



QuantCatalyst

Delivering High Performance for Financial Models and Risk Analytics

September 2008

Risk Breakfast London

Dr D. Egloff
daniel.egloff@quantcatalyst.com

QuantCatalyst Inc.

- Technology and software firm
- Incorporated in Zürich
- Focus on high-tech products based on modern many-core technology
 - Software engineering
 - High performance and distributed computing
 - Parallel and GPU computing
 - Hardware accelerated programming of financial models
- Founding members
 - Dr Daniel Egloff
 - Michael Gauckler
 - d-fine

Contents

01 Trends

Computational Finance and Technology

02 Cluster and Grids

Credit Portfolio Analytics, Trading Floor Acceleration

03 GPU

GPUs and CPUs, Taking Advantage of Many-Core GPUs, PricingCatalyst™

04 Wrap Up

Trends in Finance IT

- Computing capacity is mission critical
 - Fast and accurate pricing for traders and market makers
 - Hedging of derivative positions
 - Realistic risk quantification for all non-linear positions
 - Risk simulations at portfolio level, e.g. VaR of large book of structured products
 - Pre-deal check and intraday risk management
 - High volume processing
 - Algorithmic trading
- The financial services industry crucially depends on computing power

Trends in Systems Engineering

Because of the “frequency wall” the industry is going parallel:

- Modern desktop and server infrastructure **evolves** towards parallel systems
 - Basis: classic CPU architecture, advent of multi-core CPUs
 - **HPC clusters and grids** yield approx linear scalability
- New graphics processing units (GPUs) display **revolutionary** performance metrics
 - The GPU board itself is already massively parallel (200-300 cores)
 - Originally developed for the gaming and video industries with a strong focus on numerical performance, now suitable for **generic applications**
 - New programming model and tools for scalable parallel programming
 - NVIDIA CUDA
 - ATI Brook+, CTM and CAL
 - Performance multiplication of **30-50 per GPU**



Image:
Tesla C870
card with one G80 GPU

Contents

01 Trends

Computational Finance and Technology

02 Cluster and Grids

Credit Portfolio Analytics, Trading Floor Acceleration

03 GPU

GPUs and CPUs, Taking Advantage of Many-Core GPUs, PricingCatalyst™

04 Wrap Up

Clusters and Grids

Exploiting HPC grid computing in several key projects

- Distributed Credit portfolio risk engine
- Real time pricing of structured equity baskets for trading and quoting

Pros

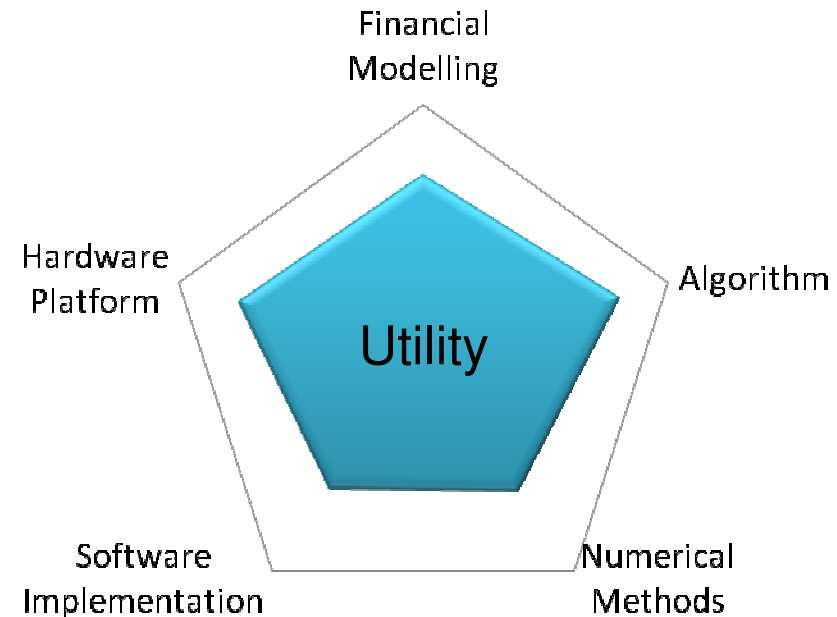
- Established technology since almost 15 years
 - First Beowulf cluster in 1993
 - MPI 1.2 in 1994
- Used across many industries
- Commodity hardware and communication technology
- MPI Message passing standard well supported, also by high-end interconnects
- Several grid engines available for resource scheduling and planning
- Lot's of open source resources

