly as system matures

BGL

Bottlenecks: Distribute and collect times

Computation and communication times can improve with tool chain

Amber Control Flow for RUB
(RuBisCO with Generalized Born method)

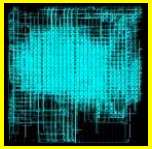


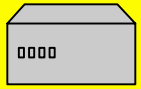
RunMD

Main method of Sander

Force | Shake | Update | CRD distribute

Non-bond | Bond | FRC Collect





Short_ene: mapped to FPGA
Divided and distributed in MPI model

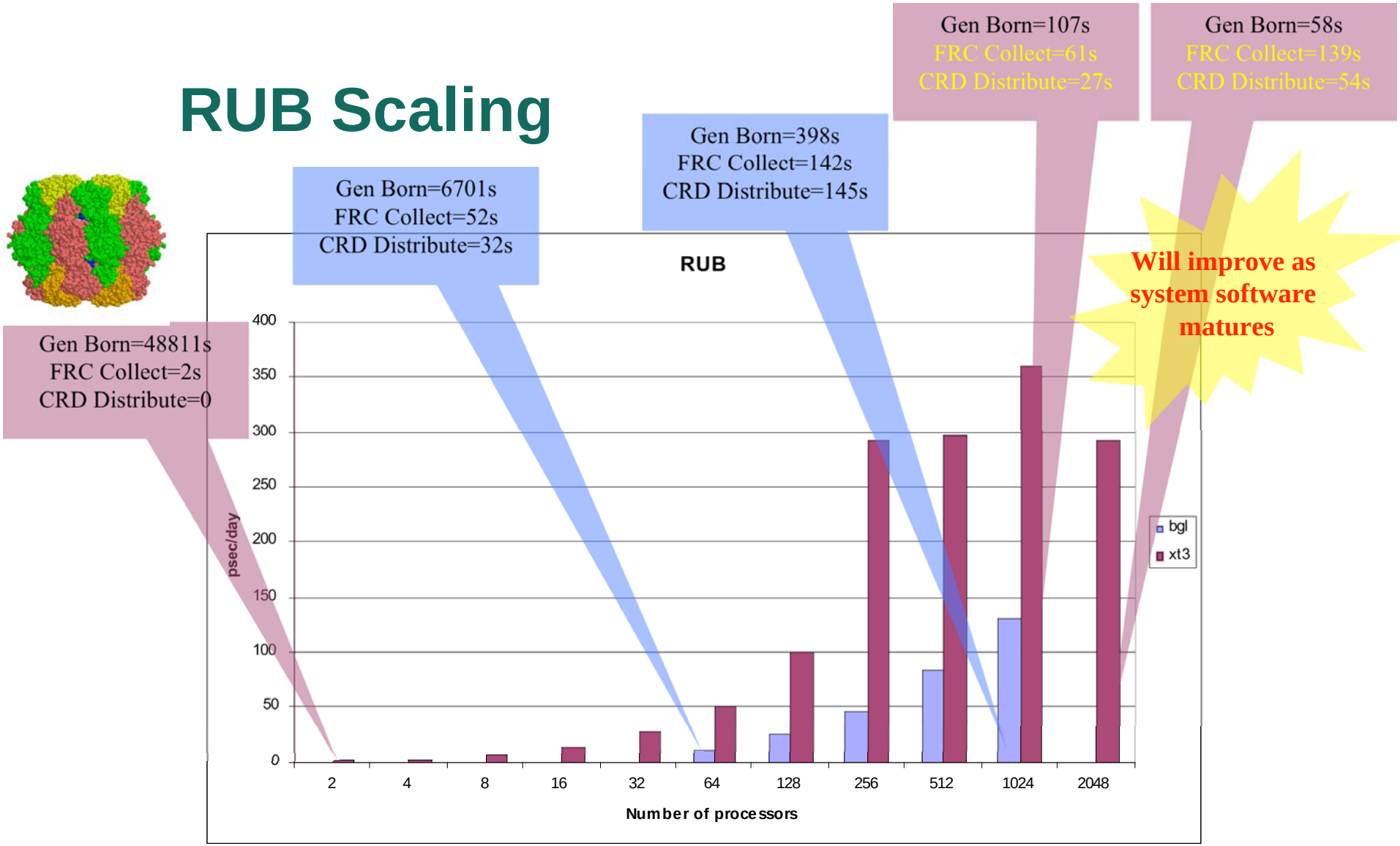
Cost increases with number of processors

