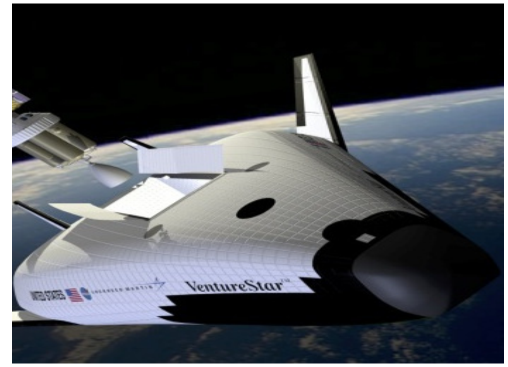
Benchmarking of Parallelized Thermal Solver Technology Using NX9 Space Systems Thermal

Carl Poplawsky, Maya HTT
Chris Blake, Maya HTT
Chris Jackson, Maya HTT

Thermal & Fluids Analysis Workshop
TFAWS 2013
July 29-August 2, 2013
Kennedy Space Center
KSC, FL

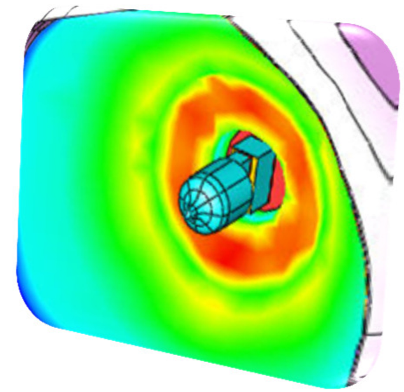
Summary

- Maya has added parallelization to the main analyzer module (ANALYZ) for NX9 Space Systems Thermal
  - NX9 due to be released in Oct. 2013
- Available for Windows and Linux
- Can take advantage of an unlimited number of processors with optional license
  - Up to 8 processes with standard license
- This paper presents benchmarking of this new capability
  - Normalized solution speeds during ANALYZ and for the entire solution
  - 2 test hardware configurations
  - 10 thermal models



MAYA Heat Transfer Technologies

- Software development partner / reseller for Siemens PLM
- Over 25 years of experience in design & development of Thermal, Flow and Structural simulation software
- Siemens OEM software developed by Maya
- Engineering consulting services
- Software sales, implementation, training
- Software customization



NX Space Systems Thermal

- Comprehensive application for simulation of all heat transfer mechanisms in spacecraft
- Finite-volume based solver technology using finite elements for modeling purposes
- Thermal model is fully associative with CAD geometry for parametric analysis
- Specialized spacecraft tools:

– Built-in Orbit Modeler

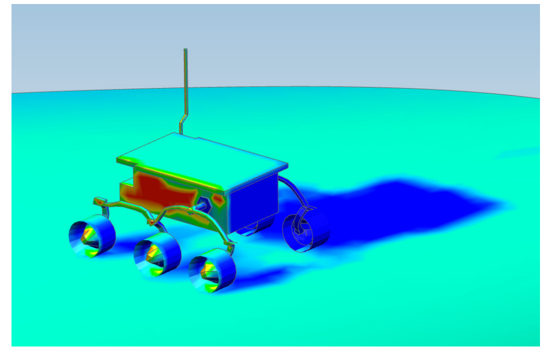
– Deterministic ray tracing

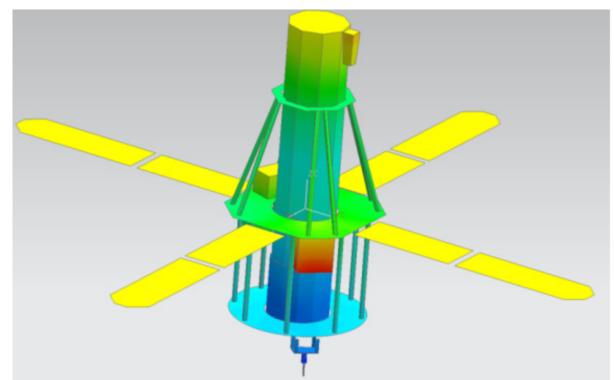
•Specular reflections & transparency

–Monte Carlo ray tracing

–Parallel Processing for view factors

– Transient articulation and moving mesh rigid body motion, including parent-child



