Grid Computing @The Hartford…

• Using Condor in our production environment since 2004
• Computing Environment
  – Two pools (Hartford, CT and Boulder, CO)
  – Linux central managers and schedulers
  – Windows execute nodes (~7000 cores)
  – CycleServer from Cycle Computing LLC
• Workload
  – Mix of off-the-shelf tools and in-house custom software
  – Actuarial modeling
  – Financial reporting
  – Compliance
  – Enterprise risk management
    – Hedging
    – Stress testing
The Challenge…

• Compress the time it takes to compute market sensitivities to enable rapid response to large market movements
  – Current compute time: ~8 hours on ~3000 cores
  – Target: A.S.A.P. (P being practical)
• Compress the time it takes to simulate our hedging program
  – Current compute time: ~5 days on ~5000 cores
  – Target: 1 day
• Create a mechanism to calculate specific sensitivities in near real time
• Support Entire Model Portfolio: ~20 models
• Maintain Accuracy and Precision
• Enterprise IT Targets
  – Reduce Datacenter Footprint
  – Reduce Costs
The Approach...(everything’s on the table)

- **Modeling**
  - Variance Reduction
  - Optimize algorithms
  - Eliminate Redundant or Un-necessary Work

- **Processes**
  - Optimize submission pipeline
  - Reduce file transfers
  - Implement Master/Worker framework

- **Models**
  - Optimize code
    - Caching
    - Dynamic scenario generation
  - CUDA/OpenCL/OpenMP

- **Infrastructure**
  - Improve storage
  - GPUs
The Plan…

- **Modeling**
  - Test convergence with low-discrepancy sequences
  - Evaluate closed-form or replicating portfolio approach
  - Remove un-necessary workload

- **Processes**
  - Interleave scenario/liability/asset submissions
  - Improve nested stochastic analysis
  - Develop Master/Worker scheduling

- **Models**
  - Port model portfolio to CUDA
  - Optimize algorithms

- **Purchase GPU Infrastructure**
  - 250 NVIDIA Tesla 2070s
The Results...

- **Modeling**
  - Convergence achieved faster with low-discrepancy sequences
    - 2x improvement
  - Removed non-essential tasks
    - 2x improvement

- **Processes**
  - Streamlined submission pipeline for scenario/liability/assets
  - Eliminated ~1TB/run of file transfer
  - Using Work Queue for Master/Worker
    - 4-6x improvement
The Results (cont.)…

• Models
  – Developed code generator for CUDA
    – Automated development and end-user automation (priceless!)
    – DirectlyCompiled Spec Models
  – Ported entire model portfolio to CUDA (GPU) and C++ (CPU)
  – 40-60x improvement

• Infrastructure
  – 125 Servers with 250 M2070s
  – 3x reduction in data center footprint
  – 50% cost reduction

• Summary
  – Success!
    – Improved Performance
    – Reduced Cost
  – Improved our long-term capabilities
What’s Next?
Complete integration of GPUs into our Condor environment

• Quickly find the GPU nodes
  – GPU = “None”
  – SLOT1_GPU =“NVIDIA”
  – SLOT2_GPU=“NVIDIA”
  – STARTD_EXPRS = $(STARTD_EXPRS), GPU

• Identify GPGPU submissions
  – +GPGPU=True

• Reserve Slots for GPGPU jobs
  – START=( ( ( SlotID < 3 ) && ( GPUGPU =?= True ) ) || ( (SlotID > 2) && (GPGPU =!= True) ) )

• Work with Todd on GPU wish list
  – Benchmarking
  – Monitoring (corrupt memory, etc.)
What’s Next?
Refine our job scheduling architecture

• Minimize Scheduling Overhead
  – Continue development on our Work Queue implementation
  – Leverage new Condor features – key_claim_idle?

• Optimize Work Distribution
  – Need to prevent starvation of fast GPU resources while still leveraging existing dedicated and scavenged CPUs
  – Integrate with CycleServer

• High-availability/disaster recovery
  – Persistent queues
  – Support for multiple resource pools
What’s Next?
Expand Condor’s footprint @The Hartford

• Condor for Server Utilization Monitoring
  – Install Condor on all servers
  – Improved reporting and,
  – Foot-in-the-door for scavenging!
• Condor in the Cloud
• Condor Interoperability (MS HPC Server)
• Evangelize Condor to ISVs
Thank you!

Bob Nordlund
Enterprise Risk Management Technology
The Hartford
robert.nordlund@thehartford.com