A Distributed Approach to Large Scale Security Constrained Unit Commitment Problem

Kaan Egilmez
Cambridge Energy Solutions

FERC Technical Conference on Increasing Real-Time and Day-Ahead Market Efficiency through Improved Software

June 22-24, 2015
Washington, DC

About CES

- Cambridge Energy Solutions is a software company with a mission to develop software tools for participants in deregulated electric power markets.
- CES-US provides information and tools to assist market participants in analyzing the electricity markets on a locational basis, forecast and value transmission congestion, and to understand the fundamental drivers of short- and long-term prices.
- CES-US staff are experts on market structures in the US, system operation and related information technology.

Presentation overview

- The convergence of machine virtualization and the maturing of multi-core computing has had a dramatic impact on the ease with which high performance computing techniques can be brought to bear on real world problems.
- At CES we are actively working on improving the performance of our DAYZER market modeling and simulation software by making use of multi-core parallel programming on individual compute nodes combined with distribution of work load across multiple such compute nodes organized into high performance computing clusters.
- This talk provides an overview of the techniques we are using to accomplish this goal as well as simulation results of performance improvement on both small and large scale models such as our combined model for PJM and MISO.
- These techniques if applied to market operations and planning would allow many more scenarios to be concurrently examined and/or more detailed individual models to be solved within reasonable time limits allowing novel solutions to existing concerns regarding robustness of market results to various kinds of uncertainties.

DAYZER

CES has developed DAYZER to assist electric power market participants in analyzing the locational market clearing prices and the associated transmission congestion costs in competitive electricity markets. This tool simulates the operation of the electricity markets by mimicking the dispatch procedures used by the corresponding independent system operators (ISOs), and replicates the calculations made by the ISOs in solving for the security-constrained, least-cost unit commitment and dispatch in the Day-Ahead markets. Models are available for the CAISO, ERCOT, MISO, NEPOOL, NYISO, ONTARIO, PJM, SPP and WECC markets, as well as a combined model for the PJM-MISO region.

DAYZER SCUC MILP (MUC) Formulation

Minimize the total cost over 24 hours of:

Generation + Startup/Shutdown + Imports/Exports + Generation Slacks + Spin Reserve Slacks + Non Spin Reserve Slacks + Transmission Overloads + PAR Angle Overloads

Subject to the following constraints for each hour:

- System energy balance
- Spin reserves requirement
- Non spin reserves requirement
- Unit commitment constraints (capacity, min up/down, start/stop, ramping)
- Pump storage constraints (efficiency, reservoir)
- Transmission constraints (line, contingency, interface, PAR, nomogram)
- PAR angle constraints
- DC line constraints

Examples of DAYZER Model Characteristics

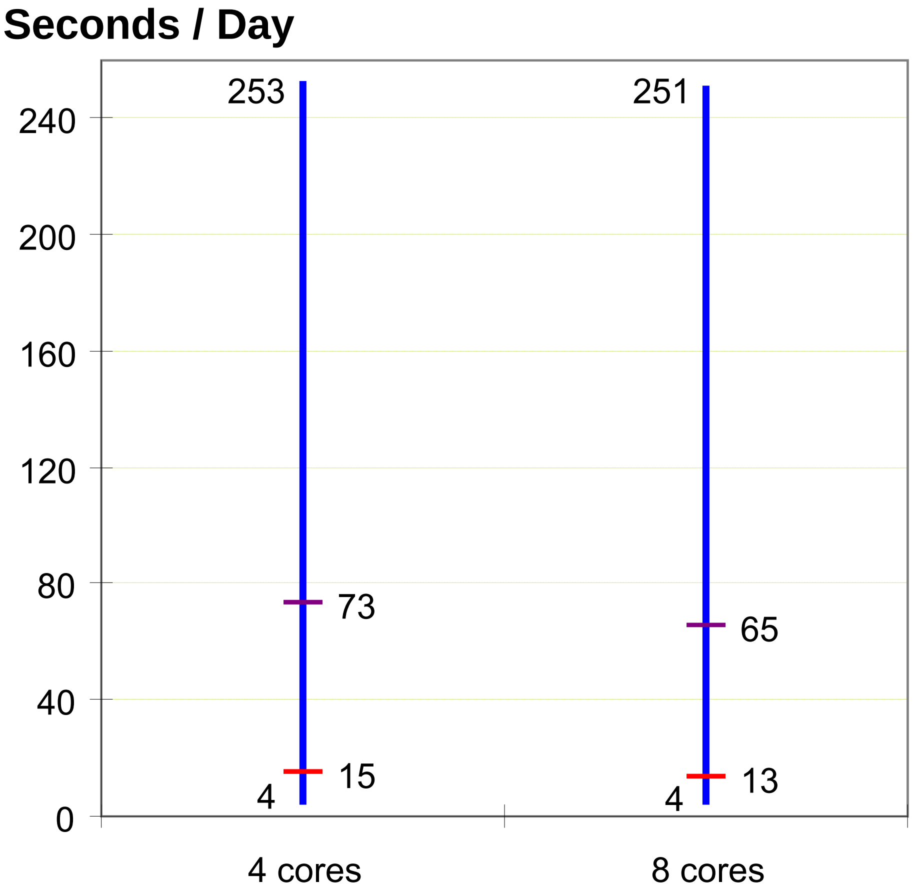
NEPOOL (2014)

8 load zones
1 reserves pool
6 import/export interface units
2 pumped storage units
416 generation units (Nuclear, Hydro, Wind, Solar, CC, ST, GT)
2612 transmission constraints
11 PARs

PJM+MISO combined (2014)

54 load zones + 88 industrial load units
7 reserves pools
39 import/export interface units
8 pumped storage units
1972 generation units (Nuclear, Hydro, Wind, Solar, Battery, CC, ST, GT)
16161 transmission constraints
37 PARs
5 DC Lines

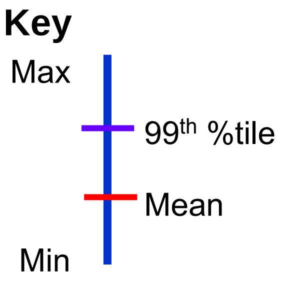
MUC Performance for NEPOOL Model



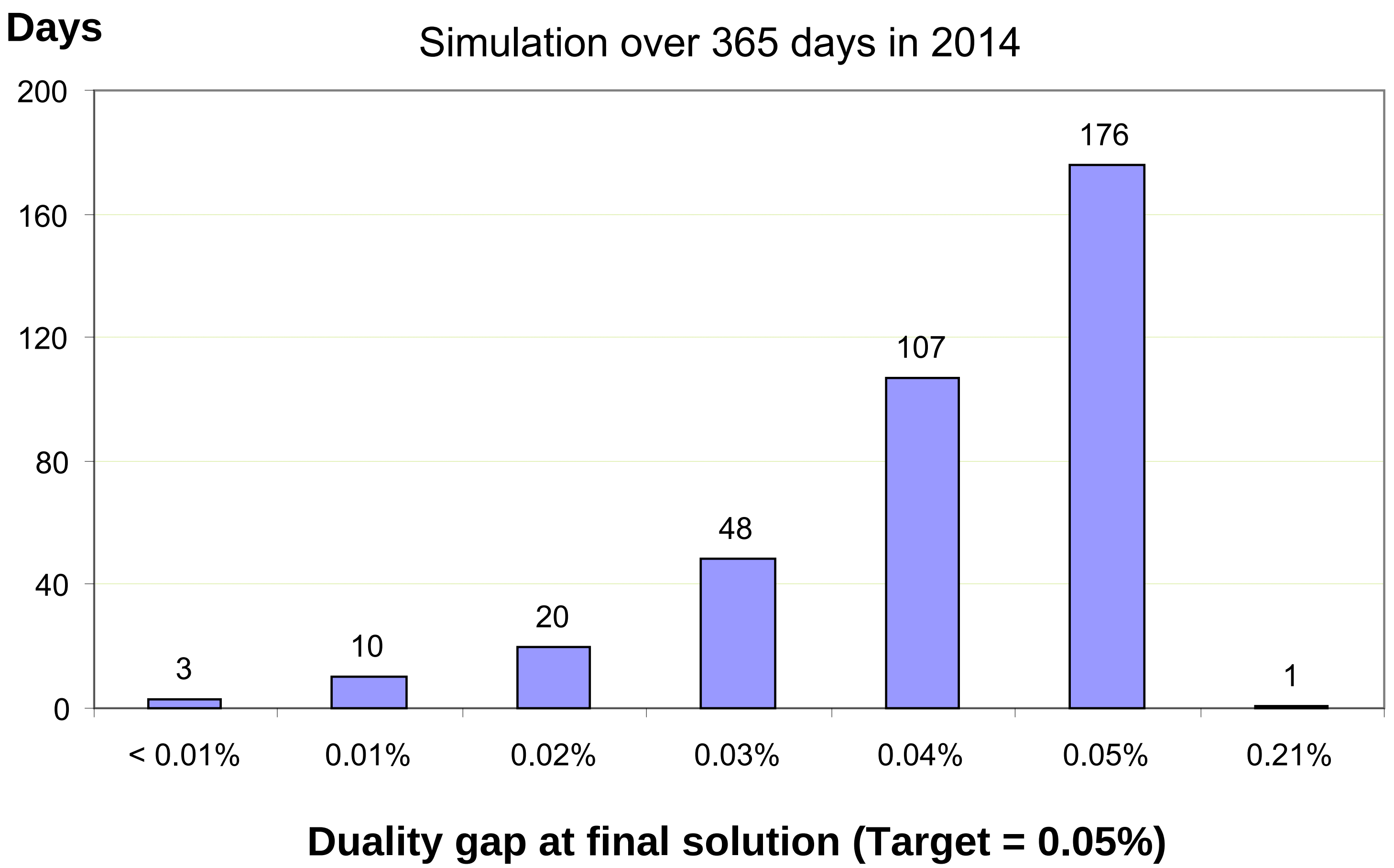
Run Time statistics over 365 days in 2014

