# Status of Recent Nodal Kinetics Upgrades in RELAP5-3D

2014 International RELAP5 Users Group Meeting

Idaho Falls, Idaho

September 11-12, 2014

Doug Barber
Information Systems Laboratories, Inc.

# Presentation Outline

- RELAP5-3D Nodal Kinetics Upgrades
  - Asynchronous Nodal Advancement
    - Purpose:  Provide automatic time step control for the nodal kinetics.
  - Parallel Processing with Domain Decomp
    - Purpose:  Resurrect parallel domain decomposition logic for the nodal kinetics.

# Asynchronous Nodal Advancement

- Purpose:
  - Nodal kinetics solution can be computationally intensive
  - Need dynamic time step control
    - Use small time steps when conditions are changing
    - Use large time steps when conditions are quasi-steady
  - Implement automatic time step prediction based on change in:
    - Absorption + removal cross section
    - Neutron flux
- Status: Completed

# Asynchronous Nodal Advancement

$$\tau_\theta = \frac{1}{\left|\frac{1}{\theta}\frac{\partial \theta}{\partial t}\right|}$$

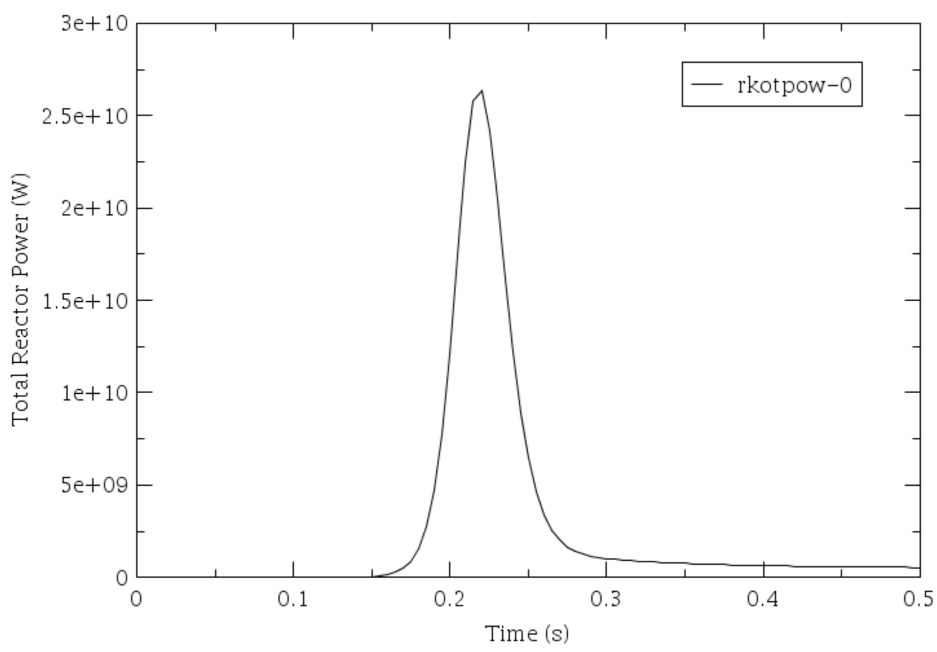
$$\Delta t_\theta^n = \eta_{dyn}\left[\min \tau_{\theta,i,j}^n\right]$$

- Approach:
  - Use dynamic time scale[1]
  - Determine the linear rate of change
- Apply user-defined fractional allowable change
  - Ratio of kinetics and T/H time step size is restricted to be a rational number
  - Take the minimum linear rate of change across all nodes and all parameters (cross section and flux)
  - Option for using extrapolation when kinetics is supercycling T/H
  - Potentially unstable
    - Synchronization will be key
    - User input min/max kinetics time step size will help

[1] Pope, Michael A. and Mousseau, Vincent A., "Accuracy and Efficiency of a Coupled Neutronics and Thermal Hydraulics Model," Nuclear Engineering and Technology, Vol. 47 No. 7, September 2009.

