Advancing the Adoption of High Performance Computing for Time Domain Simulation

Liang Min and Steve Smith
Lawrence Livermore National Laboratory (LLNL)

Team: Carol Woodward, Steve Smith and Liang Min, LLNL
Siddhartha Khaitan and James McCalley, Iowa State University

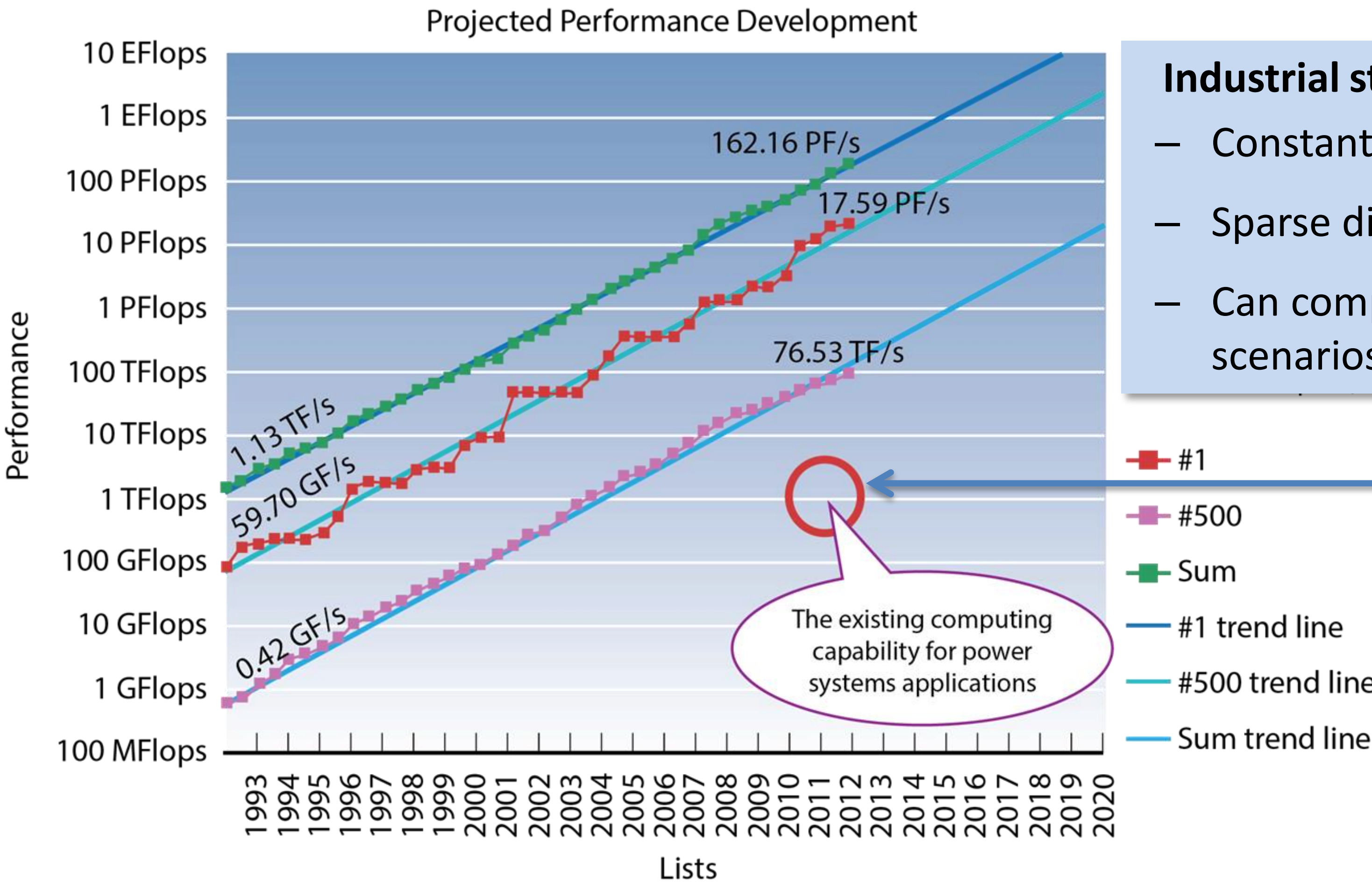
June 18th, 2014

Lawrence Livermore National Laboratory

This work was supported by the Office of Electricity Delivery and Energy Reliability of the U.S. Department of Energy. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC.