Machine A – 4 cores
E3-1240 V2 CPU @ 3.4 GHz
32 GB memory
Windows 8 server 64 bit

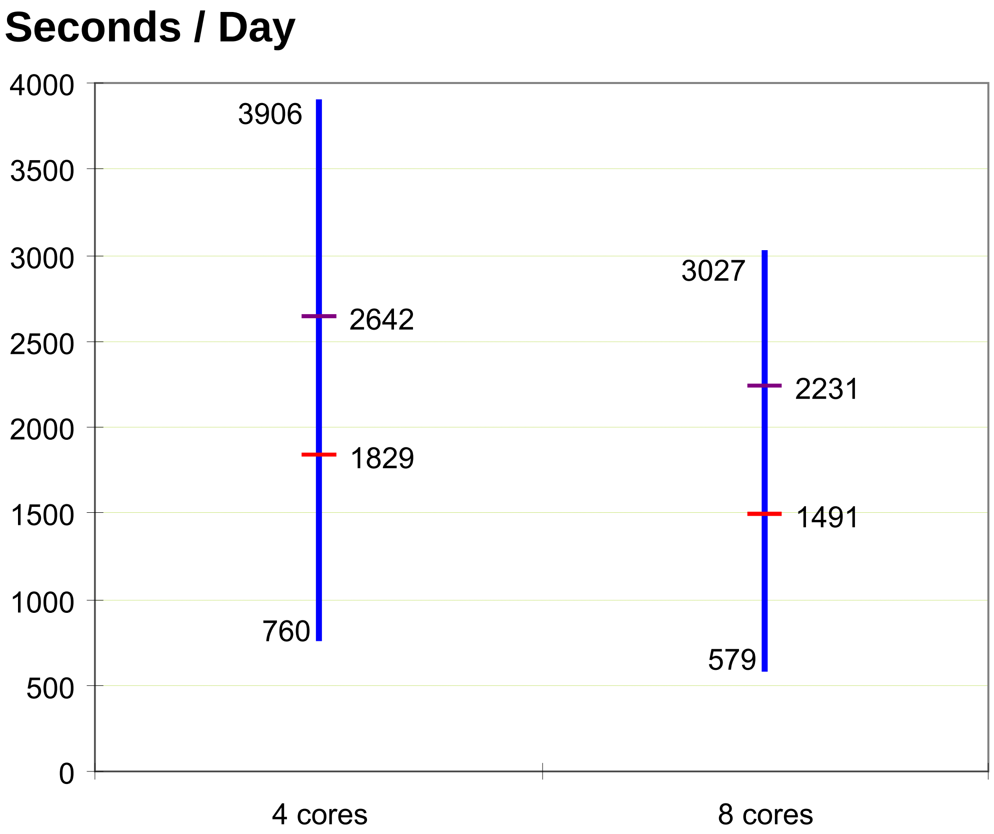
Machine B – 8 cores
i7-5960X CPU @ 3 GHz
(over clocked at 3.87 GHz)
32 GB memory
Windows 8.1 Pro 64 bit



MUC Solution Quality for NEPOOL Model

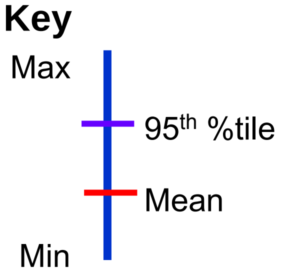


MUC Performance for PJM+MISO Model

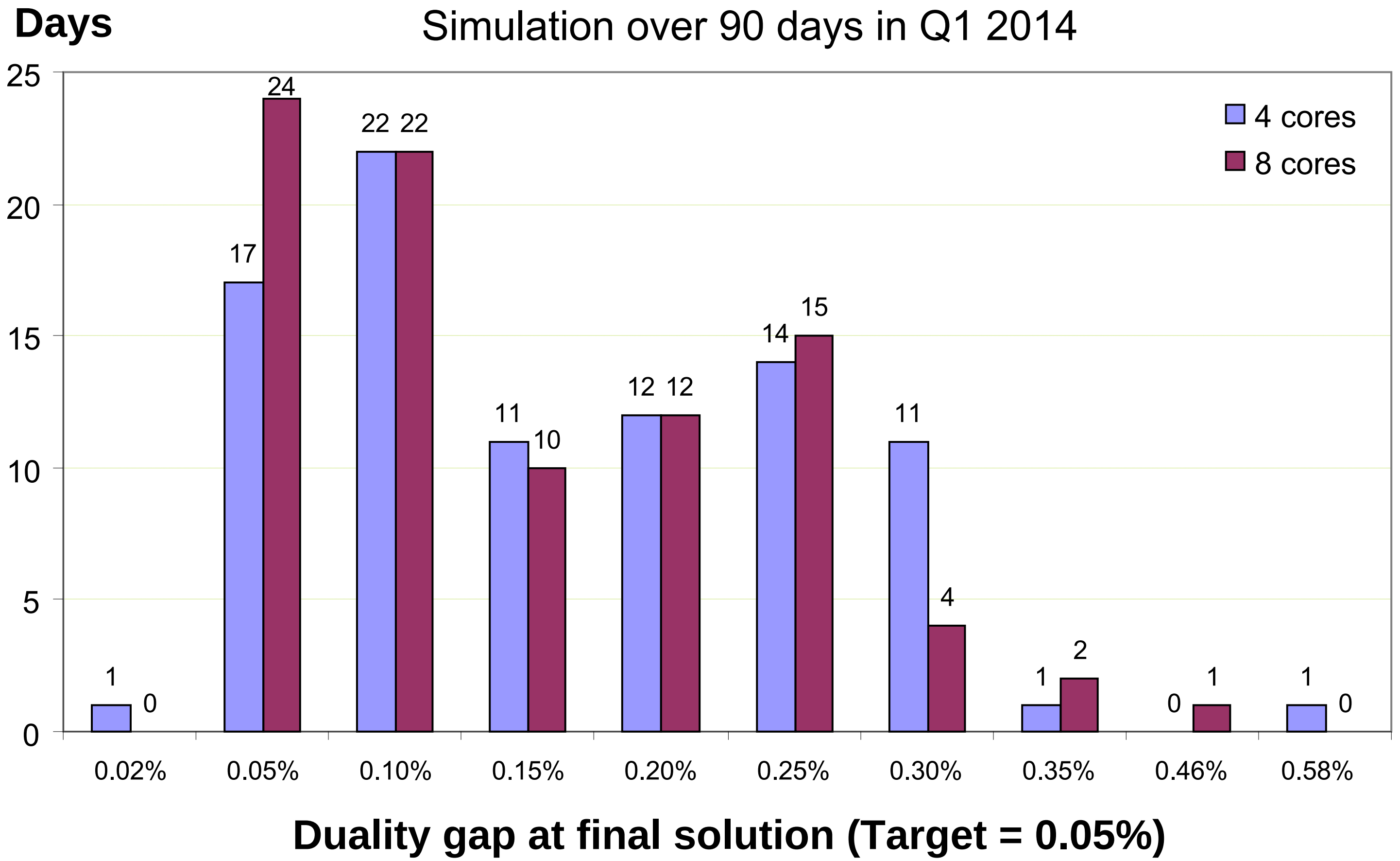


Run Time statistics over 90 days in Q1 2014

The difference in run time performance is due to the faster CPU speed on the 8 core machine. A single MUC process cannot take advantage of multiple cores other than in incidental ways due to I/O and the presence of other workloads. These runs were performed with no other non system tasks running concurrently with MUC.