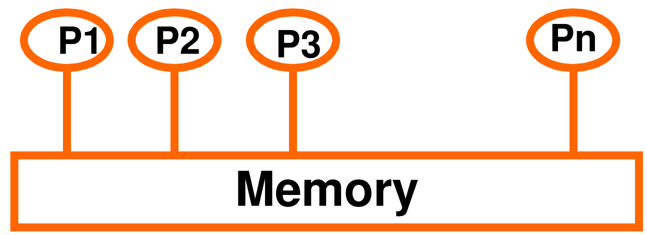


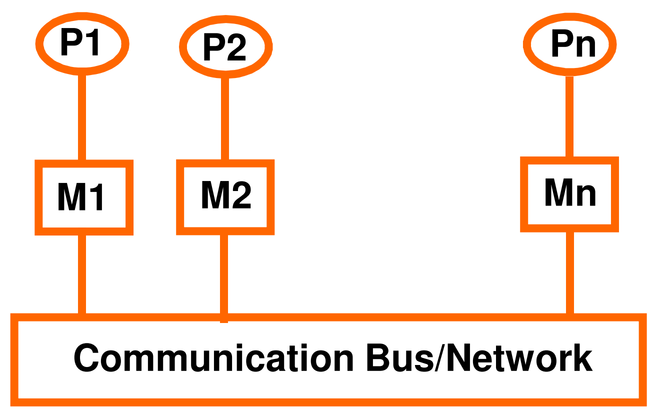
Challenges of Parallelizing a Thermal Analyzer

- Implicit methods are inherently more difficult to parallelize than explicit methods
  - Linear system solvers do not scale well to large number of processors with a fixed problem size
  - Poisson-type problems (e.g., conduction) scale less well than other types of problems
  - With typical hardware configurations and models we see today, it is still more efficient to stick with implicit methods vs. moving to explicit with very small timesteps
- Surface-to-surface radiation complicates parallelization
  - Matrices become less sparse; most of research in parallel linear solvers is applied to sparse systems
  - If domain decomposition is used, it potentially creates more connections between domains
- Different hardware require different approaches to parallelization
  - Multicore processors, Clusters, GPU's

Parallel Computing: SMP versus DMP

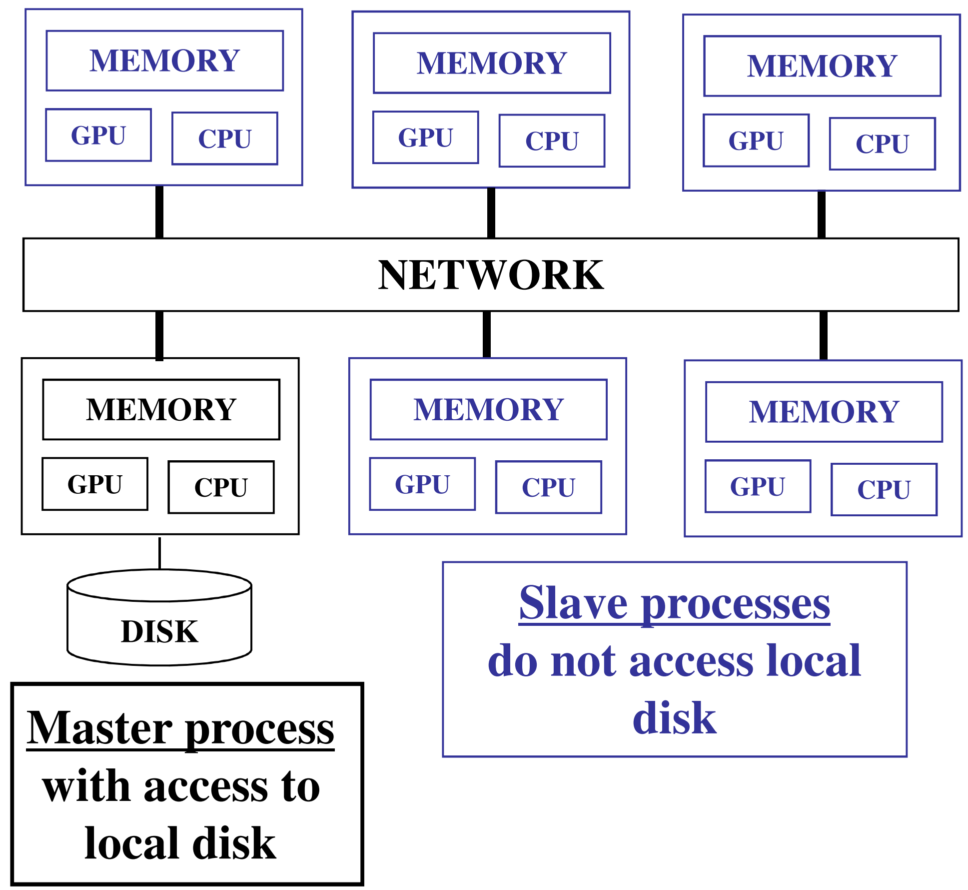
- Shared Memory Parallelization (SMP)
  - Multiple processes (execution threads) access the same memory and I/O
  - Popular SMP protocols: OpenMP, MS Windows Threads, Pthreads
- Distributed Memory Parallelization (DMP)
  - Each process has its own dedicated memory
  - Inter-process communication is used
  - Popular DMP protocol: MPI (Message Passing Interface)
  - Single-host multi-core or multi-host runs





