NORTHERN TRUST

Operational Risk Quantification System

Northern Trust Corporation

May 2012

Achieving High-Performing, Simulation-Based Operational Risk Measurement with R and RevoScaleR

Presented by David Humke Vice President, Corporate Risk Analytics and Insurance, Northern Trust

Northern Trust

Agenda

- Basel II Overview
    1. Operational Risk – Definition
    2. Requirements of an Operational Risk Exposure
- Loss Distribution Approach
    - Unit of Measure Concept
    - Severity Modeling and Frequency Modeling
    - Monte Carlo Simulation
- Potential solutions to enhance Monte Carlo Simulation
    - Describe various test environments
    - Various Graphs/Timing Tables from Touch Point Meetings with Revolution R

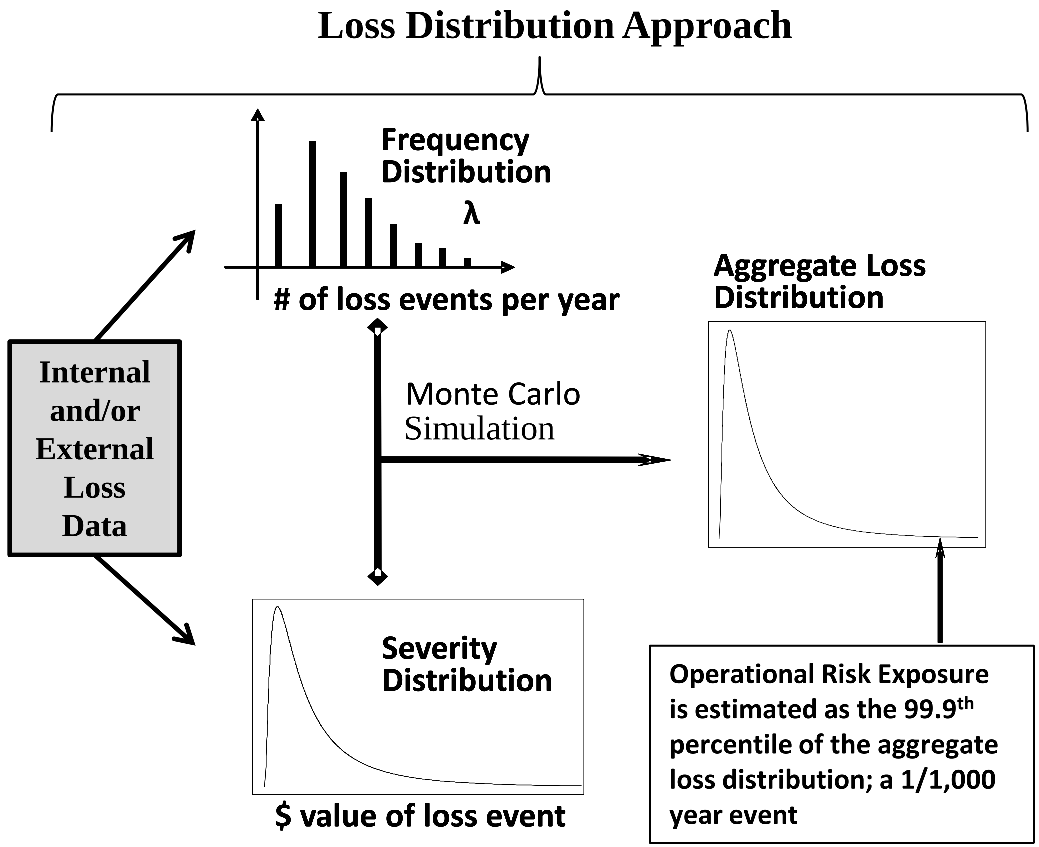
Basel II and Operational Risk

- In December of 2007, the US Federal Reserve System finalized a document commonly referred to as the "Final Rules" which set forth general requirements for the measurement of operational risk by large US financial institutions[1]
  - These rules defined **operational risk** as the risk of loss resulting from inadequate or failed internal processes, people, and systems or from external events (including legal risk but excluding strategic and reputational risk)
    - Seven Distinct Basel Loss Event Types:
      1. Internal Fraud
      2. External Fraud
      3. Business Disruptions/System Failure
      4. Execution, Delivery and Process Management
      5. Damage to Physical Assets
      6. Clients, Products, and Business Practice Matters
      7. Employee Practices and Workplace Safety Issues.
  - The Final Rules require banks to produce an **operational risk exposure** that corresponds to the 99.9th percentile of the distribution of potential aggregate operational losses, as generated by the bank's operational risk quantification system over a one-year horizon.
    - Exposure estimates must:
      a) Incorporate four data elements: Internal Loss Data, External Loss Data, Scenario Analysis Data, and Business Environment/Internal Control Factor data.
      b) Be calculated using systematic, transparent, verifiable, and credible methodologies
- In recent years, the banking industry has focused on the use of the Loss Distribution Approach (LDA) to calculate operational risk exposure estimates based on internal and external loss data.