Generalized Born

GB Computation | GB Communication

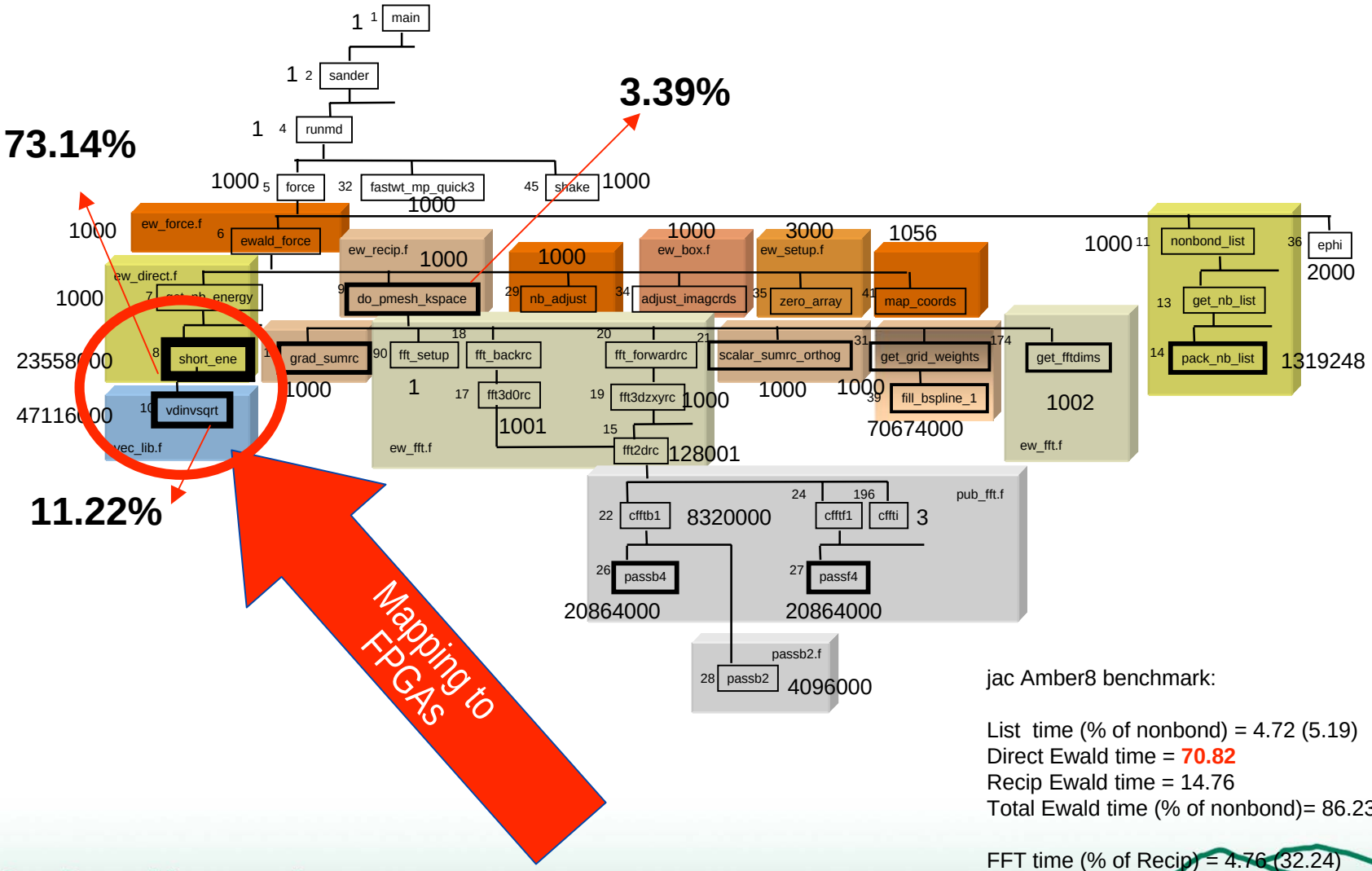
Most expensive on a single node system

RUB Scaling



Rubisco with Generalized Born solvation method (ORNLtest3). Note that on BGL only results from 64, 128, 256, 512 nodes run are shown.