Pre-NX9 DMP Parallelization of View Factor Calculations

- **Parallel HEMIVIEW,VUFAC:**
  - Master/Slave system
  - Master:
    - Performs all I/O
    - Sends model to slaves
    - Instructs slaves which VFs to compute
    - Receives VFs from slaves and writes results to single file
    - Computes some VFs when it has time
  - Slave
    - Receives model, instructions
    - Computes VF's
    - Sends VF's to Master
  - Dynamic load balancing to assure all processes are busy

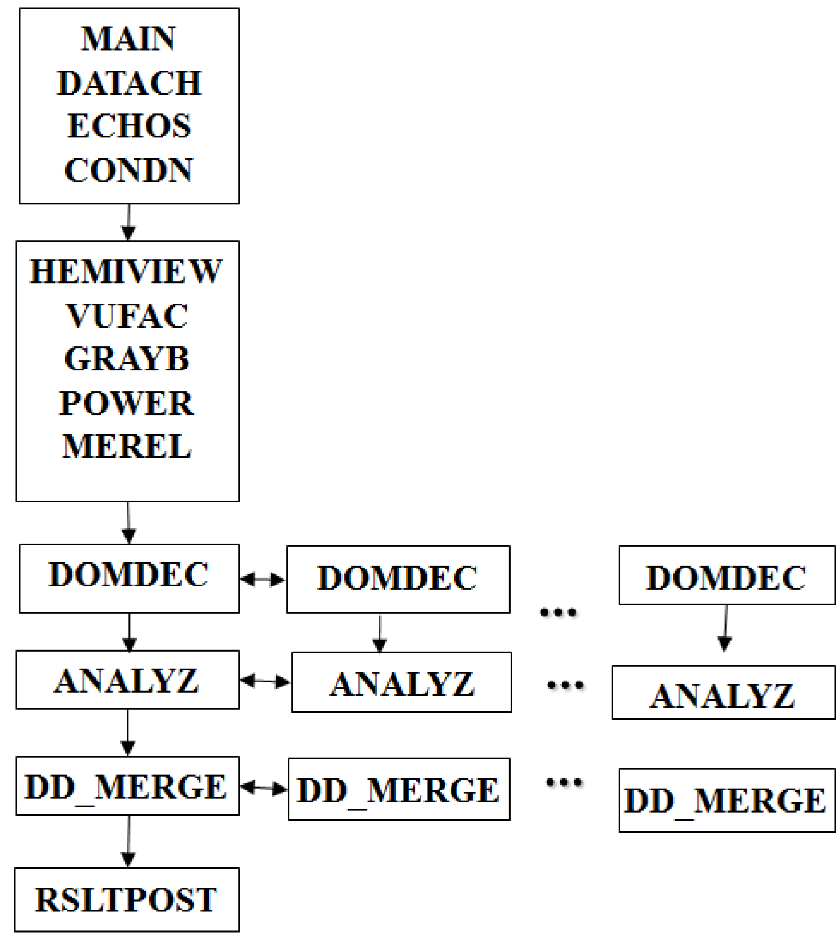


Pre-NX9 SMP Parallelization of Analyzer

- Applies to a limited selection of inherently parallelizable bottlenecks
  - Matrix assembly, matrix-vector-multiplication, dot products, vector scaling
  - Performance impact is limited by the relative weight of those bottlenecks in the run time
- Implemented with MS Windows threads (on Windows) and Pthreads (on non-Windows platforms)
- No MPI installation/setup needed, does not change solve convergence or overall execution workflow
  - Just set the number of threads to use (an Advanced Parameter)
- Dynamic load balancing
  - The work is split into a set of many chunks, which are concurrently processed by different execution threads
- Limited speedup (max ~25%)

