Applications of High Performance Computing for Dynamic Security Assessment of Power Systems

A Presentation at the IEEE-PES GM 2013 Panel Session

Potential Impact of High-Performance Computing on the Power Grid

Vancouver, Canada
July 2013

Lei Wang

Powertech Labs Inc.

Surrey, BC, Canada

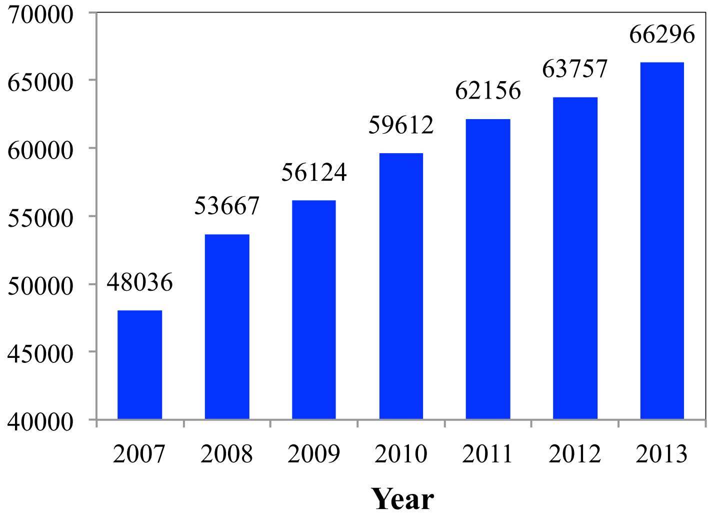
Powertech

Introduction

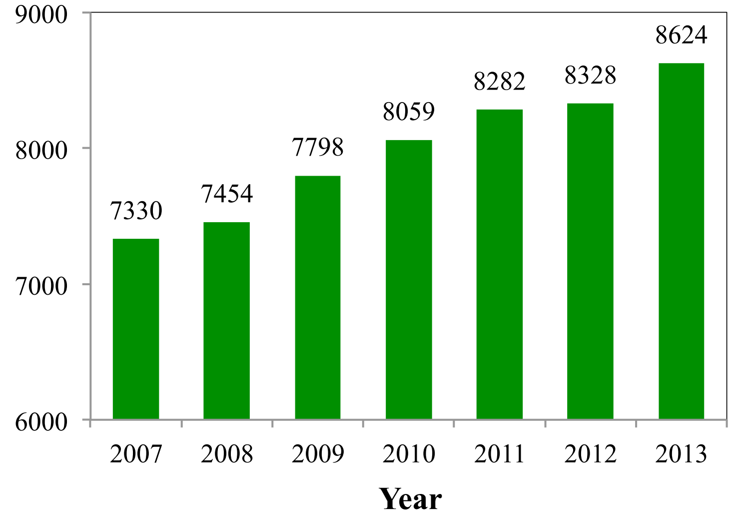
- **Stability analysis** is one of the basic forms of power system analysis
- In recent years, stability analysis has been increasingly performed as a real-time control center application, referred to as on-line Dynamic Security Assessment (**on-line DSA**),
- Traditional DSA requires **intensive numerical computations** and thus usually takes long time to complete
- **Improving the speed of DSA** has been the focus of extensive research and development for many years
- This presentation discusses
  - Challenges in on-line DSA
  - **Techniques** that have been applied, or have potential, to improve the speed of DSA by using high performance computing techniques

Challenges

- Size of power system models has been growing



No. of buses
(**38%** increase in 6 years)



No. of generators
(**18%** increase in 6 years)