LLNL-PRES-655706

Electric power industry needs software applications that ride the HPC technology curve



Industrial state of the art

- Constant time steps
- Sparse direct solvers
- Can complete ~1000 scenarios in a study

Dell XPS 8500
Intel Core 3rd Gen i7
~130 GFlops

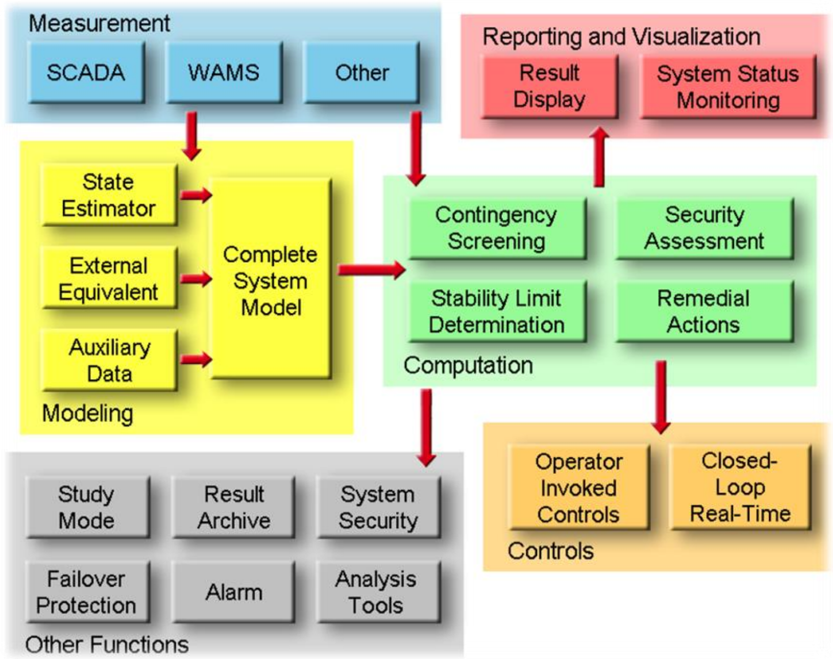
Power system applications lag significantly behind other industries in use of advanced computing technologies.

Presentation outline

- Introduction and project purpose
- Significance and impact
- Technical approach
- Technical accomplishments
- Acknowledgements and contact

Introduction

- Time domain transient stability analysis (TSA) is one of the key forms of power system stability analysis.
- TSA is moving toward becoming an on-line control center application (PJM, MISO, ERCOT, CAISO, ISO-NE).
- On-line TSA of large interconnected power is extremely computationally expensive.
- Distributed computing technology has been explored to improve the computational speed.
- Enhancing and extending TSA tools via algorithms and adding new model features are difficult and costly.



Components in PJM's on-line TSA implementation

Project purpose

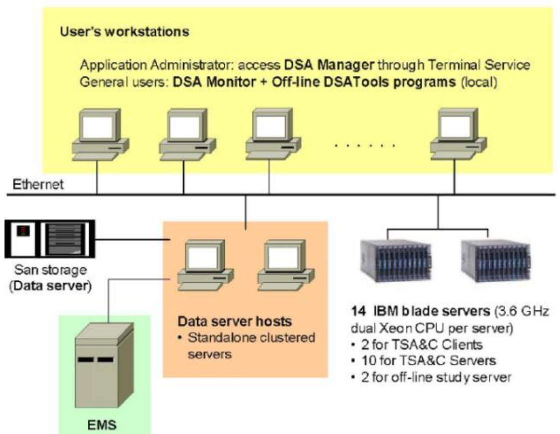
- Online TSA of large interconnected power systems is becoming more computationally expensive:
  - Growing size of power system models
  - Variable generation dynamics
  - Cascading events
- High performance computing (HPC) would be helpful in a real-time mode to solve important problems of high complexity, providing a factor from hundreds to millions times improvements in time-to-solution.
- The objective of this project is to explore advanced computing technologies and applied mathematics to enhance the computational speed of online TSA.