Mapping Amber Kernel to FPGAs



jac Amber8 benchmark:

List time (% of nonbond) = 4.72 (5.19)
Direct Ewald time = **70.82**
Recip Ewald time = 14.76
Total Ewald time (% of nonbond)= 86.23 (**94.81**)

FFT time (% of Recip) = 4.76 (32.24)

AMBER Summary

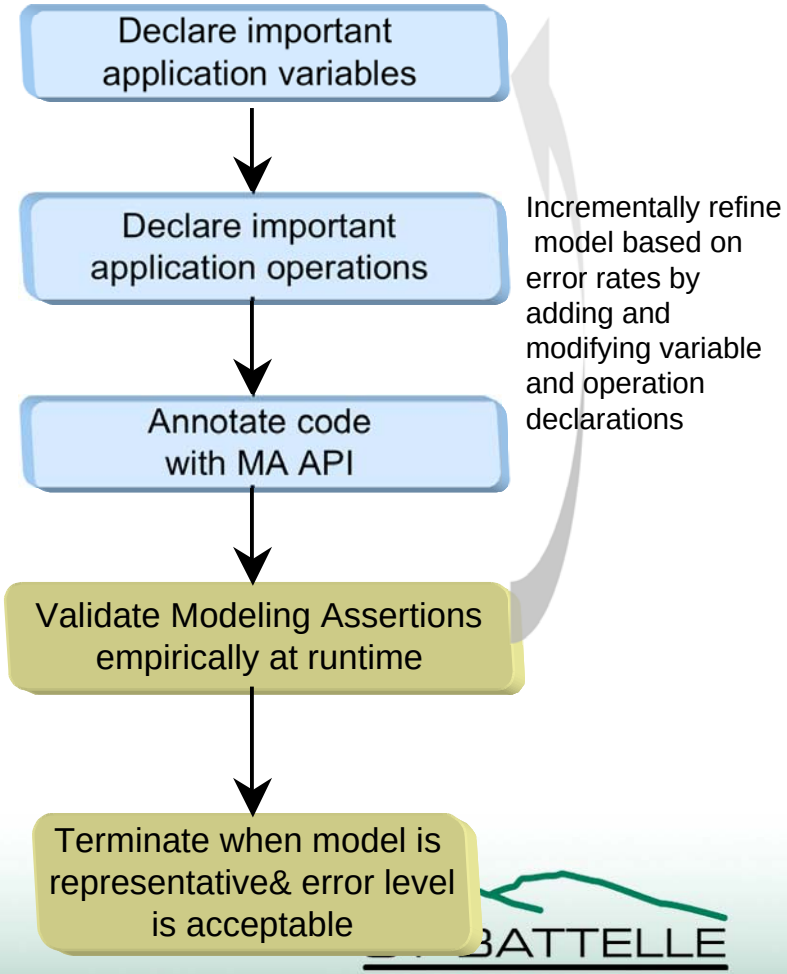
- Through performance analysis, we have identified the components that are limiting the scalability, and improved its scaling
- Amber scaling was limited to 128 nodes but now we have run experiments on 1024 nodes on Bluegene/L and on 2048 nodes on Cray XT3
- Achieved close to order of a nano-second/day on early evaluation stage supercomputing systems
- Mapping compute intensive kernel to SRC MapStation (a reconfigurable computing system)