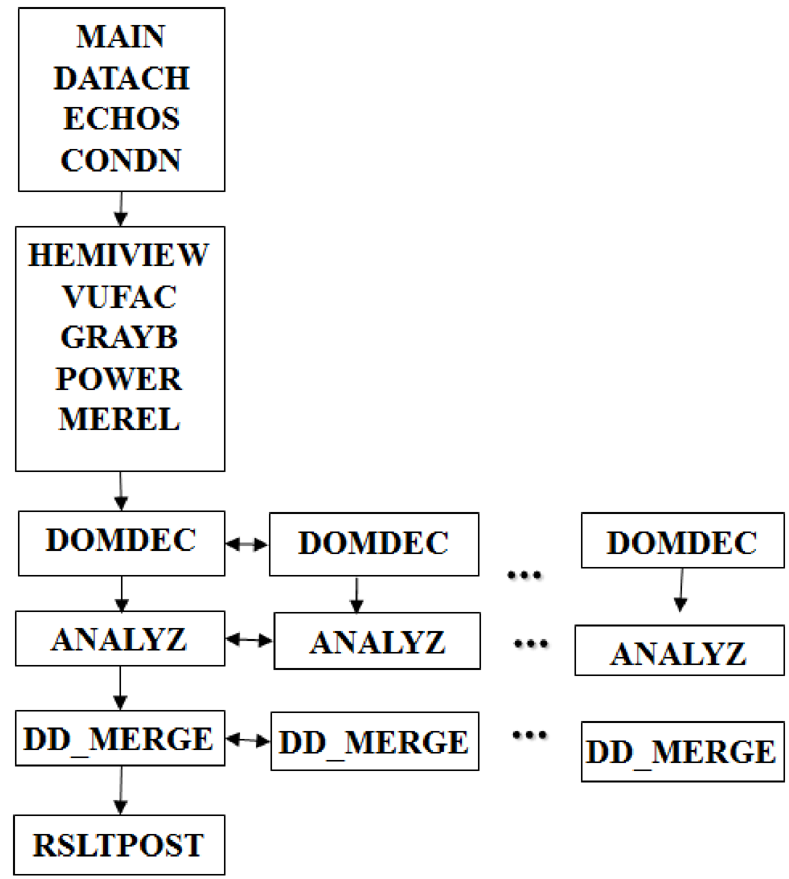
NX9 DMP Parallelization of Analyzer

- Domain decomposition approach
- Implemented with MPI
- Each analyzer process
  - runs in its dedicated temporary directory
  - starts from its own modified input files (domain's submodel)
  - does its own portion of results data output
- DOMDEC module
  - Does model partitioning and interdomain data dependencies analysis
  - Sets up domain directories
  - Splits and distributes analyzer input files into domain directories



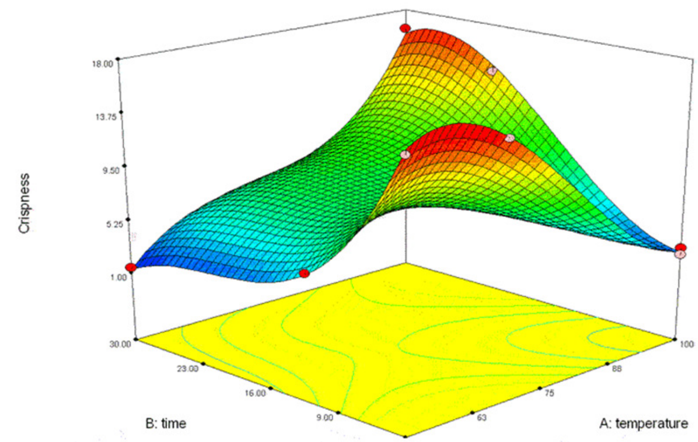
NX9 DMP Parallelization Domain Decomposition