LLNL-PRES-655706

Significance and impact
- The degree of innovation

State-of-the-art

- PJM Interconnection and Powertech Labs have implemented online TSA
  - EMS models with 13,500 buses and 2,500 generators
  - Processing of **3,000 contingencies**
  - Computation cycle within **20 minutes**



- Lack of modularity: computational solvers used in commercial software tools are typically intertwined with the system models

Our innovative approaches

- ***Significantly enhance performance through parallelism and HPC.***
  - PJM EMS test case with 13,029 buses, 431 generators, and12,488 branches.
  - Processing **1 million contingencies**
  - Reduced run time from 138 days to **26 minutes**
  - The performance can be further improved by reducing the I/O bottleneck

- ***Promote the use and development of open-source software***
  - Open-source numerical solver
  - No vendor lock-in
  - Simulating interaction and collaboration
  - Clear separation between Modeler/Solver

*To support operational needs.. perform the simulation **on-line** in **real time***

Significance and impact
- Long term vision and impact

State-of-the-art	Long Term Vision
Bus-branch model; transmission buses only	Node-breaker model; transmission buses + distribution feeders size of the model will increase by 10-10,00 times
10-20 seconds time domain simulation	Long-term time domain simulation > 5 minutes and up to several hours
~1000 contingencies	Massive contingency analysis cascading events, millions of contingencies
15~30 minutes Wall-clock time	Within a dispatch interval < 5 minutes

To capture the dynamics from DER and improve visibility of substation

To capture the multi-scale dynamics (Wind ramping, AGC, boiler dynamics)

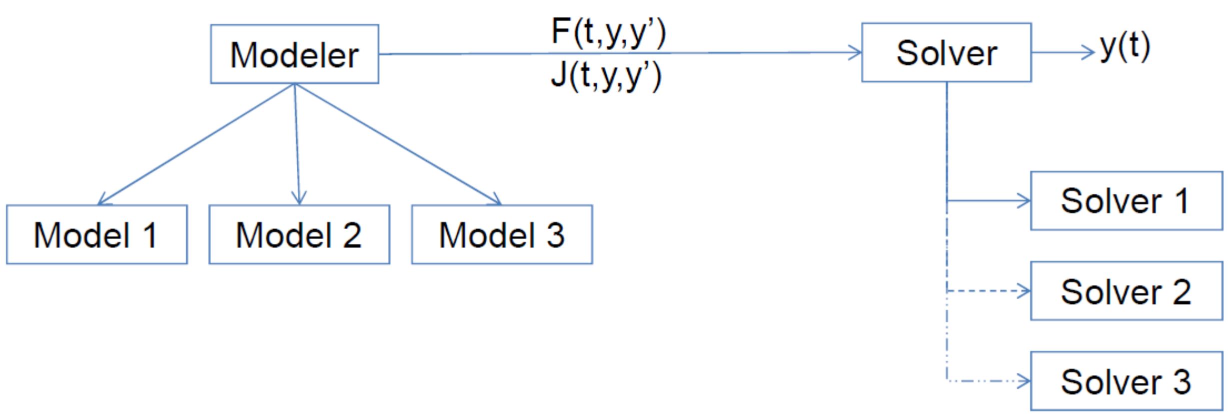
To avoid cascading outages and increase grid resilience to HILP events

To enable faster than real-time simulation and meet operational needs

Improve Both Hardware and Algorithms to Enable Fast and High Fidelity Time Domain Transient Stability Analysis

Technical approach
- Open-source numerical solvers

- Leverage the DOE SciDAC work and utilize open-source numerical solvers for large-scale power system time domain simulation.
  - SUNDIALS – a SUite of Nonlinear and DIfferential-ALgebraic Solvers
- Build and test generic direct sparse solver interface to KLU and SUPERLU_MT from the LLNL SUNDIALS solvers.
- Explore a power system simulator concept
  - Clear separation between Modeler/Solver
  - User defined models and an “as generic as possible” solvers