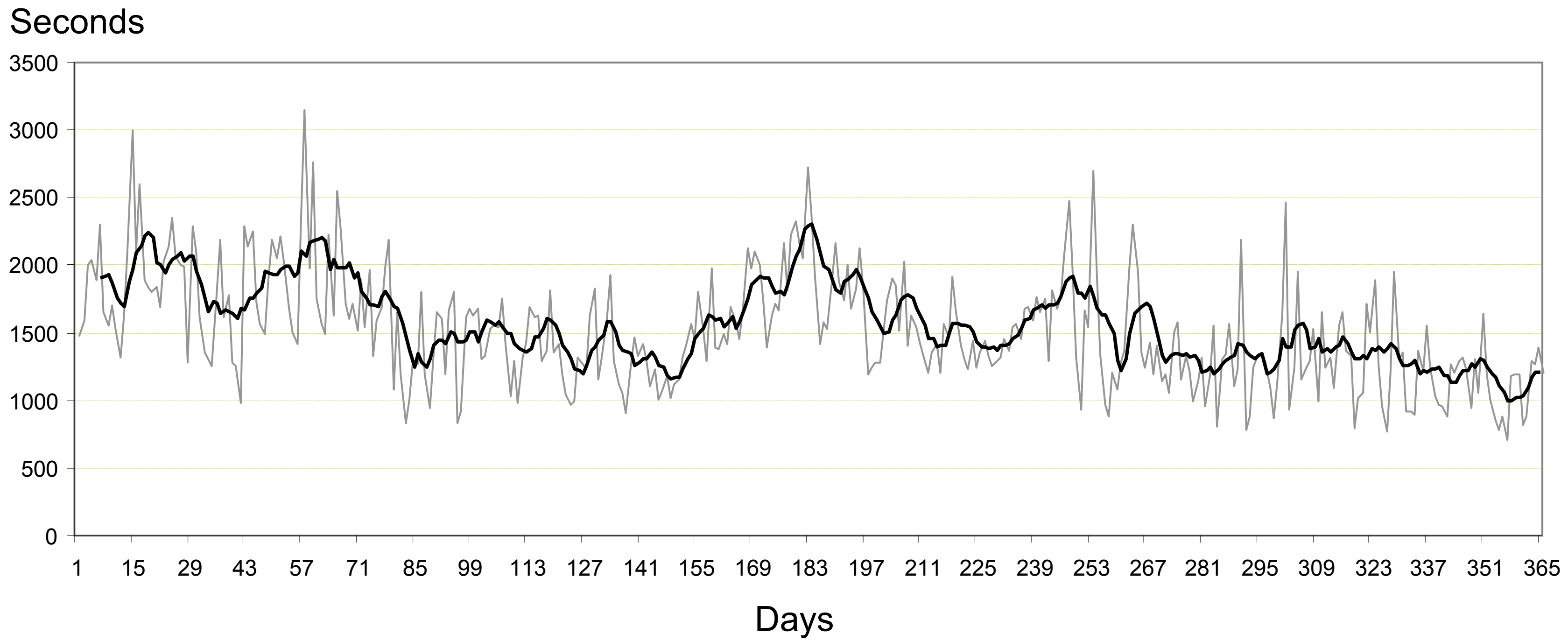


MUC Solution Quality for PJM+MISO Model



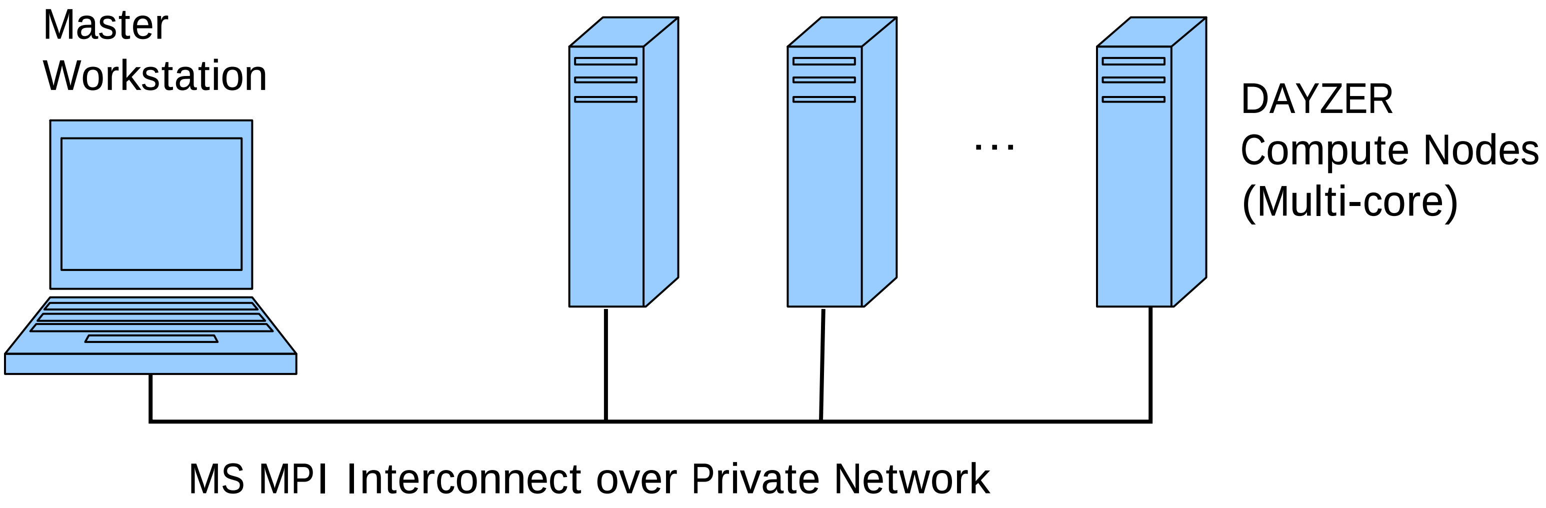
More MILP iterations were able to reach the target duality gap on the faster machine within the allowed maximum run time. The solver termination state (optimal vs. best found) differs for 18 days.

Typical MUC run time performance for a large model simulated for one year



Splitting the simulation into months or quarters and running each segment in parallel is the conventional approach to taking advantage of multi-core machines. It’s clear from the above timing pattern that a finer grained load balancing scheme can produce a much better overall run time performance.

Solution Architecture for Distributed And Parallel DAYZER


Master Workstation
DAYZER Compute Nodes (Multi-core)
...
MS MPI Interconnect over Private Network

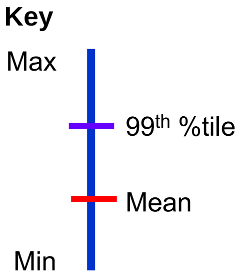
- Simulation period load balanced across all cores at compute nodes using MPI.
- Results can be sent to a central database or stored in local partial databases.
- MPI based query tool allows locally stored results to be aggregated at Master.
- MUC: each day assigned to a core at a node using single threaded MILP SCUC.
- PUC: each day assigned to a multi-core node using Parallel SCUC.