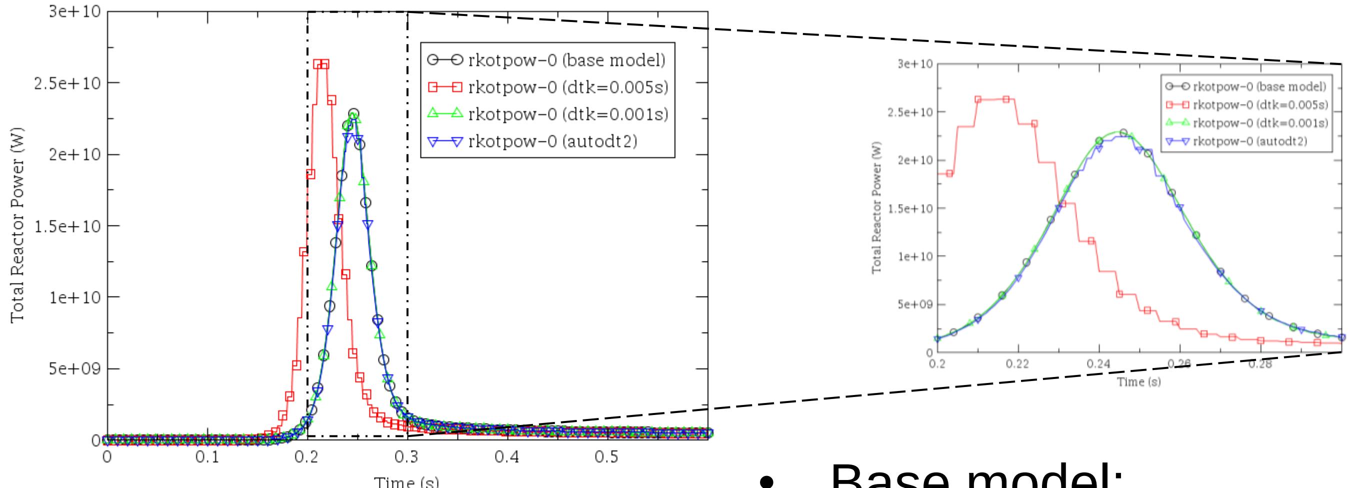
# Asynchronous Nodal Advancement

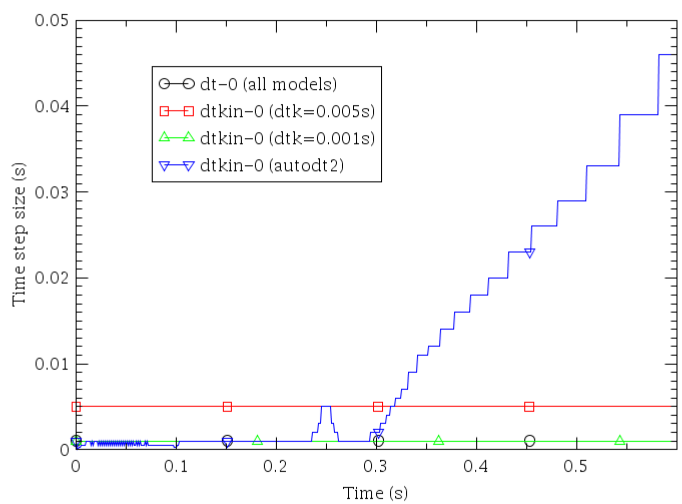
- <u>NEACRP C1 Rod Eject Benchmark</u>
  - Peripheral Rod Ejection
  - 0.1s ejection
  - Peak Power @ 0.22s
  - Time step analysis
    - Expected dt=0.001s during ejection
    - dt=0.100s – 0.250s during asymptotic phase