The sizes of the full US/Canada eastern interconnection models

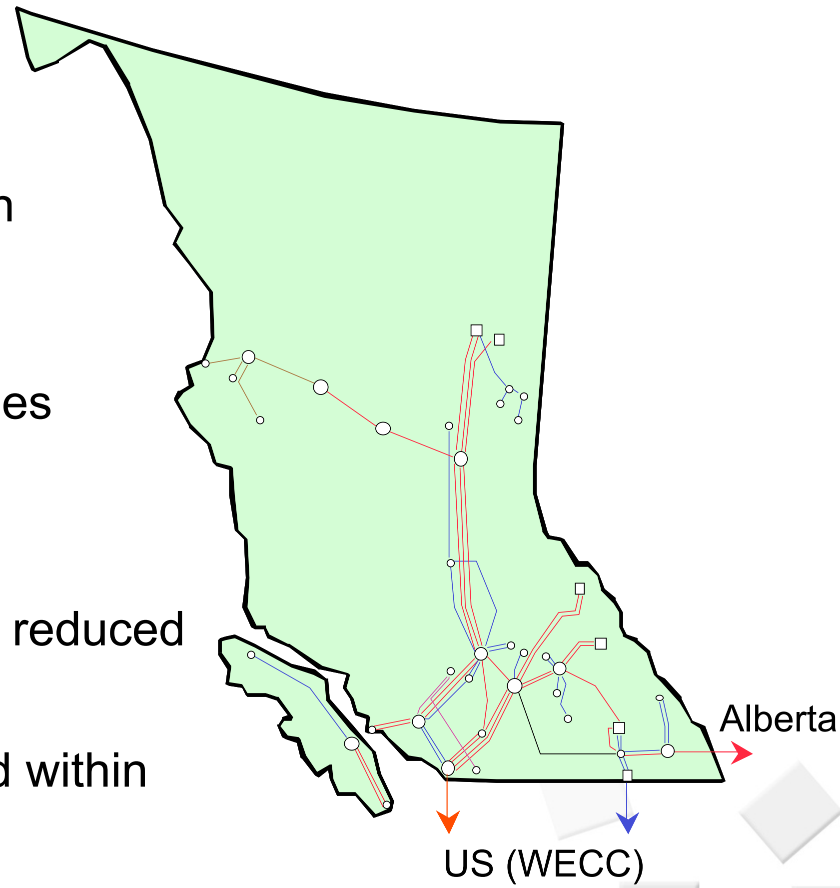
- The largest known **real-time models** are in the order of 40,000 buses
- Computation speed is a serious concern for such large system models

Challenges (cont'd)

- **Model complexity**
  - Trend to include node/breaker details in models
  - Adoption of dynamic load models such as composite load models
  - Increased applications of controls based on power electronics (such as those found in HVDC, FACTS, wind turbines, etc.)
  - Consideration of relay and protection models
- Need to study **more contingencies**
  - May be in the order of thousands
  - Compliancy with NERC criteria
- **More advanced analysis tasks**
  - Determination of stability limits
  - Identification of control actions

Challenges (cont'd)

- Example of stability limit computations – **BC Hydro** TTC requirements
  - 240 hourly, 72 daily, and 112 weekly cases
  - 7 transfer paths to study, resulting in (240+72+112)×7=**2968** total one-dimensional transfer analysis
  - Subject to all applicable contingencies
  - For all applicable security criteria (**voltage** and **transient**)
  - Using the full BC Hydro system and reduced external network (5,000 buses)
  - All computations must be completed within **one hour**



Where are we now?

- Computation speed for time-domain simulations has been improved significantly over the past decades
  - Faster computers
  - More efficient compilers
  - Better computational algorithms
- Examples
  - Time required to run a 10-second simulation
  - On Intel Xeon X5670 CPU (2.9 GHz); **no** distributed/parallel computation

System size	Medium	Large	Very large
No. buses	7,210	16,330	54,481
No. of generators	1,040	1,922	5,203
Time (seconds)	**6.4**	**39.5**	**243.7**

Faster than real time! (6.4)

Improvement required here! (243.7)

High Performance Computing (HPC)