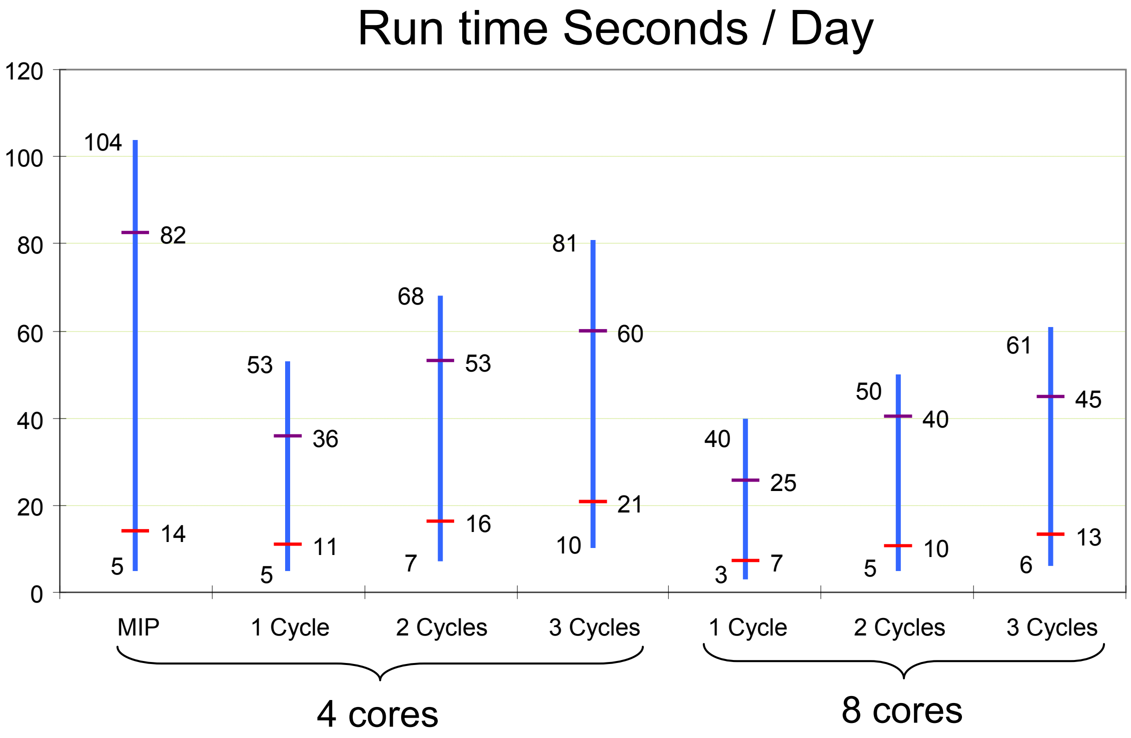
DAYZER Parallel SCUC (PUC)

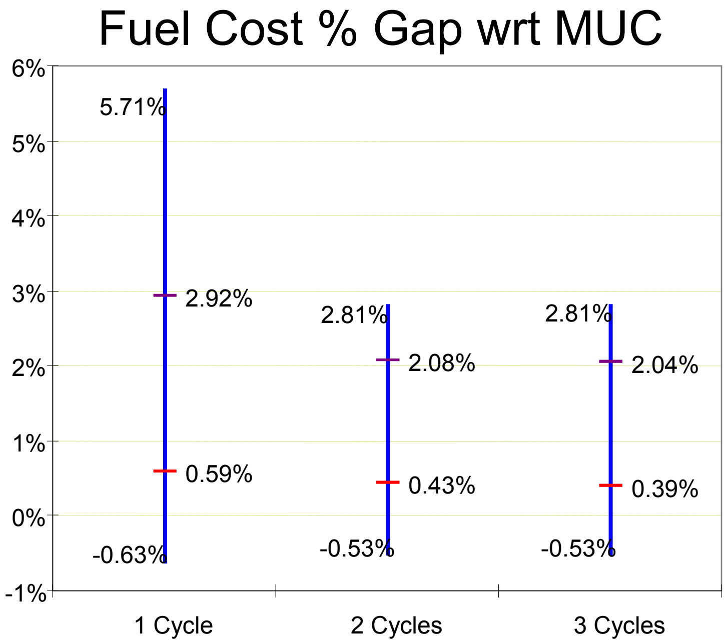
Solves the same problem as MUC but utilizes Lagrangian Relaxation Subgradient Optimization by decomposing the problem across time (hourly dispatch) as well as space (unit commitment). Some of the more distinctive aspects of our implementation are:

- Target duality gap estimated by solving an initial relaxation problem.
- Adaptive step size initialization and update heuristics incorporating the target gap estimate as well as a measure of the current over/under commit.
- Early termination heuristics based on the target gap and step size update history.
- Unit sub problems modeled and solved as MILP (same as in the global version).
- Ramping constraints imposed on hourly dispatch using latest UC solutions.
- A unit (partial) decommitment phase based on semi-global uplift minimization.
- Coverage of all transmission constraints by adaptively modifying the dispatch LPs.
- Pump storage optimization handled by updating UC for a fixed PS solution, then relaxing the associated PS constraints and updating their multipliers while UC is kept fixed. We then iterate over multiple cycles of this to achieve convergence.
- Losses and Contingency Analysis calculations interleaved with UC iterations.