# Asynchronous Nodal Advancement

- <u>NEACRP C1 Rod Eject Benchmark</u>

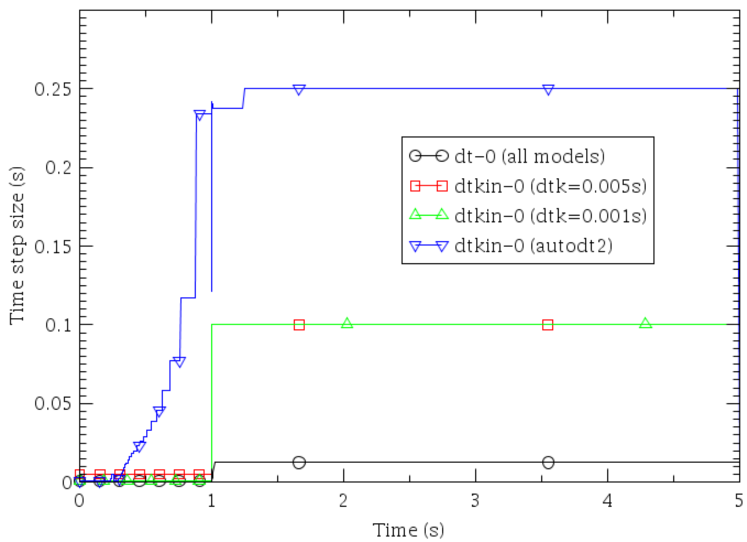


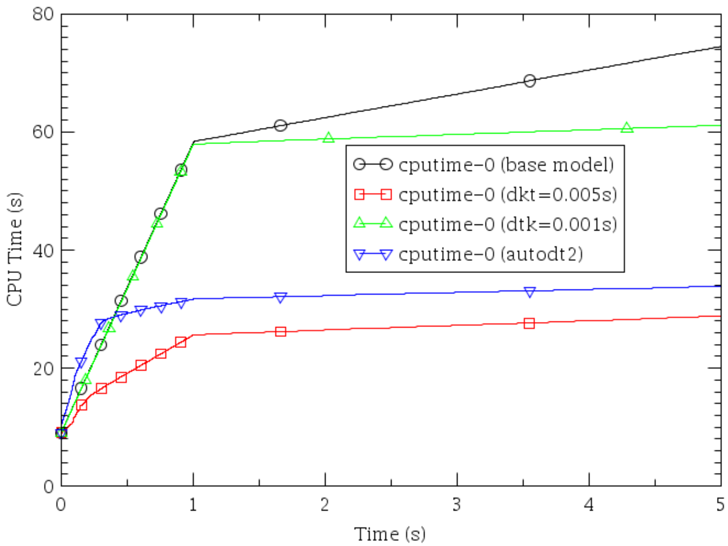




- Base model:
  - dt = 0.001 sec for first 1.0 sec
  - dt = 0.1 sec for remainder
- Auto time step control model:
  - dtkmin = 1.0E-7 sec
  - dtkmax = 0.250 sec
  - No flux extrapolation (results in “stair-stepping”)

# Asynchronous Nodal Advancement

- <u>NEACRP C1 Rod Eject Benchmark</u>




- Time step size prediction reaches 0.250 s (kinetics max.) at 1.2 s
- A slight CPU increase observed up to about 0.4 s.
  - Due to predicted time step size less than 0.001 s during initial phase
  - Kinetics minimum time step size was 1.0E-7 s.
- Roughly 50% reduction in CPU time over the entire transient simulation

# Parallel Processing

- Purpose:
  - Nodal kinetics solution is largely parallelizable.
    - This work was done 15 years ago, but coding has not been maintained
  - Resurrect parallel processing logic for the nodal kinetics.
  - Utilize axial domain decomposition.
  - Maximum of 4 axial subdomains solved in parallel.
  - Expect near 100% efficiency for 2 processors and slightly less for 4 processors.
- Status: Completed