- Model partitioning into domains is accomplished in the DOMDEC module prior to launching ANALYZ on each node
  - The full set of model elements is split into subsets (domains)
  - For a given domain, all elements are classified by how they thermally connect with other elements
  - Certain elements are further split into subcategories to optimize handling of inter-domain data dependencies
- The DOMDEC module balances the work load between processors (static balancing) while minimizing inter-process communications



Benchmark Tests

- Two sets of benchmarks tests were performed
- Spacecraft Test
  - Single node Windows 7 Power Workstation
  - Three existing customer spacecraft models
- Standard Test
  - Four node Linux cluster
  - Seven models including spacecraft, turbo machinery, aircraft, and QA; most models radiation dominated

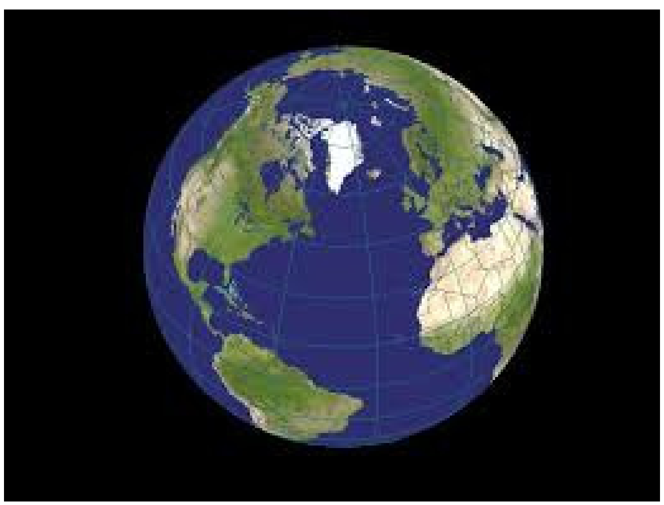


Spacecraft Test Hardware

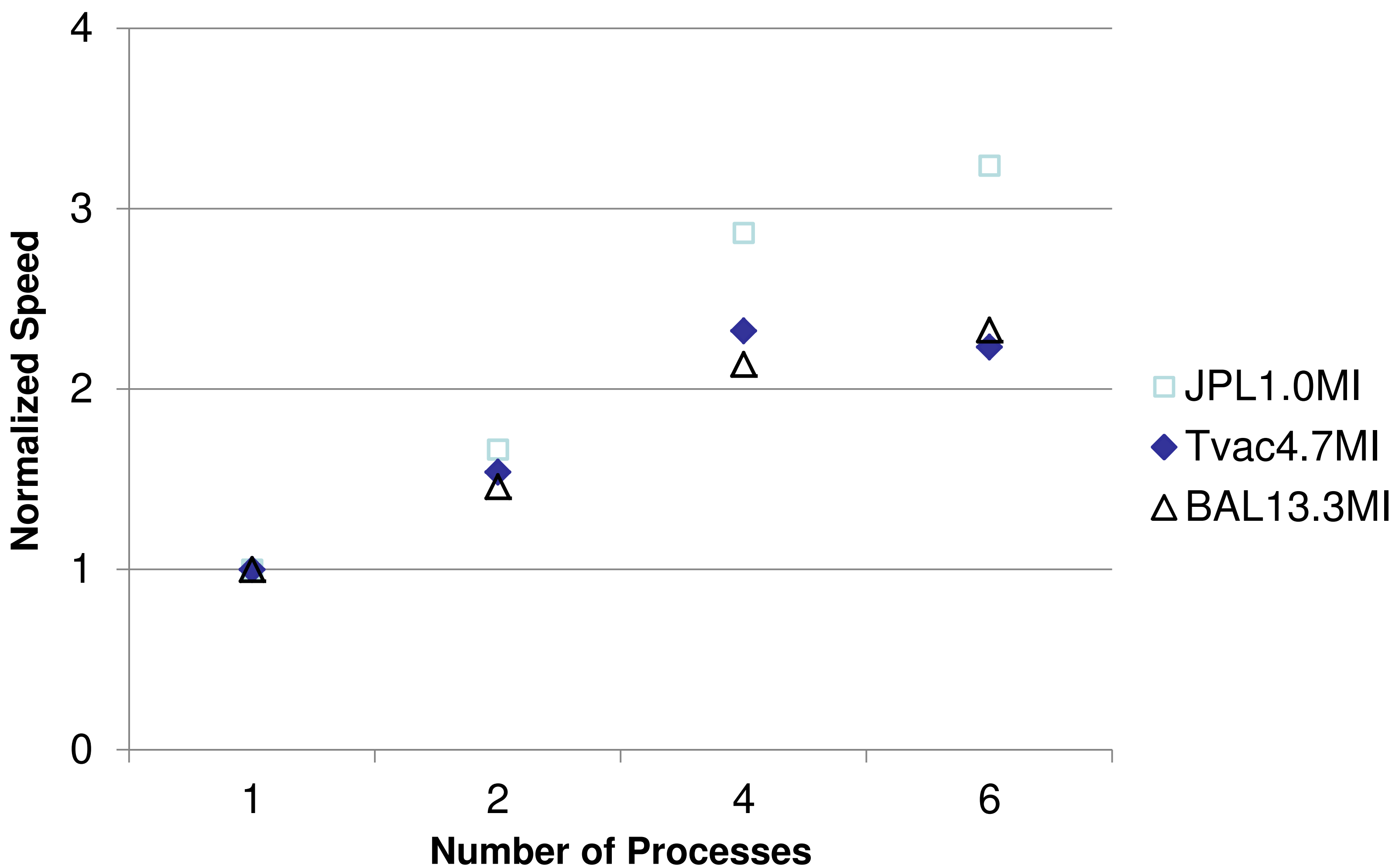
- **Power workstation (single node):**
  - Intel Core i7-3930K CPU (3.20 GHz)
    - 6 Cores
  - 64 GB RAM
  - Windows 7 Professional 64-bit operating system
  - Still uses domain decomposition; however, memory is shared