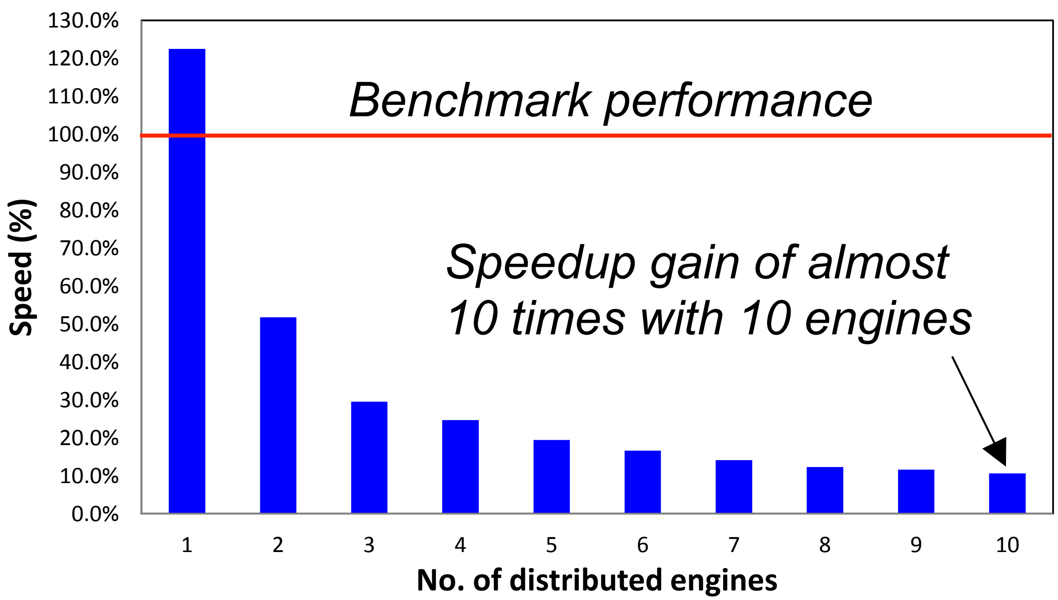
- The need to use HPC for on-line DSA is clear
- Options
  - **Super computers**
    - ✓ Highest possible performance
    - ✓ May not be easily accessible for production use; high cost (?)
  - High performance **multi-chip multi-core servers / GPU**
    - ✓ High performance/cost ratio: a high-end dual-chip, 16-core server costs less than $13K
    - ✓ Maintenance/upgrade requirements; hard to customize
  - **Cloud computing**
    - ✓ Cost effective
    - ✓ Easily customizable; maintenance free
    - ✓ Security concern (?)

Methods for on-line DSA using HPC

- **Distributed** computations
  - Multiple simulations (for different system conditions or contingencies) are performed concurrently in multiple CPUs or CPU cores
  - Relatively easy to implement
  - Difficult to apply for computation tasks that are series in nature. For example, one simulation; one stability limit analysis
- **Parallel** computations
  - One simulation (or part of a simulation such as network solution) is performed in multiple CPU or CPU cores
  - Difficult to implement, particularly when there is need to co-exist with high-efficiency series computations
  - Required to break the performance bottleneck

Distributed computation example

- Illustrates the efficiency of a distributed computation scheme
- The power system model:
  - 6,620 buses, 1,978 generators, 400 contingencies
- Different number of computation engines are used
  - Speed performance is benchmarked as % of time required to perform the same analysis in non-distributed mode using the same simulation code