New Modeling Methods: Symbolic Modeling with Modeling Assertions

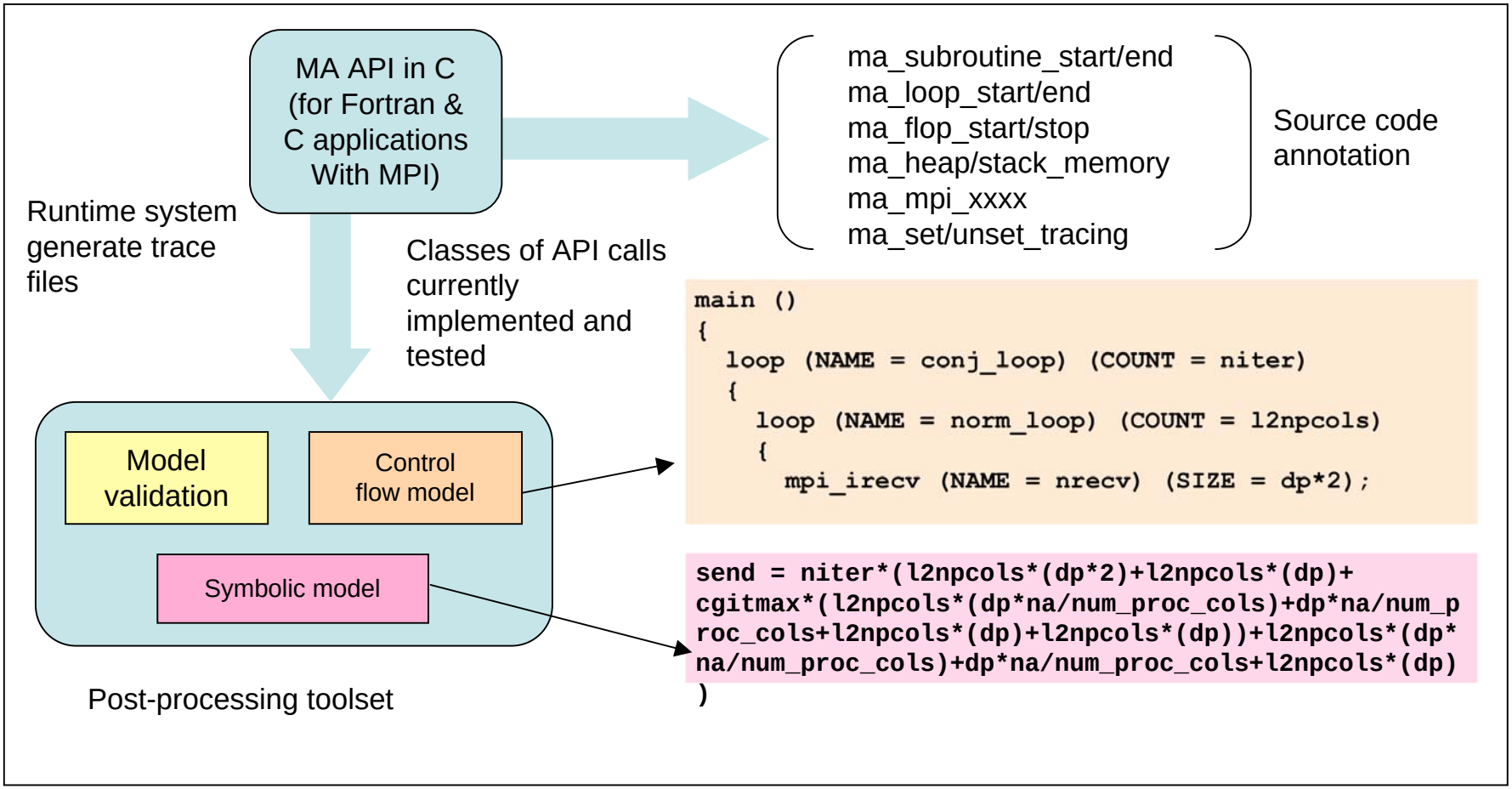
Symbolic models facility scaling and sensitivity analysis

Modeling Assertion (MA) = Empirical data + Symbolic modeling

- Advantages over traditional modeling techniques:
  1. Incremental construction of realistic performance models
  2. Isomorphism with the application structure
  3. Easy incremental validation of the model with empirical data
  4. Modularity, portability and extensibility
  5. Parameterized, symbolic models are evaluated with Matlab and Octave
- Demonstrate the use of MA using NAS MPI (3.2) CG and SP benchmarks
- Constructed, validated and projected floating-point operation count and MPI message volume as a function of input parameters



MA Framework




```
main ()
{
  loop (NAME = conj_loop) (COUNT = niter)
  {
    loop (NAME = norm_loop) (COUNT = l2npcols)
    {
      mpi_irecv (NAME = nrecv) (SIZE = dp*2);
```

```
send = niter*(l2npcols*(dp*2)+l2npcols*(dp)+
cgitmax*(l2npcols*(dp*na/num_proc_cols)+dp*na/num_p
roc_cols+l2npcols*(dp)+l2npcols*(dp))+l2npcols*(dp*
na/num_proc_cols)+dp*na/num_proc_cols+l2npcols*(dp)
)
```

Example application of/with modeling assertions

```
call ma_def_variable_int('na',na)
call ma_def_variable_int('nonzer',nonzer)
…..
call ma_def_variable_assign_int('num_proc_cols',
>   '2^ceil(log(nprocs)/(2*log(2)))',num_proc_cols)
…..
call ma_loop_start('conj_loop','niter',niter)
do it = 1, niter
…..
```

