Distributed Computation Topics

2010 PowerWorld Client Conference

June 23, 2010

Portland, Oregon

Outline

- Motivation
- Implementation
- Distributable Calculations
- Configuration
- IT Concerns
- Future Directions

Introduction

- Distributed Computation is a set of add-ons that split large problems into smaller, independent problems.
- The independent problems can then be sent to multiple processors on the same machine or even to other computers on a network to spread the workload and dramatically reduce overall calculation time.

Motivation

- Historically, problem sets were smaller
  - Slow computers and little available memory
    - Limited case size significantly
    - Limited set of contingencies (generally only the worst)
  - Fewer regulations and penalties
  - Few data warehousing tools to process and store large amounts of data
  - End result: Small problems that took lots of time and only calculated a few power flows.

Motivation

- Problem sets are much larger now
  - Faster computers with lots of memory
    - Larger, more detailed models (even to the node-breaker level)
    - More detailed contingency lists
  - More regulations and audits with significant potential penalties
    - Contingency lists significantly grew (N-1-1, etc.) & number of ATC scenarios
  - End result: much larger calculations, with *many* more power flow solutions

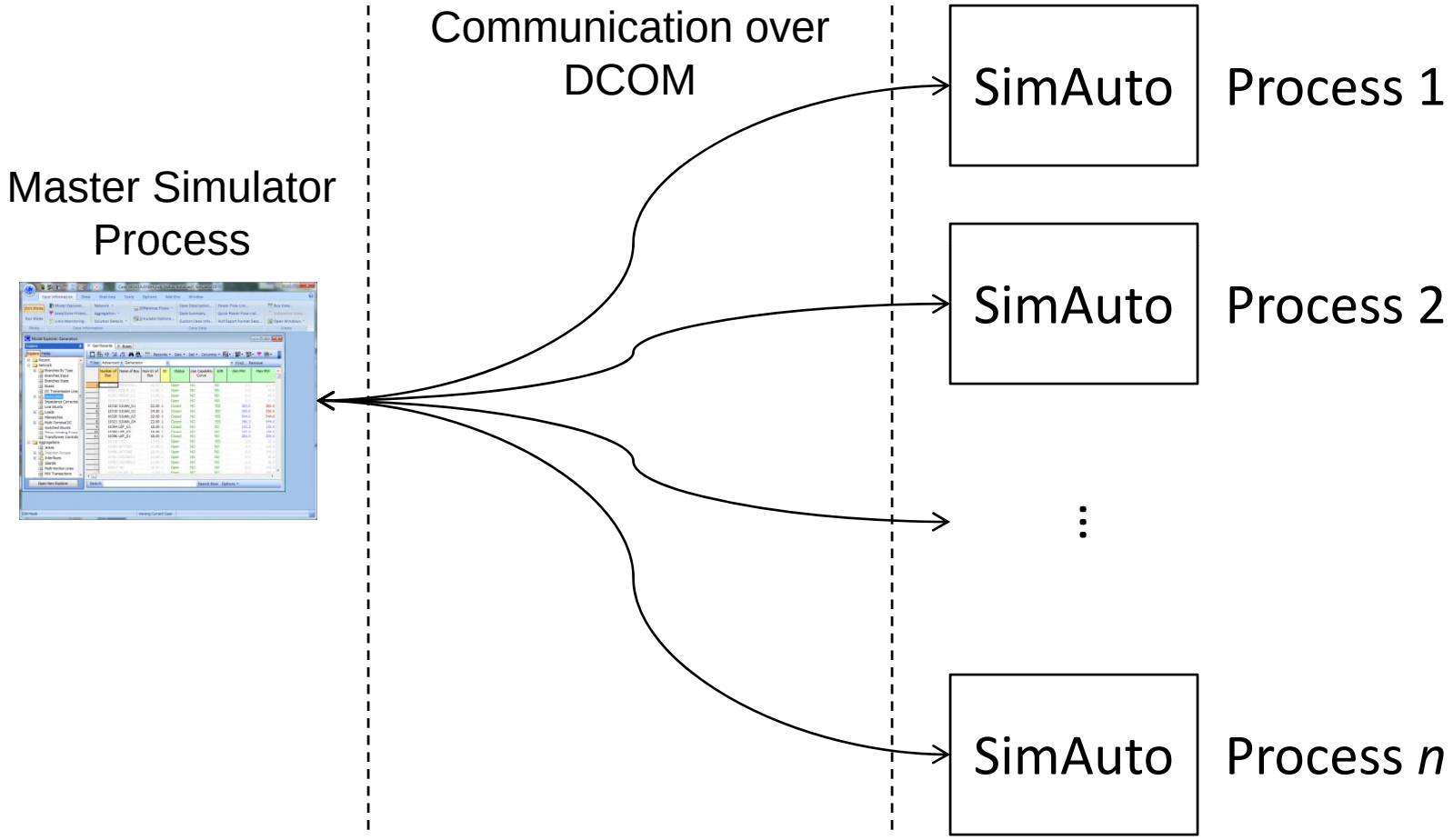
Motivation

- Ultimate problem
  - Processor speed hasn't kept up with problem set size
- Solution
  - We have many more cores/processors available to do the work
  - Split the problem set into multiple, smaller problems that can be solved independently.
    - While power flows are difficult to parallelize, contingencies and ATC scenarios are completely independent.

Implementation

- Use SimAuto and additional specialized COM (Component Object Model) programming interfaces for communication
- Use Microsoft Distributed COM to access computation resources on other machines

Implementation



Implementation

- How does Simulator distribute calculations?
  - Create a pool of SimAuto instances.
  - Send current case to each SimAuto in the pool.
  - Send current tool's settings to each SimAuto.
  - Break the calculation into small chunks (groups of contingencies, individual ATC scenarios, etc.).
  - Distribute chunks to each available SimAuto in the pool until there are no more chunks left.
    - After a chunk is completed, gather the results into a list.
    - Release the SimAuto instance back into the pool of available SimAutos
  - Take the results list and apply it to the current case