PUC performance on a small scale problem (NEPOOL) with Pump Storage Optimization

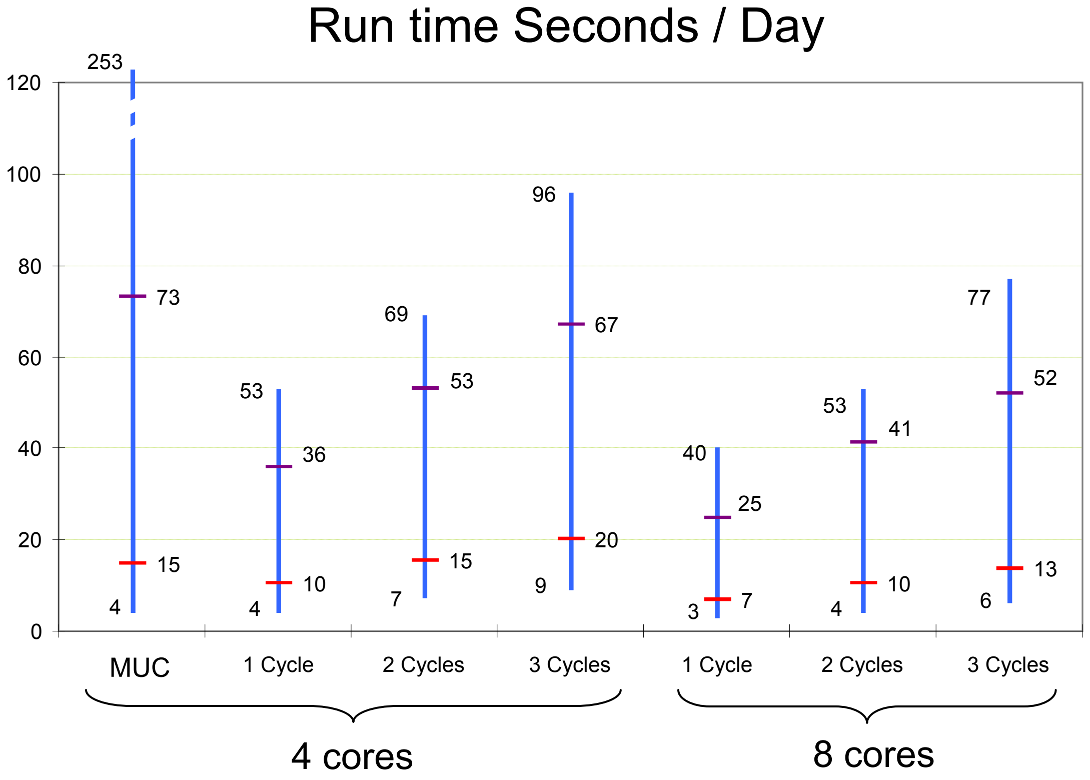


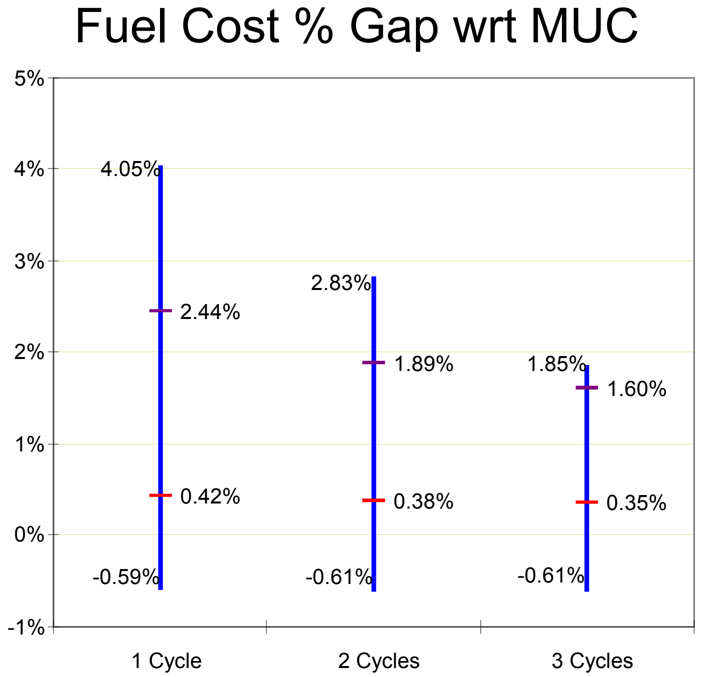

Run time Seconds / Day


Fuel Cost % Gap wrt MUC

Statistics from runs on two different machines over 365 days in 2014

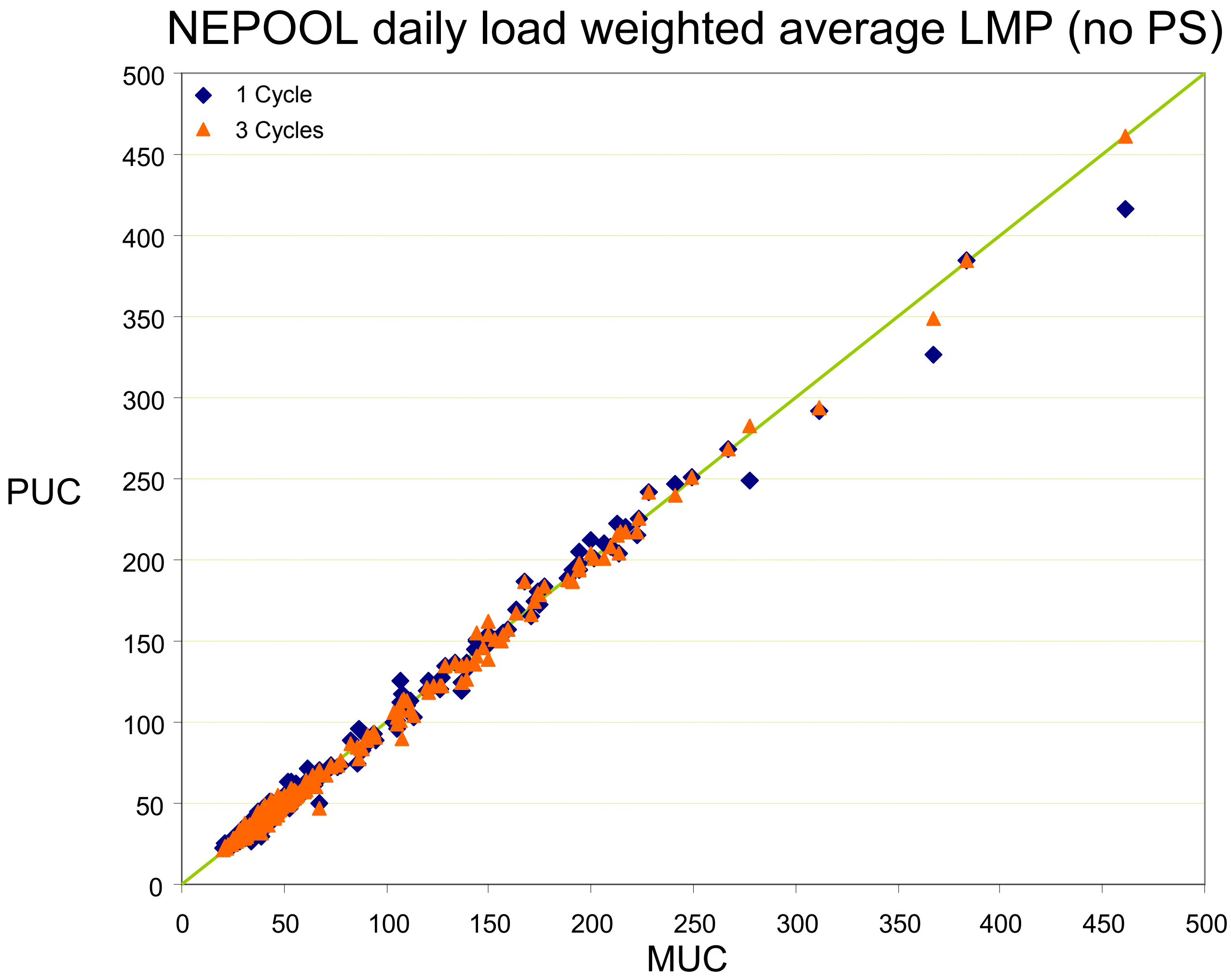
Results from same runs without Pump Storage highlight the large impact of these resources





- The effective parallelization estimated from these runs is between 88% and 93% which implies a speed up factor between 6 and 9 at 24 cores.
- Even without PS optimization PUC solution quality improves with additional cycles.

LMP comparison highlights the improvement gained from additional PUC cycles



RMS error:

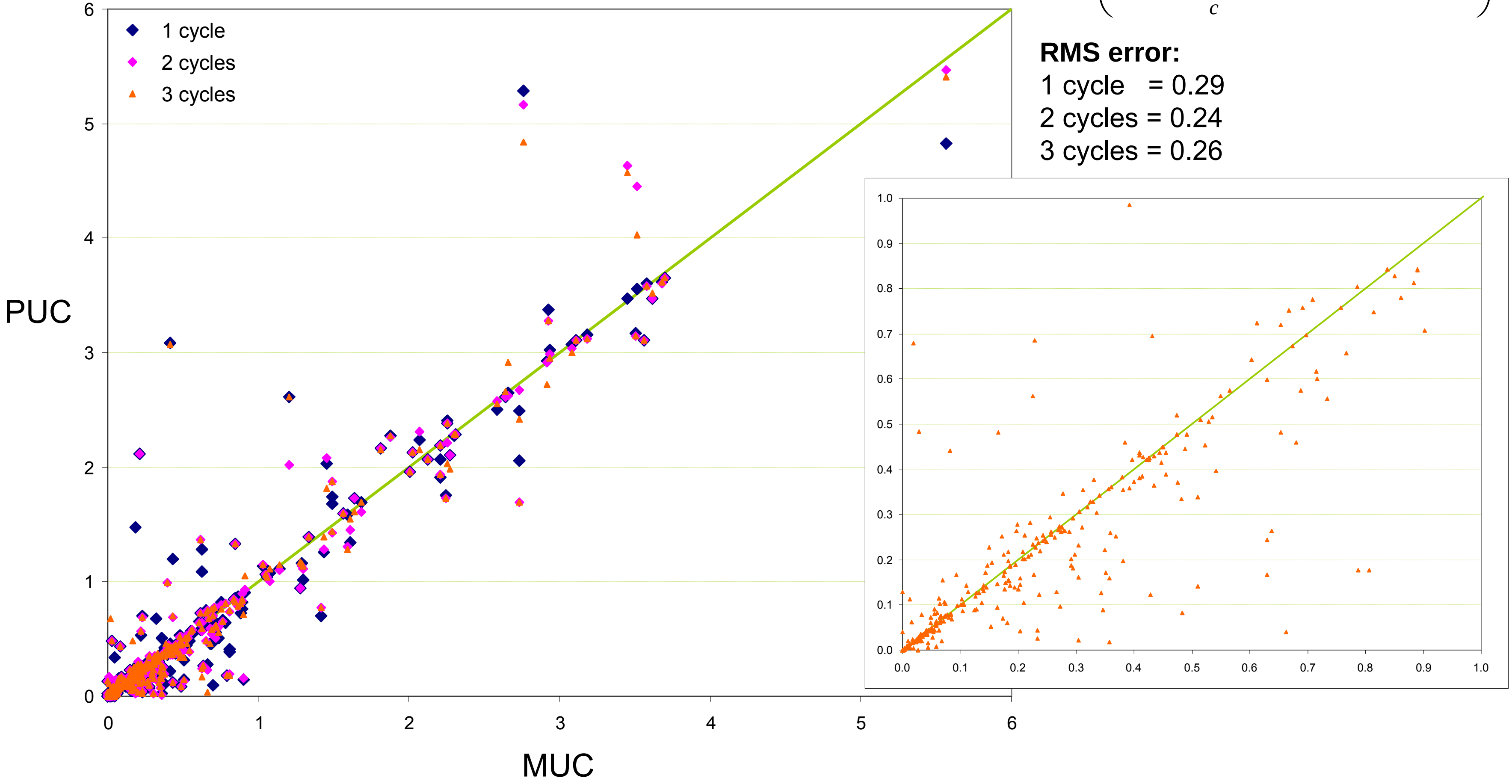
1 cycle = 5.33
2 cycles = 4.41
3 cycles = 3.82

RMS error (w/PS):