1.Risk-Based Capital Standards: Advanced Capital Adequacy Framework – Basel II; Final Rule (2007), Federal Register 72(235), 69407 – 408.

Loss Distribution Approach (LDA)

- The LDA models two primary components of operational loss data:
  - Loss Frequency
    - The banking industry has widely accepted a Poisson distribution as an appropriate distribution.
  - Loss Severity
    - Fitting a parametric distribution to operational loss data is one the biggest challenges in measuring operational risk exposure.
    - Various distributions provided in the actuar() package and GAMLSS allow for the fitting of various truncated severity distributions
      - e.g. – dlnorm, dlgamma, dpareto, etc.
- Monte Carlo Simulation is utilized to bring the two distributions together.
  - A large number of simulations must be run to observe a sufficient number of losses to reasonably assess what a 1 in 1,000 year event might look like
    - This can create a multi-day bottleneck in the modeling process.



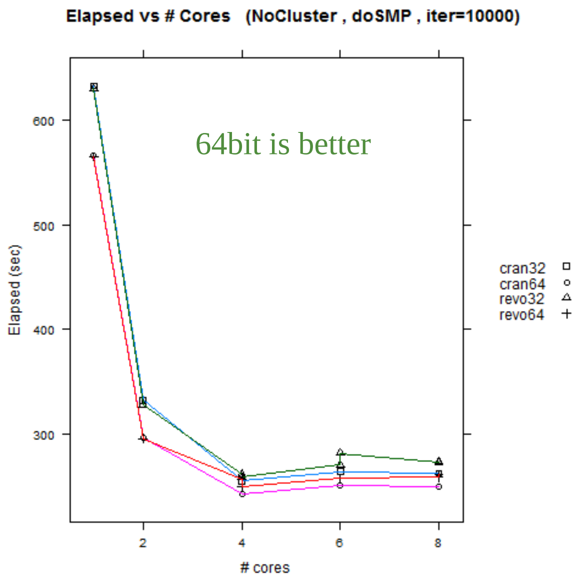
<u>Example:</u>

```
# Randomly draw n frequency observations from a Poisson distribution, then draw random severities from the specified
  truncated severity distribution, truncated at point a.  Sum up each of the individual loss amounts.
    f_tr <- function() {
      sum(do.call("rtrunc", c(n=rpois(1, lambda), spec=distName, a=a, parList)))
    }
# Simulate a large number of iterations and replicate the simulation a number of times to reduce sample noise
     simuMatrix <- replicate(nSimu, replicate(nIter, f_tr()))
```

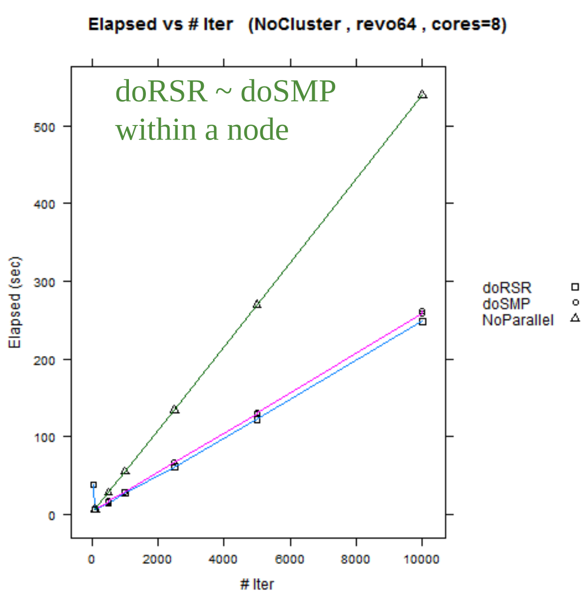
Monte Carlo Simulation Benchmarking Analysis