LLNL's SUNDIALS packages are used in power grid simulators

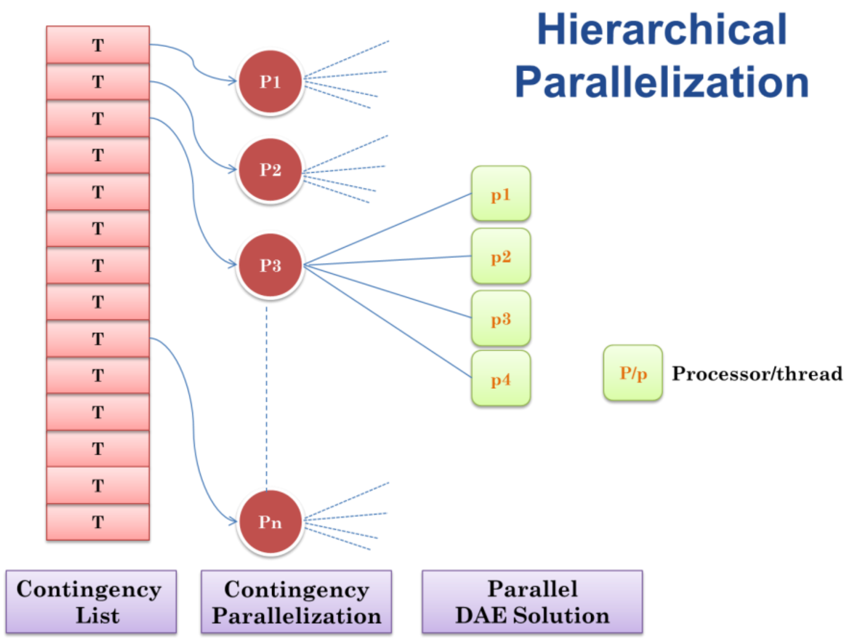
- SUite of Nonlinear and DIfferential-ALgebraic Solvers
  - ODE and DAE time integrators with forward and adjoint sensitivity capabilities, Newton-Krylov nonlinear solver
  - KINSOL - Newton solver, $F(u) = 0$
  - CVODE(S) – implicit ODE solver with sensitivities, $y' = f(y, t, p)$
  - IDA(S) – implicit DAE solver with sensitivities, $F(t, y, y', p) = 0$
  - Supplied with serial, threaded (soon), and MPI parallel structures
- Leverages LLNL expertise in time integration over three decades
- Freely available, released under BSD license (> 3,000 downloads/yr)

Used in ISU's Time Domain Power System Simulator (TDPSS) code and in RTE-France prototype transmission model…Will be included in PNNL's GridPACK Toolkit soon.

https://computation.llnl.gov/casc/sundials/main.html

Technical approach
– Parallelism and high performance computing

- We leverage hierarchical parallelization approach. We implemented two-level parallelism:
  - ***Coarse-grained:*** both distributed and shared memory parallelization over contingencies
  - ***Fine-grained:*** shared memory parallelization within a single contingency



- Software utilized:
  - Iowa State University's Time Domain Power System Simulator (TDPSS)
  - LLNL's SUNDIALS/IDAS solver for the solution of differential-algebraic equations (DAEs)
  - LBNL's SuperLU_MT threaded sparse direct linear solver library

Technical approach
– Parallelism and high performance computing

Coarse-grained parallelism over contingencies

- Shared Memory Parallelization using Pthreads and OpenMP.
  - We utilize two different static schedulers, one uses OpenMP, and the other uses Pthreads.
  - To reduce the overhead of thread creation and destruction; threads are only created once, and then are assigned multiple tasks.
  - Contingencies are distributed on the threads, such that they receive a nearly equal number of tasks.
- Distributed Memory Parallelization using MPI.

LLNL-PRES-655706

Technical approach
– Parallelism and high performance computing

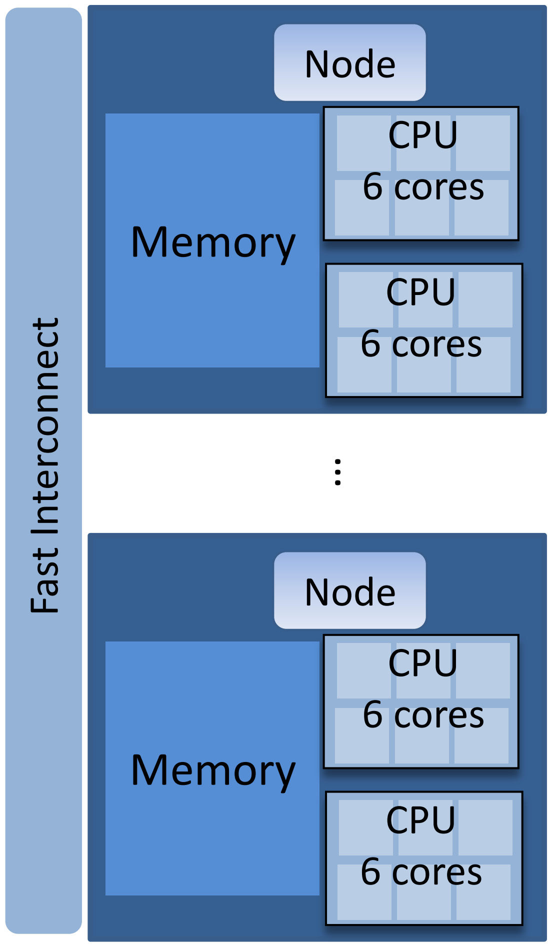
Fine-gained parallelism within a contingency