Input parameters: na, nonzer, niter and nprocs

Derived parameters: nz, num_proc_cols, l2cpcols and dp (size of REAL)

```
send = niter*(l2npcols*(dp*2)+l2npcols*(dp)+
   cgitmax*(l2npcols*(dp*na/num_proc_cols)+
   dp*na/num_proc_cols+l2npcols*(dp)+l2npcols*(dp))+
   l2npcols*(dp*na/num_proc_cols)+
   dp*na/num_proc_cols+l2npcols*(dp))
```

```
…..
call ma_loop_end('conj_loop',it-1)
…..
call ma_subroutine_start('conj_grad')
……
call ma_loop_start('cj_matvec','l2npcols',l2npcols)
   do i = l2npcols, 1, -1
      call ma_mpi_irecv('l2rcv','dp*na/num_proc_cols',
>                        dp*naa/npcols,l2npcols)
      call mpi_irecv( q(reduce_recv_starts(i)),
>                     reduce_recv_lengths(i),
>                     dp_type,
……
```

End markers used for validation

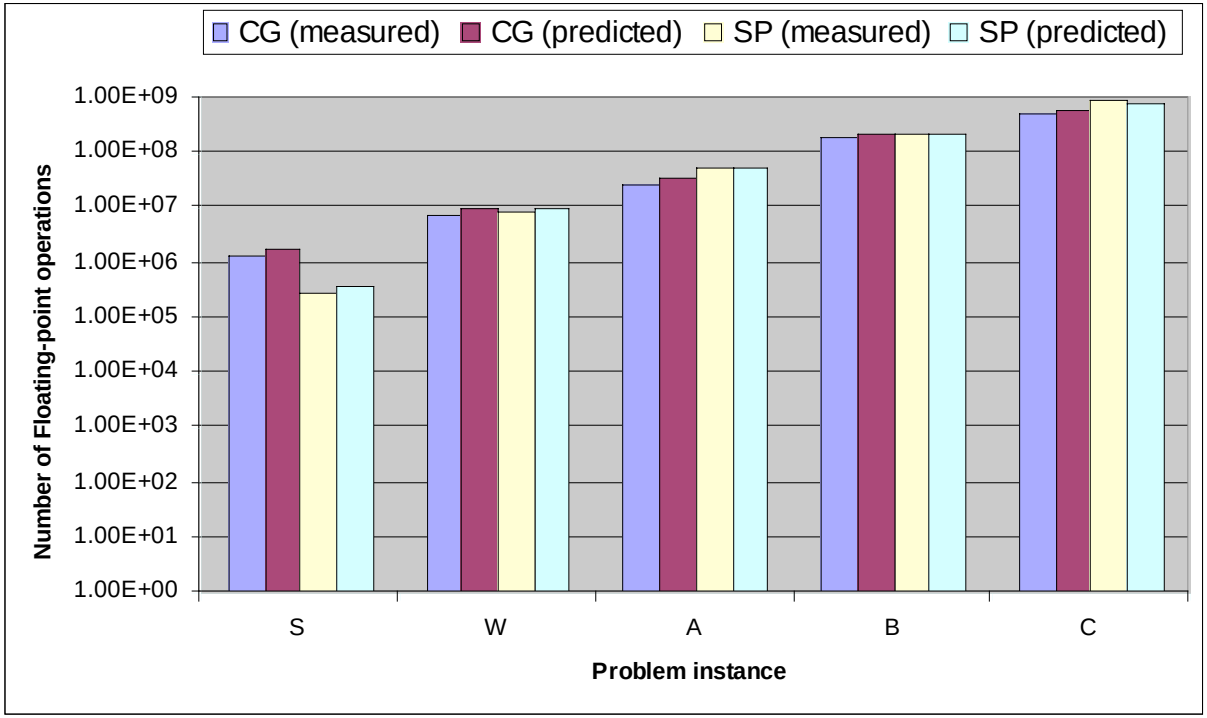
Markup for subroutine invocation

MA MPI API call

Model Validation

```
NAS CG
Class S: na=1400, nonzer=7
Class W: na=7000, nonzer=8
Class A: na=14000, nonzer=11
Class B: na=75000, nonzer=13
Class C: na=150000, nonzer=15
```

```
NAS SP
Class S: problem_size=7
Class W: problem_size=36
Class A: problem_size=64
Class B: problem_size=102
Class C: problem_size=162
```