Spacecraft Test Models

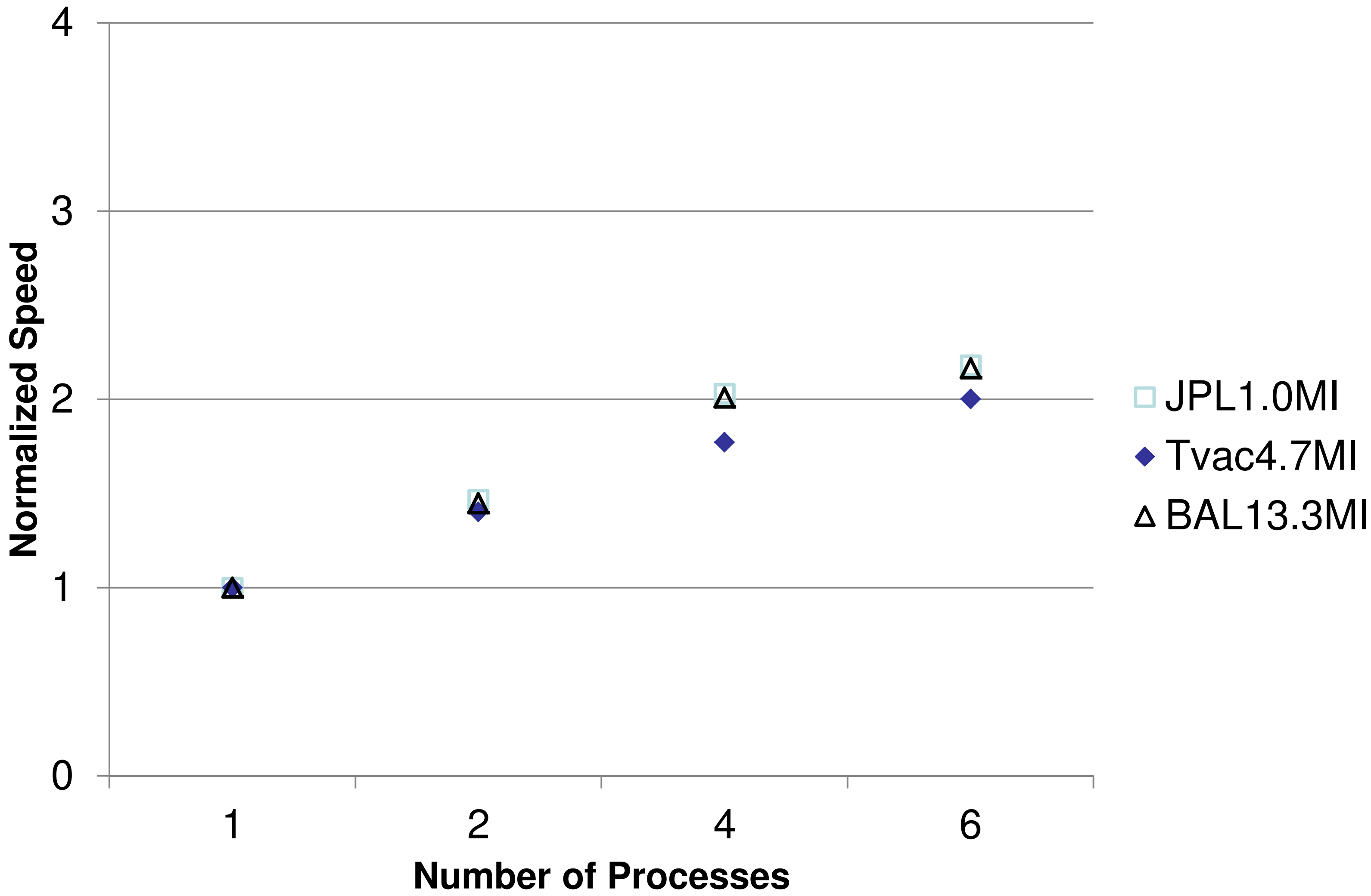
LABEL	ELEMENTS	CONDUCTANCES		ENCLOSURES	COMMENTS
		RADIATIVE	TOTAL		
Tvac4.7MI	80K	4.5M	4.7M	1	small S/C in thermal vacuum chamber
JPL1.0MI	18K	1.01M	1.04M	5	small JPL S/C model in orbit
BAL13.3MI	190K	12.8M	13.3M	24	large S/C model in orbit