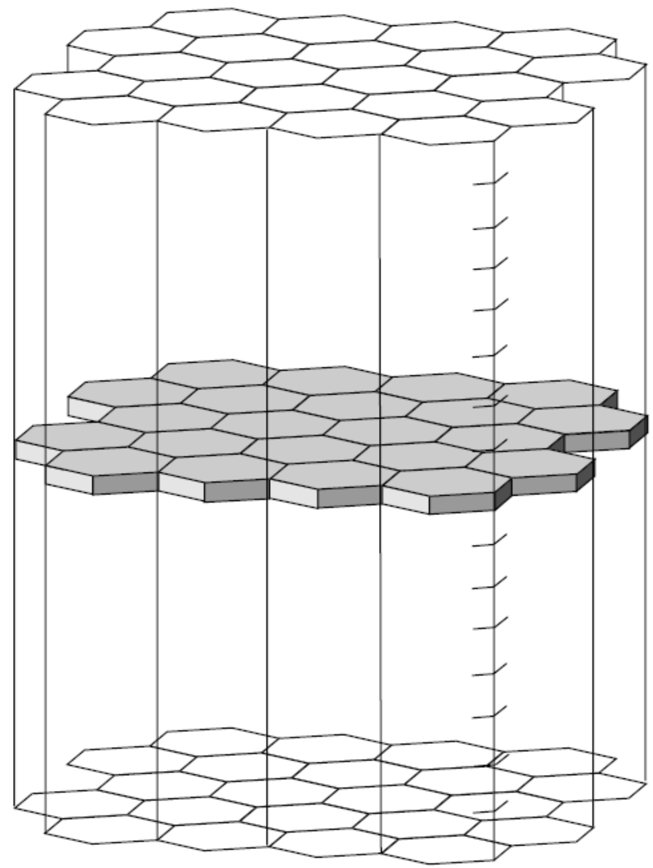
# Parallel Processing

- Parallel Coarse Mesh Finite Difference (CMFD)
  - Requires extra solution at the interface
  - Incomplete Domain Decomposition (IDD) Preconditioner is utilized
  - Near 100% efficiency is possible
- Parallel Nonlinear Nodal Solver
  - Two-node solutions are perfectly parallelizable
  - Super-speedups are expected since memory fetch times are reduced (more on-chip storage per domain)
- Support Calculations
  - e.g., cross section evaluation, linear system setup, etc.
  - Inherently parallel
  - Should see 100% efficiency

# Parallel Processing

- Diagonal Incomplete Domain Decomposition (DIDD)

$$\begin{bmatrix} P_1 & U \\ L & P_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (x_1, x_2) \in \mathbb{R}^{N/2}$$



- Costly to invert, so approx.
  - Solve 2-plane linear system
  - Only use block diagonals

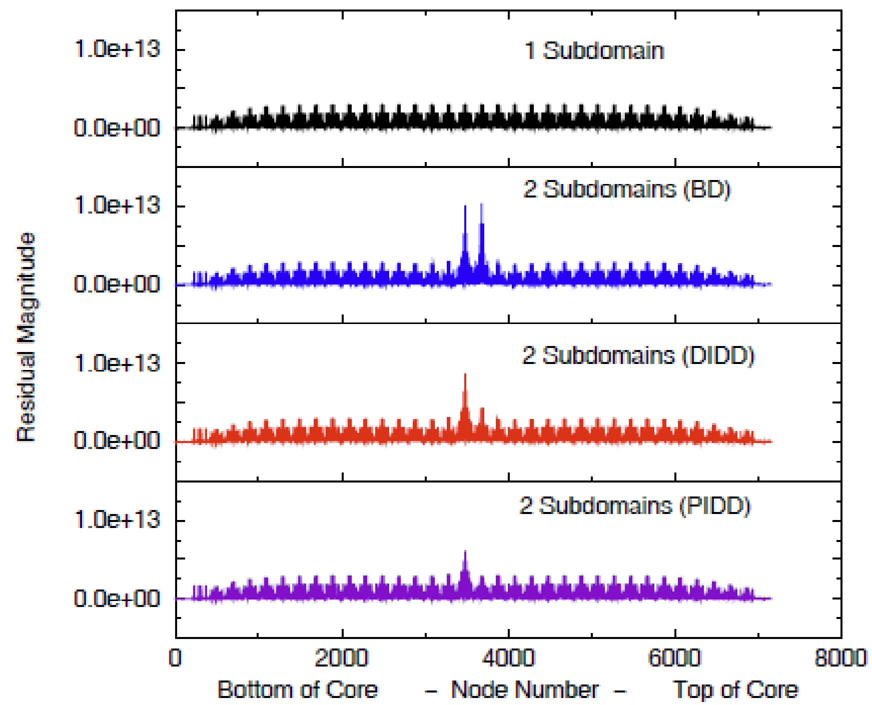
$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} \Gamma_1^{-1} & \\ & \Gamma_2^{-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (\tilde{x}_1, \tilde{x}_2) \in \mathbb{R}^{N/k}$$

- Substituting yields:

$$P = \begin{bmatrix} I_1 & & & & & \\ & \ddots & & & & \\ & & I_{k/2} & -U_{k/2}\Gamma_2^{-1} & & \\ & & -L_{k/2+1}\Gamma_1^{-1} & I_{k/2+1} & & \\ & & & & \ddots & \\ & & & & & I_k \end{bmatrix}^{-1} \begin{bmatrix} P_1 & U_1 & & & & & & \\ L_2 & \ddots & \ddots & & & & & \\ & \ddots & \ddots & U_{k/2-1} & & & & \\ & & L_{k/2} & P_{k/2} & & & & \\ & & & & P_{k/2+1} & U_{k/2+1} & & \\ & & & & L_{k/2+2} & \ddots & \ddots & \\ & & & & & \ddots & \ddots & U_{k-1} \\ & & & & & & L_k & P_k \end{bmatrix}$$