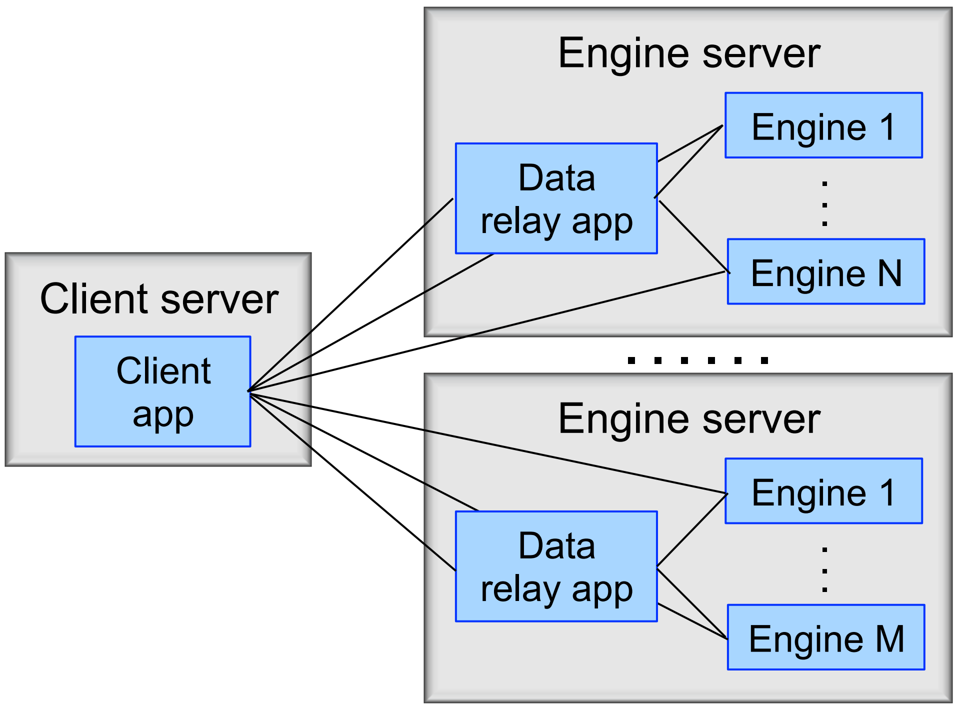


Improving the efficiency in distributed computations

- There may be serious performance barrier when using **a large number of servers** for a distributed computation task
  - Network Latency
  - Network Throughput
  - Concurrency / Locking
  - Contention for system resources
- The solutions
  - Better load balancing (use dynamic scheduling)
  - Delegation of work (data relays)
  - Go with parallel computation if possible

Example: Data Relay

- Efficient to minimize network data traffic during distributed computations if a large number of servers are used
  - This has been implemented in a number of on-line DSA systems which use multiple servers for large scale distributed computations



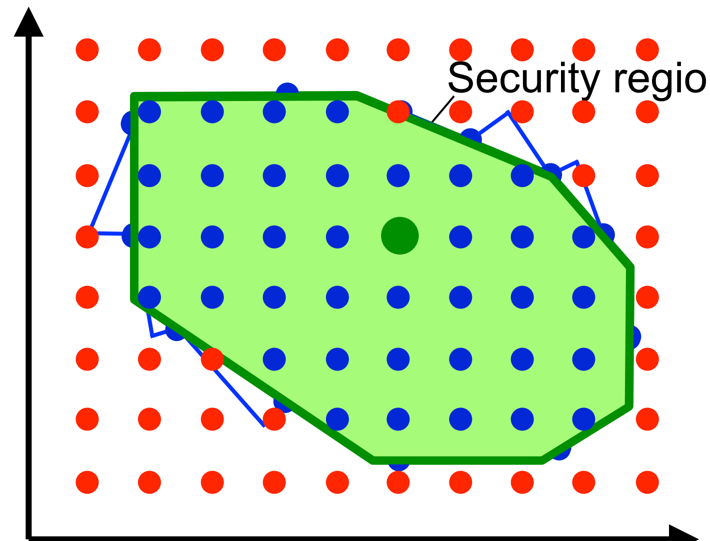
Parallel computation example

- Network solution in time-domain simulations may take up to 40% of computation time for large system models
  - Previous research done at UBC showed that speedups of **nearly 7 times** was achieved for such computations with research-grade parallel code
- A parallel algorithm is implemented recently by Powertech in the **production code** of a transient stability analysis program
  - It can automatically perform network solutions in multiple threads
- Performance results
  - For a system model with 38,611 buses (this is the dimension of the network equation) and 2,510 generators
  - With 4 threads enabled for parallel network solution, a speedup of **1.3** is observed for the overall simulation speed
  - Or, the speedup for the network solution portion is roughly **2.3**

Challenges for determination of stability limits

- A 2-D stability limit search problem is shown
- Computationally very intensive since many time-domain simulations are required to determine the security region

Algorithm 1: suitable for **full** distributed/parallel processing + more number of simulations are required



Conclusions

- **Challenges** of performing DSA, specially on-line DSA
  - Growing size of power system models
  - Model complexity
  - Need to study more contingencies
  - More advanced analysis tasks
- **Possibilities** of improving DSA speed
  - Options available
  - Different approaches
- **Examples** are given to show the achievements and potentials