- We thread-parallelized the vector structure in SUNDIALS.
  - The main data structure holding problem unknowns within the SUNDIALS IDAS time integrator and nonlinear solver is a vector.
  - An OpenMP parallelized version of the IDAS vector kernel was utilized and speedups using this kernel were assessed on multicore machines at LLNL.
  - This kernel was then interfaced with TDPSS and further tests were done to assess whether threading the vector kernel provided benefit.
- We interfaced the SuperLU_MT to SUNDIALS/IDAS for the TDPSS.
  - Developed at LBNL as part of the DOE SciDAC program.
  - Reordering and factorization threaded with Pthreads and OpenMP.
  - The Pthreads version of SuperLU_MT was used in this project.

LLNL-PRES-655706

Technical approach
- HPC machines used in this study

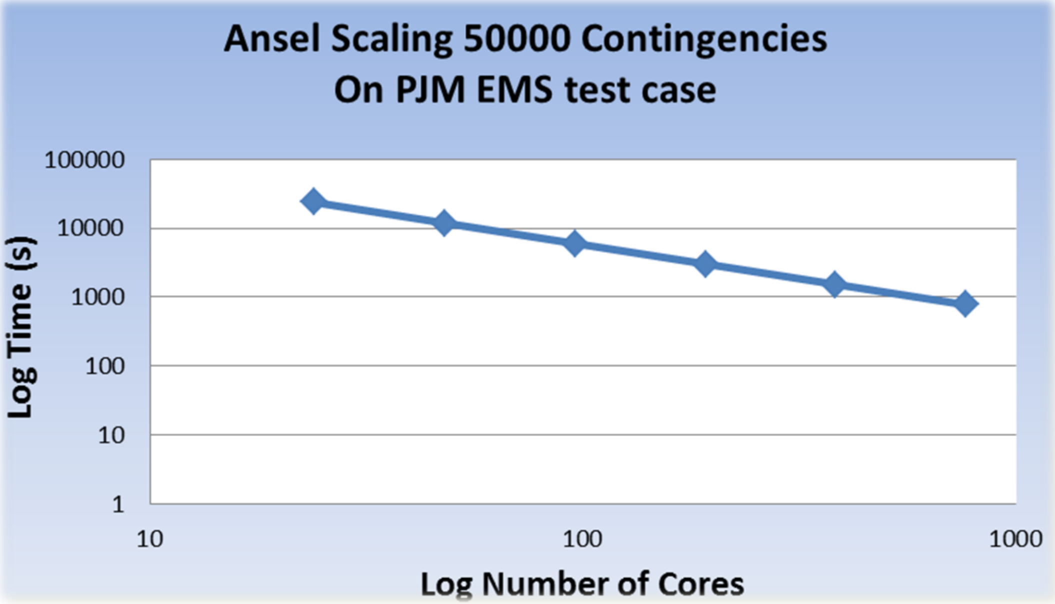
- 3 LLNL machines and 2 ISU machines were used to test thread and MPI performance under varying conditions.
- **ISU Cystorm**
  - 396 dual quad core nodes with AMD processors in12 racks
  - 8GB memory per node
  - Nodes connected with DDR infiniband interconnect
- **LLNL Sierra**: Linux cluster machine configured for large parallel jobs, supported a very large study
  - 1,856 batch nodes with 2 Intel Xeon EP X5660 CPU’s with 6 cores each
  - Each node has 24 GB of memory
  - Nodes are connected with an InfiniBand QDR high-speed interconnect