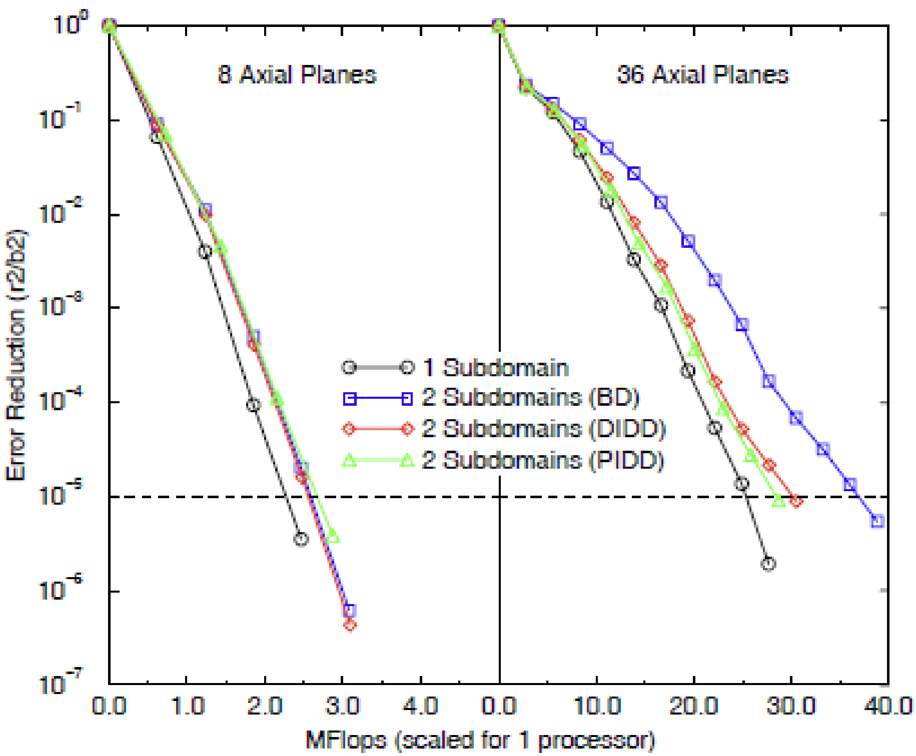
# Parallel Processing

- Comparison of 3 IDD Preconditioners
  - Block Diagonal (BD)
  - Diagonal IDD (DIDD)
  - Planar IDD (PIDD)



# Parallel Processing

- Comparison of 3 IDD Preconditioners




- PIDD generally shows the best residual performance
- DIDD has similar performance but is easier to construct

# Parallel Processing

- Critical Characteristics (Expectations)
  - Solution Results
    - Identical results are obtained compared to the base code when the entire solution, except for the coarse-mesh finite difference (CMFD) solver, is run in parallel mode.
    - Identical results are obtained when the nonlinear nodal solver (NEM, TPEN) is run in parallel and serial mode.
    - Differences in the CMFD solution for single-threaded and multi-threaded runs should be relatively small.
    - Using 2 CPUs and 4 CPUs, each test case should generate five identical results when run five times.
  - Solution Performance
    - Speedup efficiencies of near 100% are obtained for just the nonlinear nodal solver (NEM, TPEN).
    - Speedup efficiencies of near 100% are obtained for all other nodal kinetics functions, except the CMFD solver.
    - Speedup efficiencies greater than 75% on 2 CPUs and 50% on 4 CPUs are obtained for the CMFD solver.

# Parallel Processing

- Verification Testing
  - Following runs were made:
    1. Base code
    2. Modified code (regression mode), 1 thread
    3. Modified code (regression mode), 2 threads, 5 times
    4. Modified code (regression mode), 4 threads, 5 times
    5. Modified code (standard mode), 1 thread
    6. Modified code (standard mode), 2 threads, 5 times
    7. Modified code (standard mode), 4 threads, 5 times
  - Expectations:
    - Runs 1, 2 and 5 should be identical except for TPEN
    - Runs 3 and 4 should be identical to 2
    - Runs 6 and 7 should be different compared to Run 5
    - All 5 executions for Runs 6 & 7 should be identical

# Parallel Processing

- Verification Testing (Solution Results)
  - Regression Testing
    - Single-threaded results for base code and modified code (both regression and standard mode) were identical for most cases.
      - Exception was for cases that used the TPEN solver, which was expected
    - 2-threaded results with the modified code (regression mode) were identical to single-threaded results
      - All 5 executions were identical
    - 4-threaded results with the modified code (regression mode) were identical to single-threaded results
      - All 5 executions were identical
    - 2- and 4-threaded results with the modified code (standard mode) were different compared to single-threaded results
      - Differences were small and expected, but the accuracy of the solution was not impacted