Normalized Solution Speed During ANALYZ



Normalized Solution Speed – Complete Analysis



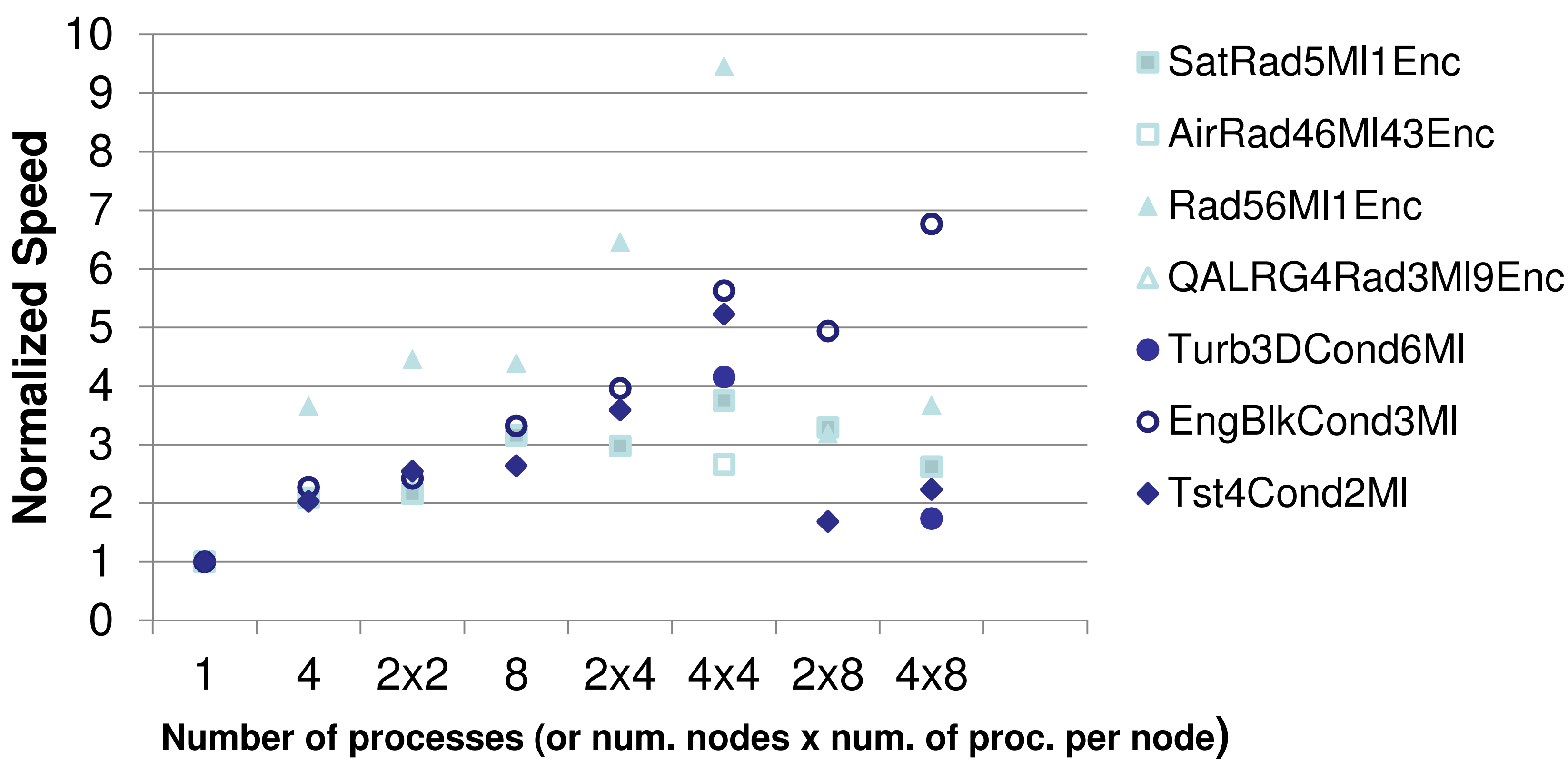
Standard Test Hardware

- **Server cluster**
  - 4 nodes, 8 cores/node
    - 2 Zeon X5550 processors (Nehalem) 2.66 GHz, per node
    - 48 GB DDR3 RAM at 1066 MHz per node
    - Local hard drive for each node
  - Sun Grid Engine (SGE) scheduler
  - Linux operating system

Standard Test Models

LABEL	ELEMENTS	CONDUCTANCES		ENCLOSURES	COMMENTS
		RADIATIVE	TOTAL		
SatRad5MI1En	73K	4.9M	5.0M	3	small satellite model in orbit
AirRad46MI43Enc	1.5M	41M	46M	43	aircraft model
Rad56MI1Enc	71K	55.8M	56.0M	1	large radiation enclosure
QALRG4Rad3MI9Enc	113K	2.8M	3.0M	9	large QA model
Turb3DCond6M	1.7M	620	6.4M		large 3D turbo machinery model
EngBlkCond3M	1.7M	0	1.7M	0	finely meshed engine block
Tst4Cond3MI	.7M	0	2.3M	0	large conductive model

Normalized Solution Speed During ANALYZ



Observations

•Running parallelized ANALYZ improves the speed of ANALYZ approximately 3-6X, and the speed of the total solution approximately 2-3X

•The performance benefit of additional cores per node seems to diminish after 4-6 cores

–Competition for resources (I/O and memory access bandwidth) between the cores starts to negatively impact performance

–For a given number of total cores, a configuration spread across multiple nodes will outperform a configuration with fewer nodes

•Improvements in newer generations of chips often focus on improving the efficiency with which cores can access and share resources

–In this case the Windows machine (spacecraft test models) has a newer processor, and clearly shows continued improvement beyond 4 cores/node

Observations

•As solid state drives become more prevalent, it will be interesting to see how their improved I/O will influence normalized speeds

•Sensitivity to available memory is another area worth investigation

Benefits

- Users running parallelized ANALYZ in NX Space Systems Thermal can expect the speed of the total solution to improve 2-3X
- Enables the user to:
  - Complete large thermal runs in significantly reduced real time
  - Run more parametric analyses within a given time frame
  - When necessary, have more ability to refine meshes, reduce integration time steps, increase radiation solution resolution, etc.
  - Provide more flexibility when defining solution strategy, for instance whether to include specularity in an analysis or run extra orbits
- Allow for larger model sizes, with element and node label limits now as high as 100M

Q & A?

THANKS!