Technical accomplishments - Parallelization over contingencies results

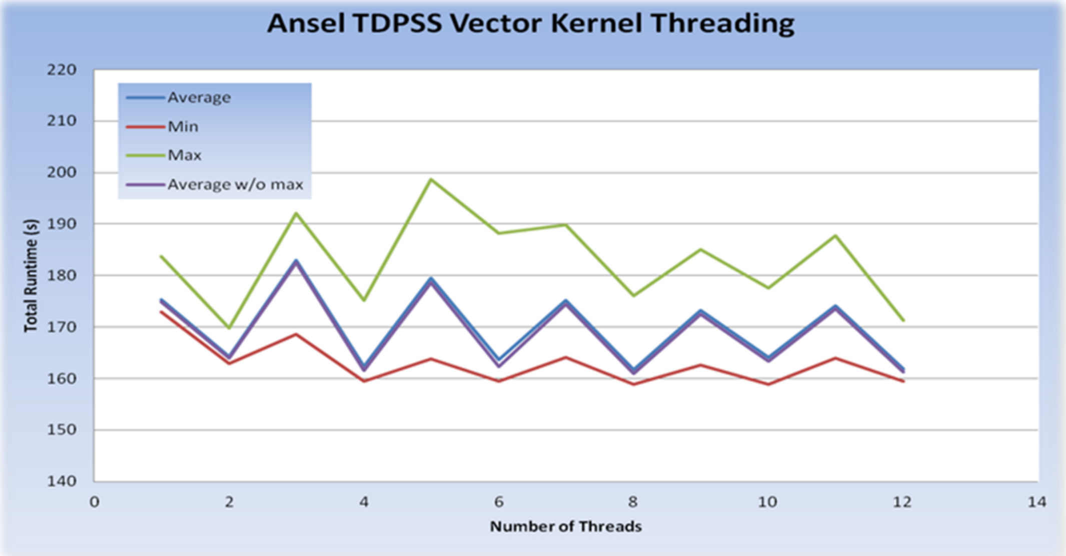
- Explored use of MPI, OpenMP and Pthreads to run contingencies in parallel.
- Results for MPI are shown using the 24 core run as baseline for speedup. Obtained near ideal speedups out to 768 cores.

Number of Cores	Time (s)
24	23826
48	11842
96	5922
192	2978
384	1514
768	784



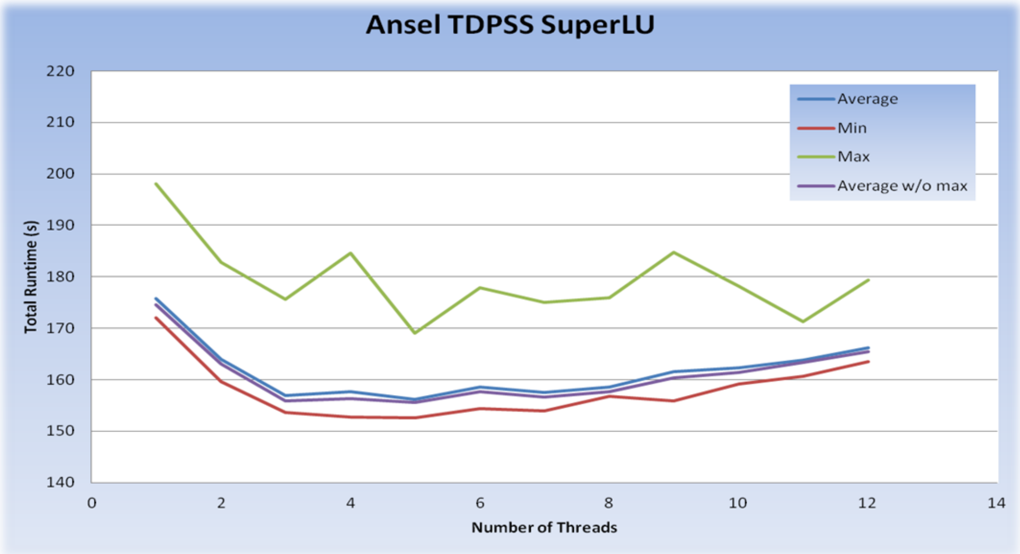
Ran 1 million contingencies test on 12,228 cores at LLNL's Sierra machine: reduced run time from 138 days (sequential) to 26 minutes (parallel)

Technical accomplishments - Single contingency vector kernel parallelism results



- Applied OpenMP to the vector kernel for the SUNDIALS IDAS solver.
- Tests show little benefit from threading the kernel.
- Based on profiling results, not enough time is spent in the vector kernel routines to result in significant gain from this parallelization strategy for problems sizes tested.