# Parallel Processing

- Verification Testing (Performance Results)
  - Effective efficiency for Krylov Solver (%)

| NK Case | 1 Thread | 2 Thr (avg) | 4 Thr (avg) | 4 Thr (max) |
|---|---|---|---|---|
| neacrp-c1-4node-krlv-nem[1] | 117.4 | 99.3 | 78.4 | 82.1 |
| smart330-c1g4-tr-krlv-cmfd[1] | 140.1 | 96.3 | 50.6 | 84.5 |
| smart330-c1g4-tr-krlv-nem[1] | 128.2 | 99.7 | 57.7 | 75.9 |
| smart330-c1g4-tr-krlv-tpen[1] | 123.2 | 96.6 | 79.9 | 84.0 |
| vver440-tr-hzp-krlv-cmfd | 104.6 | 100.3 | 83.7 | 98.9 |
| vver440-tr-hzp-krlv-nem | 100.3 | 100.5 | 93.5 | 98.4 |
| vver440-tr-hzp-krlv-tpen | 102.7 | 98.0 | 95.8 | 98.9 |

[1] Load imbalance on 4 threads

# Parallel Processing

- Verification Testing (Performance Results)
  - Effective efficiency for Nonlinear Nodal Solver (%)

| NK Case | 1 Thread | 2 Thr (avg) | 4 Thr (avg) | 4 Thr (max) |
|---|---|---|---|---|
| neacrp-c1-4node-krlv-nem[1] | 97.7 | 95.6 | 77.0 | 83.0 |
| smart330-c1g4-tr-krlv-cmfd[1] | N/A | | | |
| smart330-c1g4-tr-krlv-nem[1] | 113.6 | 96.1 | 81.2 | 84.0 |
| smart330-c1g4-tr-krlv-tpen[1] | 91.3 | 99.8 | 84.6 | 86.7 |
| vver440-tr-hzp-krlv-cmfd | N/A | | | |
| vver440-tr-hzp-krlv-nem | 97.3 | 94.9 | 90.8 | 92.6 |
| vver440-tr-hzp-krlv-tpen | 88.5 | 98.5 | 93.5 | 95.8 |

[1] Load imbalance on 4 threads

# Parallel Processing

- Verification Testing (Performance Results)
  - Overall “Realized” Performance Efficiency (%)

| NK Case | 1 Thread | 2 Thr (avg) | 4 Thr (avg) | 4 Thr (max) |
|---|---|---|---|---|
| neacrp-c1-4node-krlv-nem[1] | 105.9 | 97.7 | 68.5 | 71.2 |
| smart330-c1g4-tr-krlv-cmfd[1] | 117.0 | 105.9 | 57.1 | 80.7 |
| smart330-c1g4-tr-krlv-nem[1] | 118.5 | 112.5 | 64.4 | 69.6 |
| smart330-c1g4-tr-krlv-tpen[1] | 168.8 | 164.8 | 124.8 | 127.9 |
| vver440-tr-hzp-krlv-cmfd | 104.3 | 94.7 | 68.0 | 78.0 |
| vver440-tr-hzp-krlv-nem | 101.0 | 92.9 | 80.0 | 82.9 |
| vver440-tr-hzp-krlv-tpen | 94.3 | 88.2 | 78.7 | 81.1 |

[1] Load imbalance on 4 threads

# Parallel Processing

- Summary
  - Entire Nodal Kinetics solver has been parallelized (for Krylov)
  - Regression testing yielded expected results
  - Multiple multi-threaded runs showed no variability in results
  - Parallel performance was better than expected
    - Parallel efficiency for Krylov solver on 2 and 4 threads was over 95% for cases with load balance
    - Parallel efficiency for NEM solver on 2 and 4 threads was over 95% for cases with load balance
    - Parallel efficiency for TPEN solver on 2 and 4 threads was 95% and 90%, respectively
    - TPEN solver display unstable error reduction
  - Overall performance improvement was very good
    - Between 90% and 100% on 2 threads
    - Greater than 80% on 4 threads (CPU utilization issues)

# Summary

- Automatic Nodal Kinetics Time Step Control
  - Completed December 2013
- Parallel Nodal Kinetics (Krylov-based)
  - Completed June 2014
- Both updates will be included in a post-4.2.1 version