1 cycle = 6.36
2 cycles = 5.18
3 cycles = 5.20

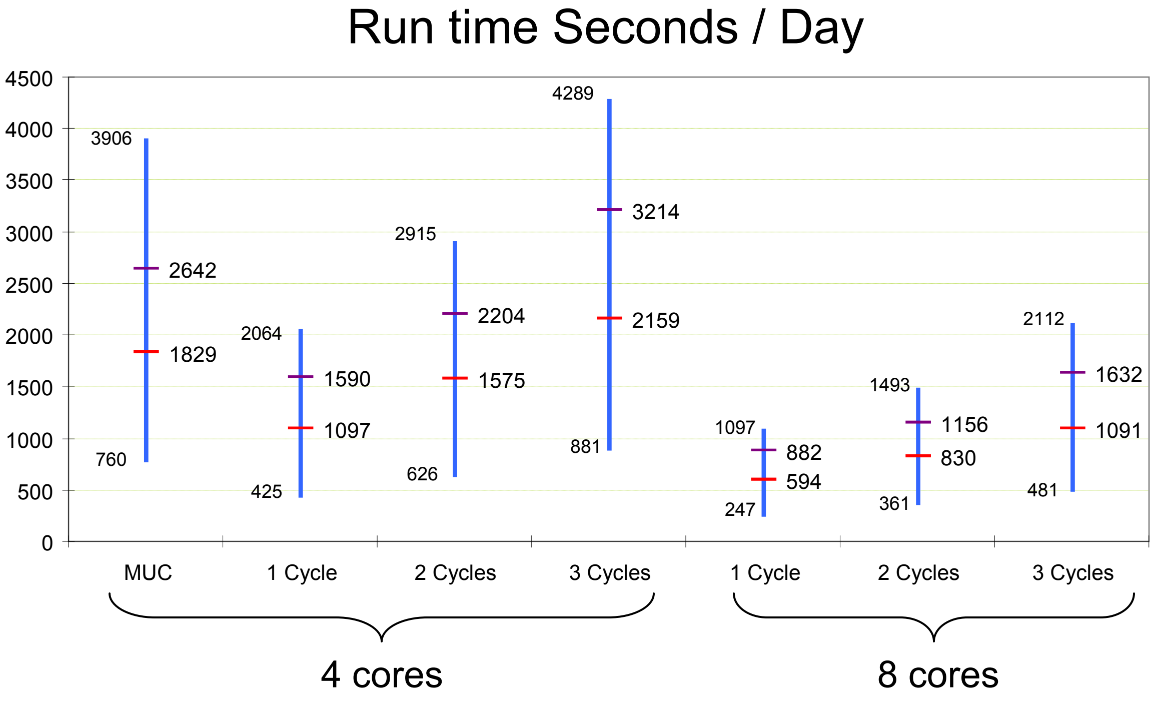
However, congestion pattern convergence may require even more cycles

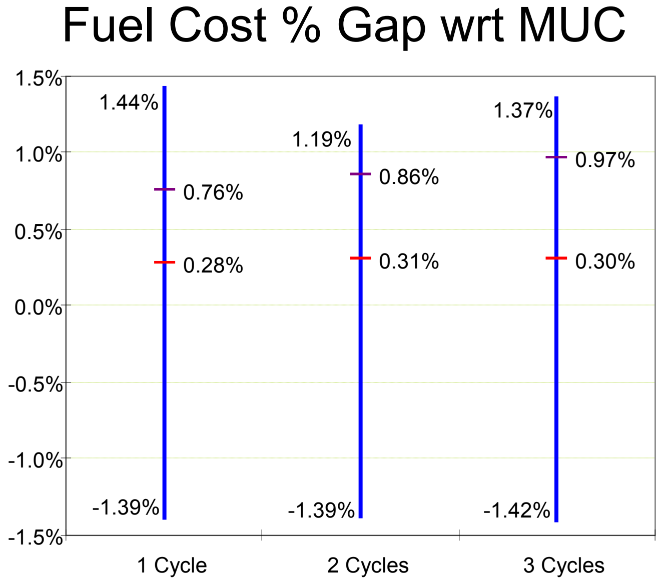
Daily average of normalized hourly transmission rent = $$\frac{1}{24}\sum_{h}\left(\frac{\sum_{c}|Flow(c,h)|\times|SP(c,h)|}{\sum_{c}|Flow(c,h)|}\right)$$

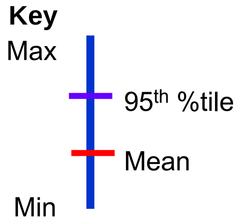


RMS error:
1 cycle = 0.29
2 cycles = 0.24
3 cycles = 0.26

PUC performance on a large scale problem (PJM+MISO combined) with Pump Storage Optimization

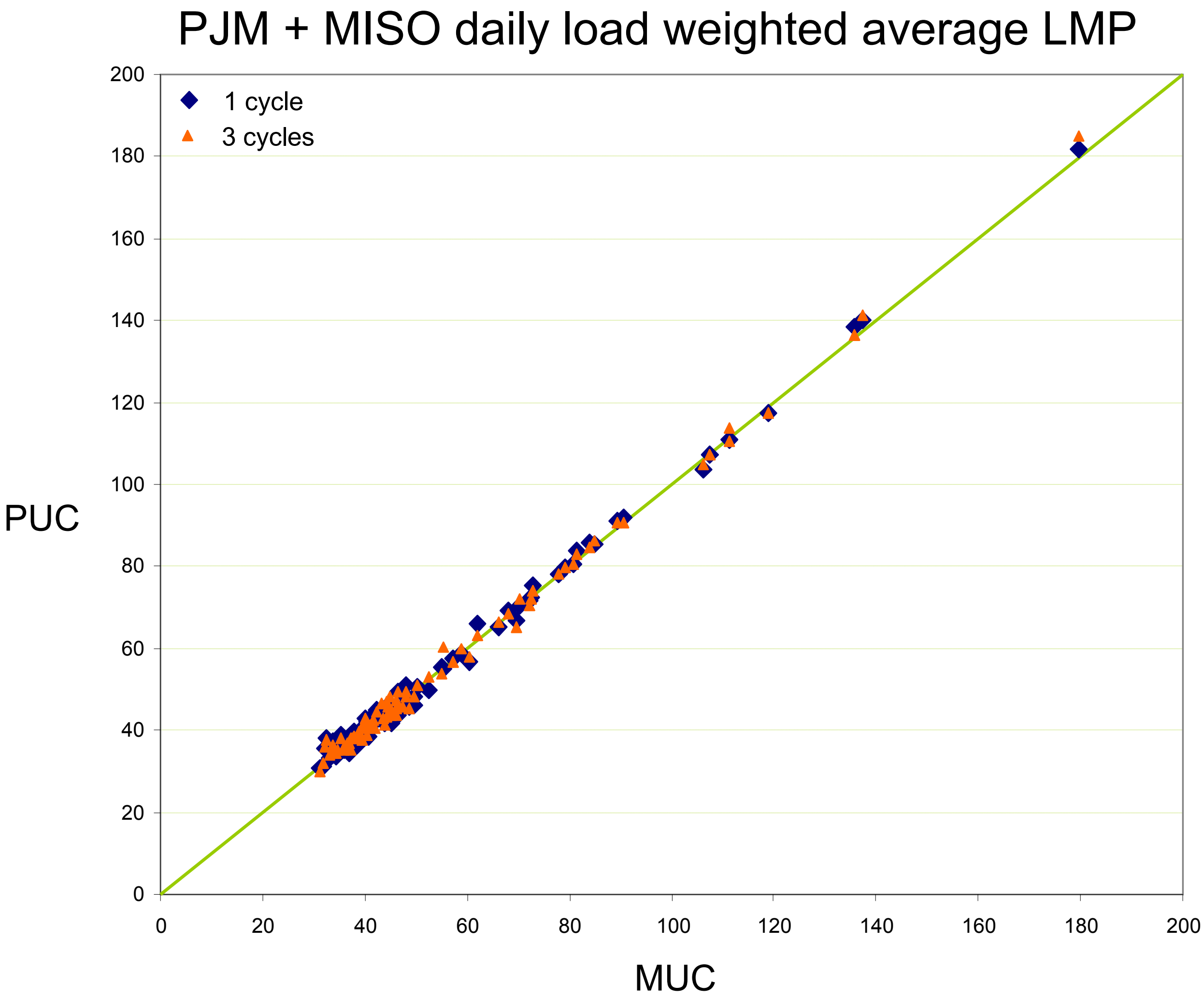






The above timing results imply an effective parallelization of almost 98% and hence a speed up factor of nearly 16 at 24 cores.