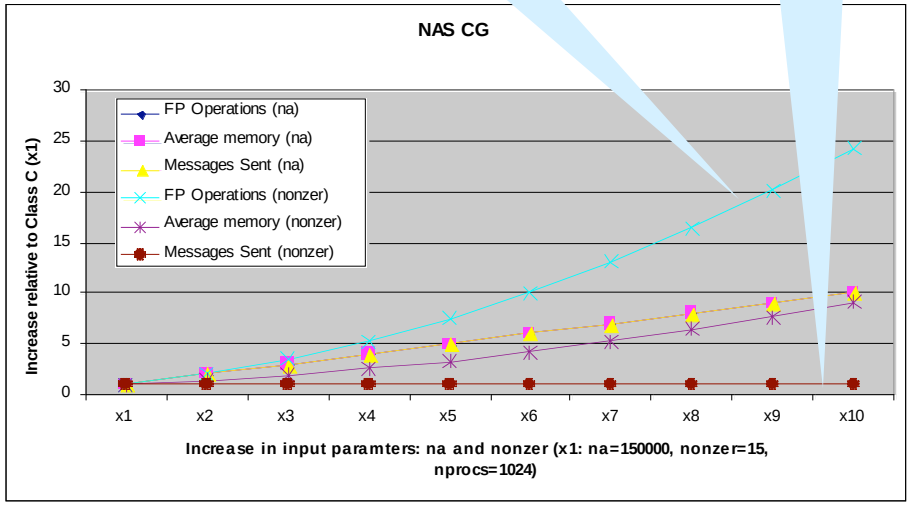

```
opq:        ma_flop:7000:7000:0.0: PASS=50: FAIL=0
cj_sumred:        ma_loop:1:1:0.0: PASS=50: FAIL=0
l4rcv:        ma_mpi_irecv:8:8:0.0: PASS=50: FAIL=0
l4snd:        ma_mpi_send:8:8:0.0: PASS=50: FAIL=0
sumred:        ma_flop:1:1:0.0: PASS=50: FAIL=0
floprhopq:        ma_flop:21001:21001:0.0: PASS=50: FAIL=0
cj_rho:        ma_loop:1:1:0.0: PASS=50: FAIL=0
l5rcv:        ma_mpi_irecv:8:8:0.0: PASS=50: FAIL=0
l5snd:        ma_mpi_send:8:8:0.0: PASS=50: FAIL=0
flopbeta:        ma_flop:7002:7001:1.426E-4: PASS=6: FAIL=44
flopnzx:     ma_flop_start:3503:4347:-0.194: PASS=0: FAIL=2
```

Model validation output

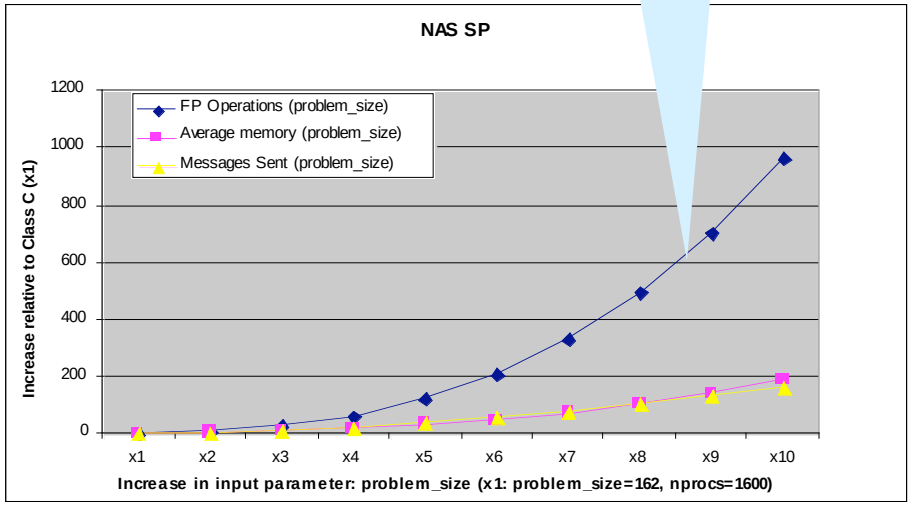
Sensitivity Analysis

FP operations sensitive to nonzer

Message volume if invariant to nonzer



Number of FP operations are relatively more sensitive to problem_size



Increase in the number of floating-point operations, memory requirement per node and message volume per processor as a function of input parameters of NAS CG bnechmark: na and nonzer. Number of MPI tasks is fixed: nprocs=1024. Baseline value (x1) = Class C input values [na=150000, nonzer=15]

Increase in the number of floating-point operations, memory requirement per node and message volume per processor as a function of input parameter of NAS SP benchmark: problem_size. Number of MPI tasks is fixed: nprocs=1600. Baseline value (x1) = Class C input values [problem_size=162]