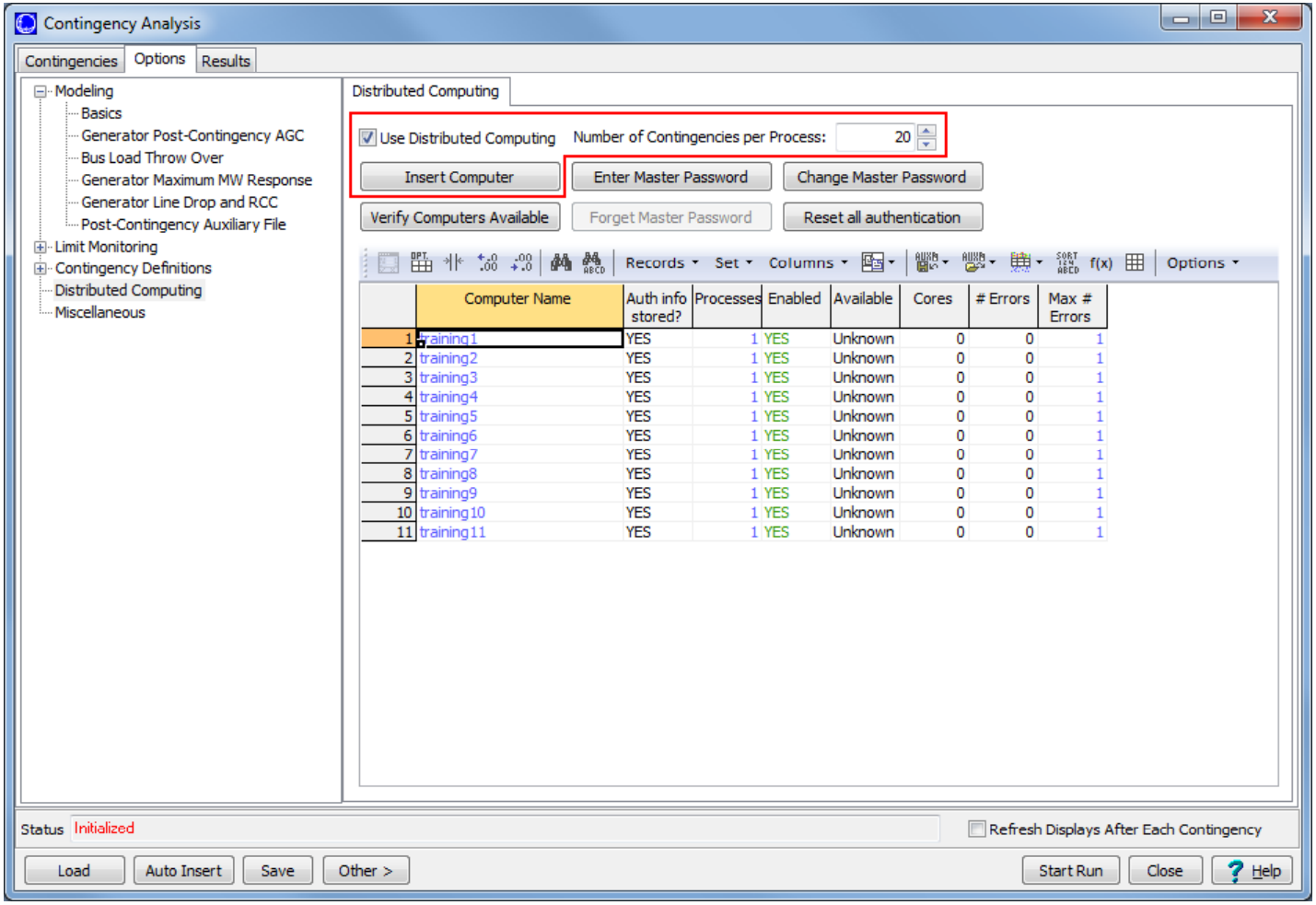
Distributable Calculations

- What makes a good distributable calculation?
  - Easily identifiable cleaving points
    - Contingency (steady state and transient stability)
    - ATC scenario
    - Bus injection (QV)
  - Independence of individual operations
    - SimAutos within pool do not communicate with each other
    - Too much network communication causes large overhead

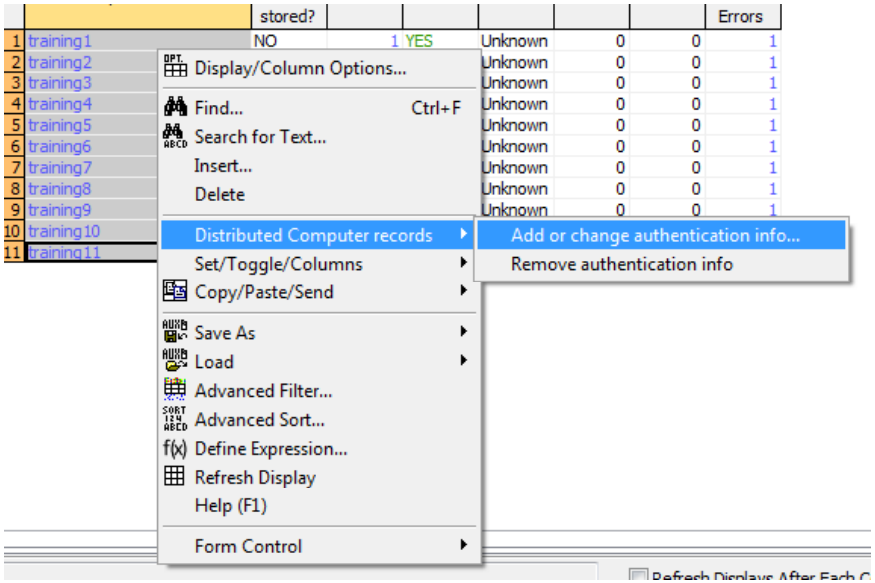
Distributable Calculations

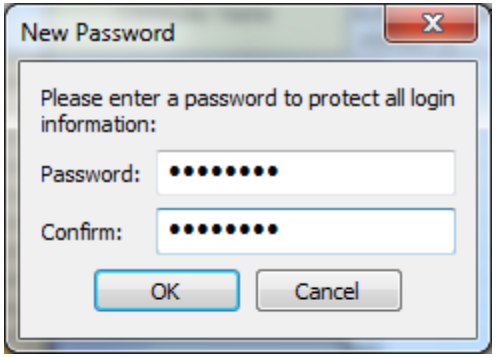
- Calculations under development (nearly complete):
  - Contingency Analysis
  - ATC
- Soon to be undertaken:
  - Transient Stability
  - QV Curves

Configuration



Configuration





IT Concerns

- All this distributed automation has a big drawback: SECURITY
  - Microsoft DCOM has been known to have security issues
  - Login accounts must be adjusted or created to allow SimAuto to run on networked machines
  - Security policies may need to be updated

IT Concerns

- The upshot?
  - DCOM is a mature and robust technology, so new, large security issues are unlikely
  - The potential for massive performance improvements in total elapsed time
  - Put unused computers to work at night, on weekends, etc.

Future Directions

- Extended to other tools?
- Load balancing/management of computation resources
- Examine efficiency of distributed computation tools
- Security diagnostic tool
- Suggestions?