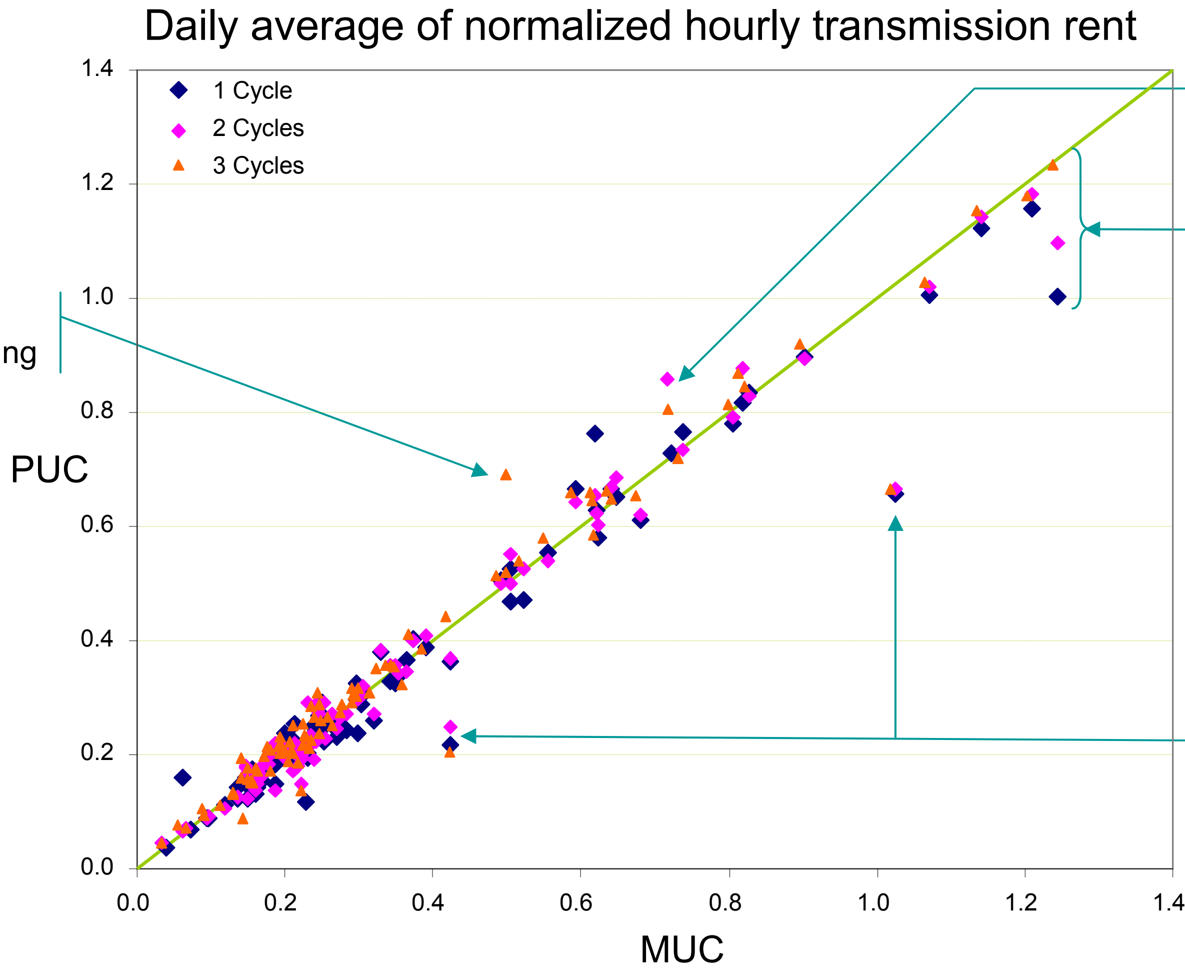
LMP comparison shows much smaller impact of additional PUC cycles compared to NEPOOL



RMS error:
1 cycle = 1.84
2 cycles = 1.92
3 cycles = 1.84

RMS error for MUC LMP < 100:
1 cycle = 1.84
2 cycles = 1.87
3 cycles = 1.75

Improvement in the alignment of congestion patterns beyond 2 cycles is not uniform



Converging but additional cycles are required

Converging to similar congestion pattern at a higher gap

Lower cost solution diverging

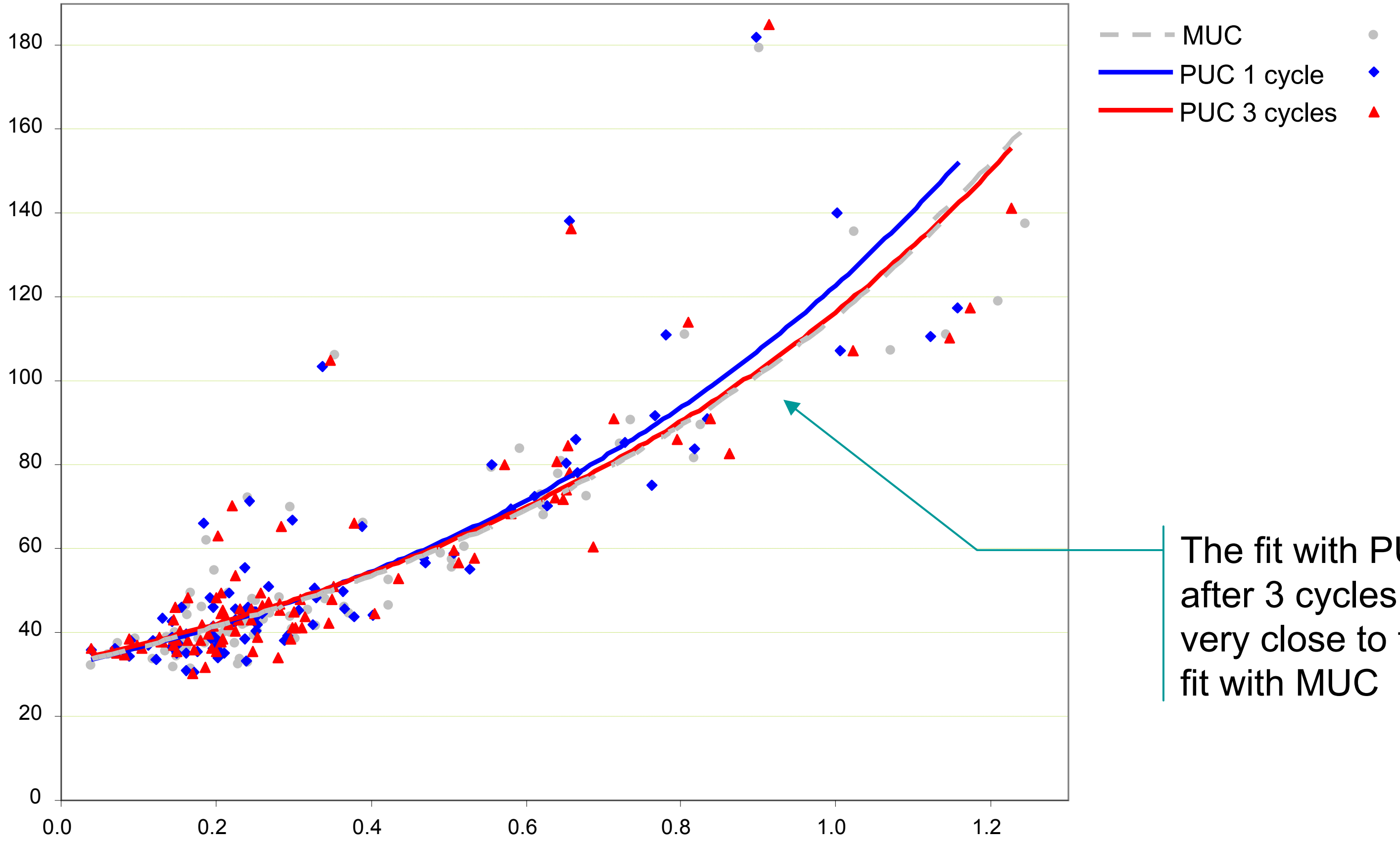
PUC finds better solution

RMS error:
1 cycle = 0.0618
2 cycles = 0.0544
3 cycles = 0.0554

RMS error for MUC rent < 0.5:
1 cycle = 0.0389
2 cycles = 0.0337
3 cycles = 0.0358

However, additional cycles may have a benefit depending on how the results are used