- Northern Trust and Revolution Analytics Evaluate Various Methods to Enhance Monte Carlo Simulation
  - Use a different version of R: 32B, 64B (e.g. – Update your operating system)
  - Use various parallelization packages: doSMP, doSNOW, doRSR
  - Use multiple processors and/or machines: single node with multiple cores, cluster of CPUs with multiple cores
- Metrics used to evaluate each method:
  - Elapsed Time by Step
  - Memory usage
- Hardware Environments:
  - 4-core laptop
  - 3-node High Performing Cluster (HPC) on Amazon Cloud
    - Configured and run with 8-cores on each node
    - Each node was restricted from 16- to 8-cores
- Comparisons:
  - Revo vs Cran
  - 32- vs 64-bit
  - Impact of parallelization within and across nodes

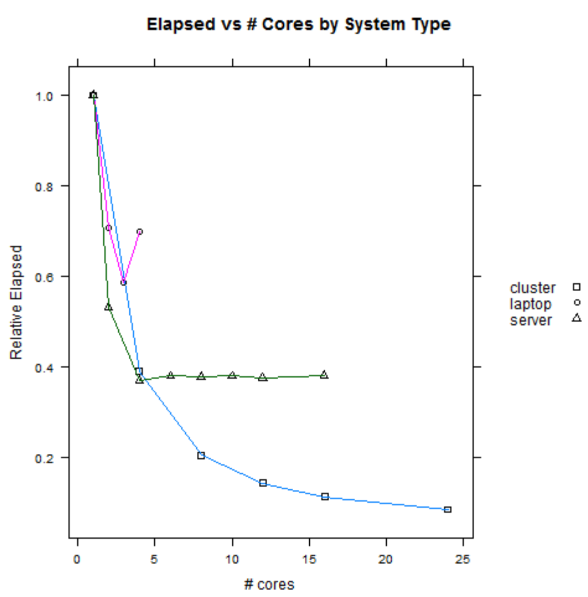
Monte Carlo Benchmarking Highlights



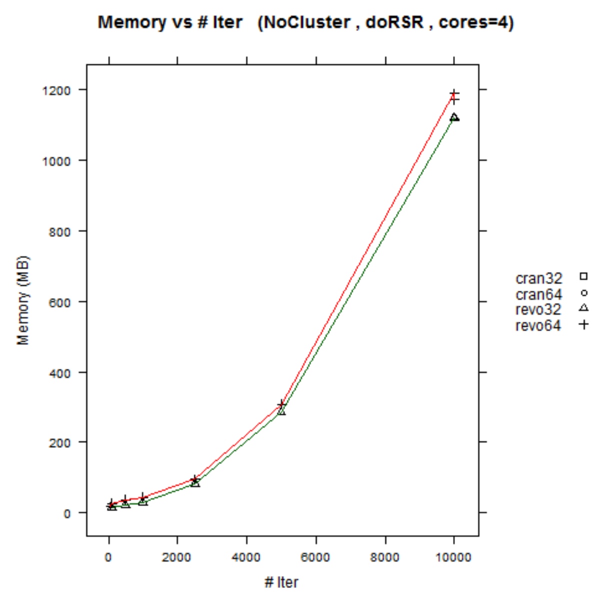
- Revolution Analytics' parallelization can be easily scaled up from laptop/server to the cluster using Revolution Analytics' distributed computing capabilities
- Parallelization greatly improves simulation performance
- Elapsed time is linear in # of iterations
- Performance improves with # of cores
- Revo ~ Cran within a node (no MKL impact in this study)
- doRSR slightly better than doSMP on a single server
- 64bit marginally better that 32bit
- Performance scales with cluster resources
- Memory use just driven by # of iterations



Scales with # Cores



Memory Trends



Take-Aways, Next Steps, and Contacts

Parallizations Offers Business Enhancements:

- Less time spent waiting on programs to complete
  - Means more time to analyze drivers of change (e.g. – underlying data changes)
- More efficient management of computing resources
  - No need to manage/schedule programs
- Scalability of the solution to available resources
  - Revolution Analytics' parallization routines are scalable to the resources available

Next Steps:

- Webinar hosted by Revolution Analytics on June 28th, 2012:
  - http://bit.ly/NTwebinar

Contact Information:

- Dave Humke, Northern Trust, Vice President, (dh98@ntrs.com)
- Derek Norton, Revolution Analytics, (derek.norton@revolutionanalytics.com)