Technical accomplishments - Single contingency linear solver parallelization with SuperLU_MT results

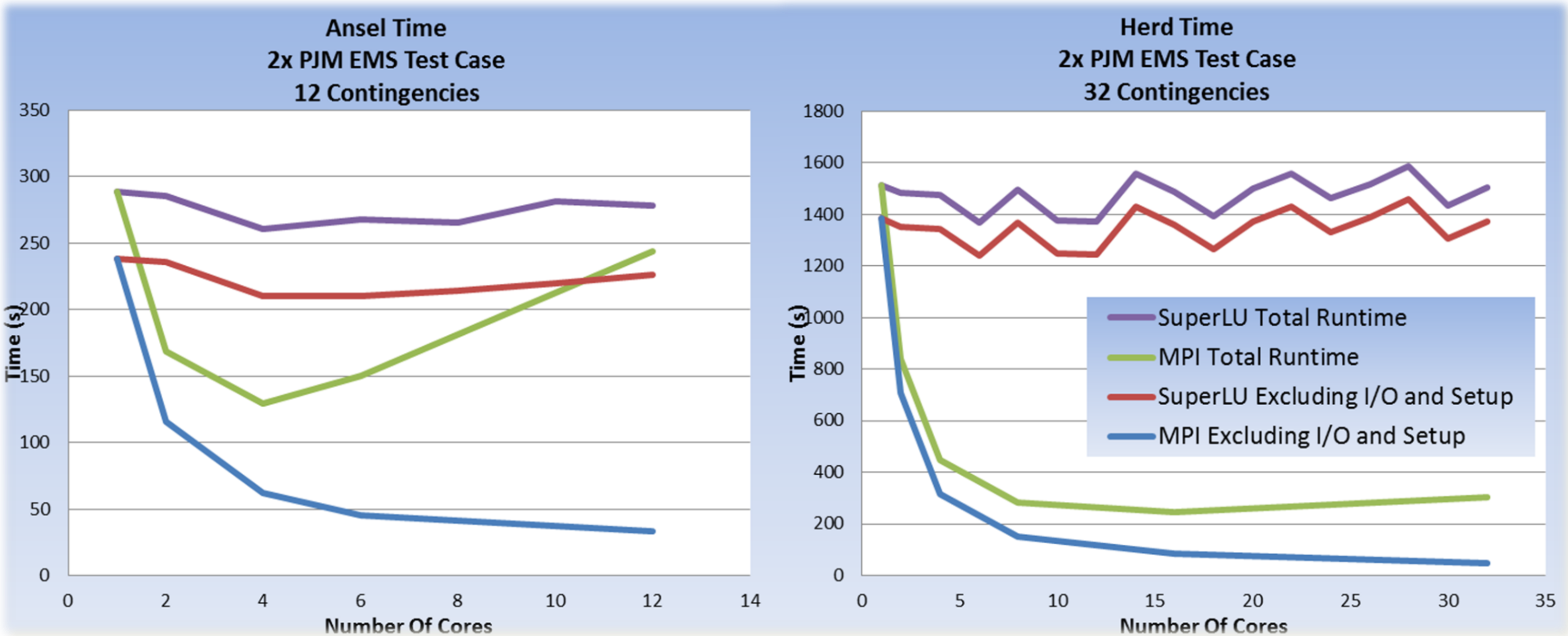


- Results are show a benefit on 2-4 threads.
- Overhead with creating and spawning > 4 threads overwhelms any benefit from the extra parallelism.
- Expect a bigger problem would benefit more from the parallelism.

LLNL-PRES-655706

Technical accomplishments - Contingency study throughput results

- The main goal was to assess whether throughput of a contingency study could be improved by running fewer full contingencies at a time using a shared memory space and thread parallelization of each contingency.



- Running multiple contingencies in parallel was observed to be more effective than parallelizing over a single contingency through SuperLU_MT.
- Significant I/O bottlenecks were observed.

Conclusion

- Parallelization over contingencies consistently yielded excellent speedups.
- Fastest throughput was always achieved parallelizing across MPI tasks without threading.
- Problem sizes tested were relatively small and threading potential was limited.
- Future work includes investigation of full parallelization of each contingency run and distributed memory iterative methods such as Krylov methods.

Acknowledgements and contacts

The authors gratefully acknowledge the guidance and support from Dr. Jianzhong Tong PJM and thank the Department of Energy’s Advanced Grid Modeling (AGM) program for the funding support.

Technical contact:
Liang Min
Lawrence Livermore National Laboratory
Phone: 1-925-422-1187
Email: min2@llnl.gov

LLNL-PRES-655706