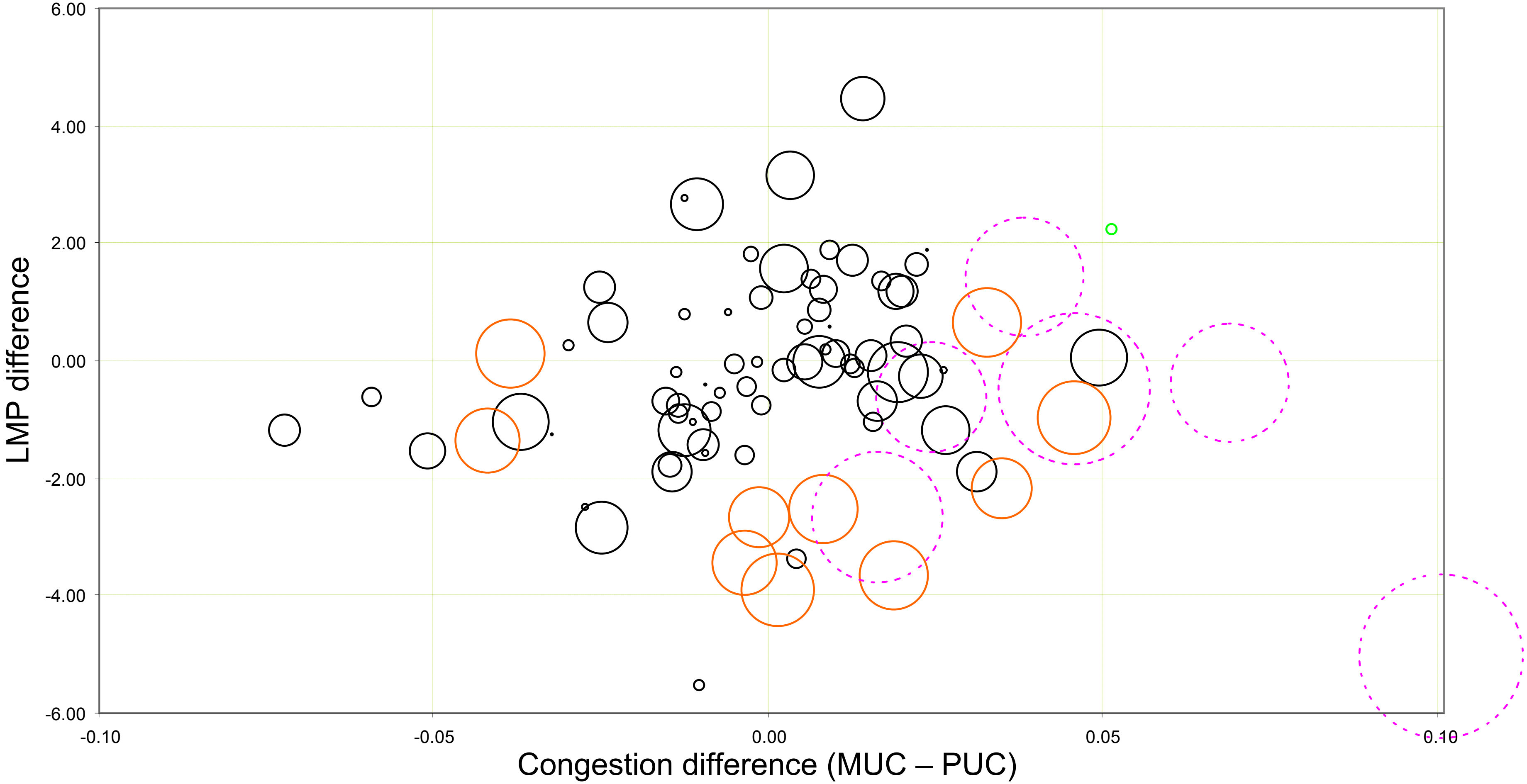
Average load weighted LMP as a function of average normalized transmission rent (exponential fit)



The fit with PUC after 3 cycles is very close to the fit with MUC

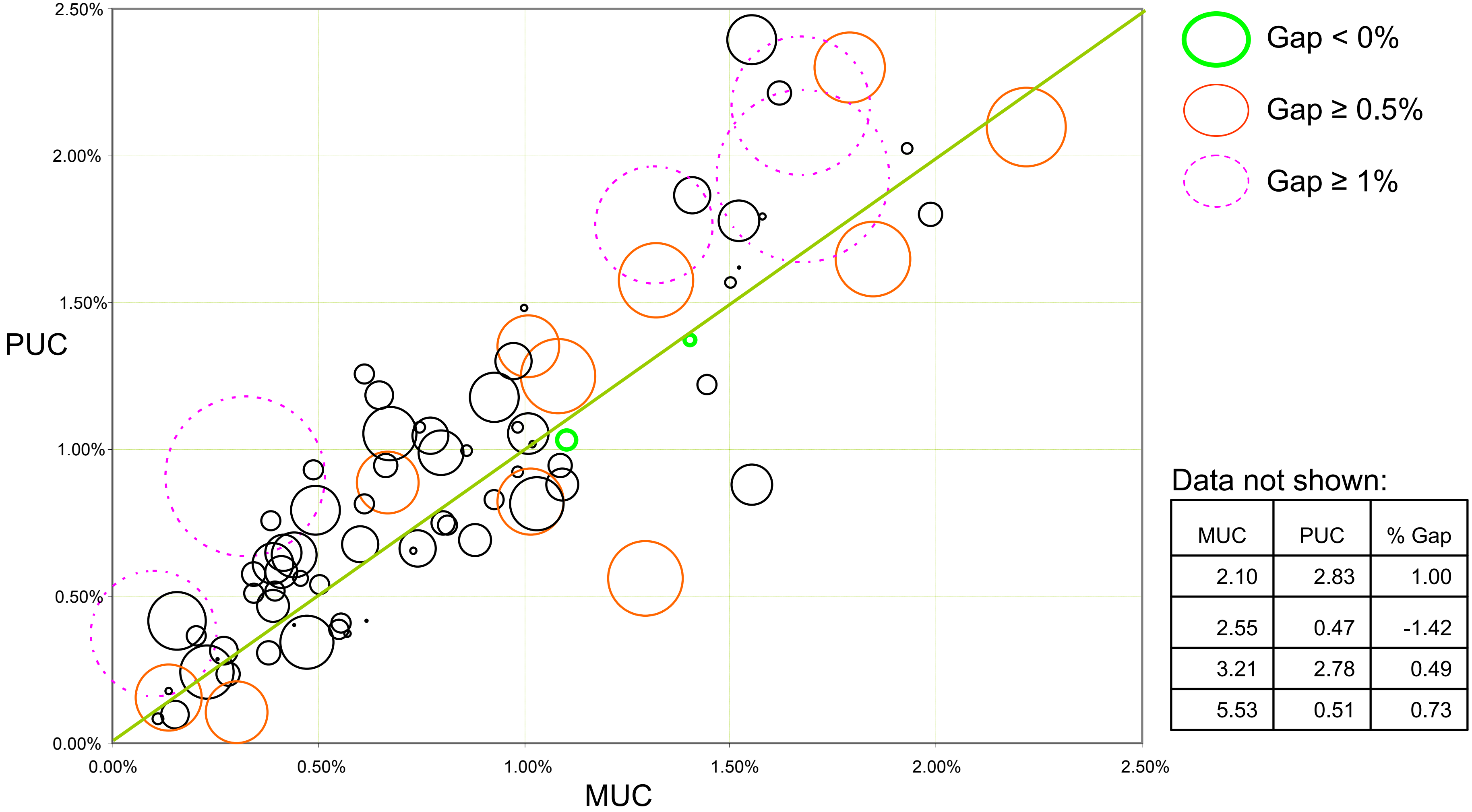
Solutions close together in terms of fuel cost may still differ significantly in prices

Bubble width indicates % fuel cost gap wrt MUC solution for PUC after 3 cycles



Uplift solutions are comparable in most cases

Uplift as a percentage of generation revenue (PS excluded)



Data not shown:

MUC	PUC	% Gap
2.10	2.83	1.00
2.55	0.47	-1.42
3.21	2.78	0.49
5.53	0.51	0.73

Conclusions

- PUC overall solution quality comes close to that of MUC and is probably acceptable for a range of applications where run time performance is more critical.
- In addition, a final MUC pass with constraints and initial solution developed via a single cycle PUC can be used to improve both solution quality and run time performance for larger models.
- The combination of distributed and parallel techniques as proposed here can be used to create a flexible and scalable computing environment that can handle the large workloads required to effectively explore the dynamics of multiple interacting energy markets without having to sacrifice on modeling fidelity.