Cons

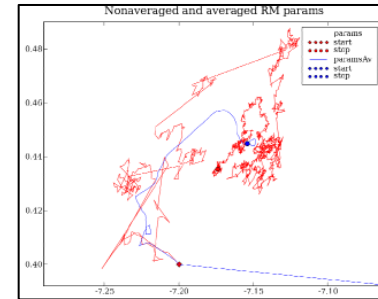
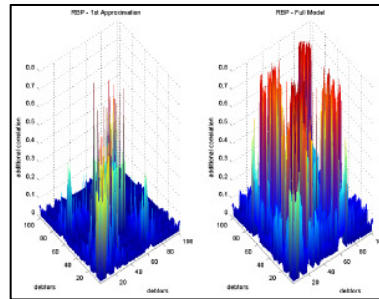
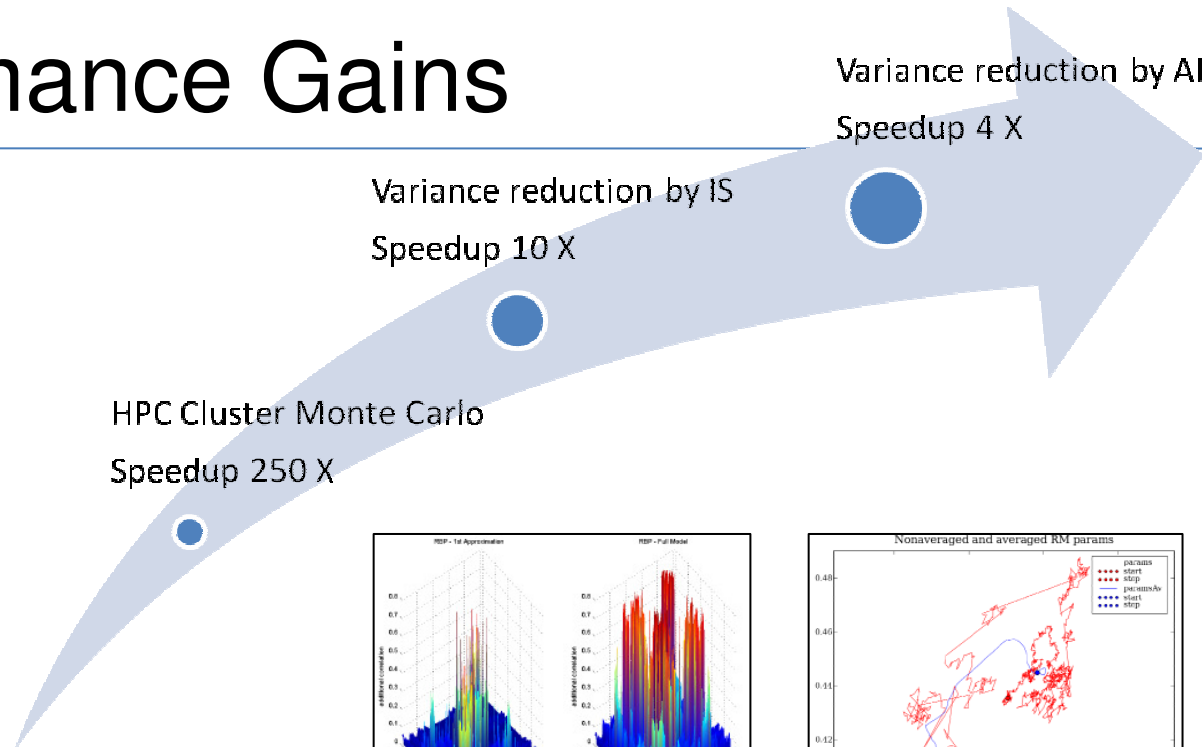
- Message passing requires abstraction layer for smooth usage with C++ (such as Boost.MPI)
- Distributed computing introduces latency and IO bandwidth issues
- Distributed address spaces complicate exploiting data locality
- Large clusters and grid face operational issues
- Power consumption, heat dissipation, and floor space
- Expensive to operate

Boosting Multiple Dimensions

- Financial modelling
 - Bespoke credit portfolio model
 - Contagion and feedback effects
 - Advanced calibration
 - Economic capital and marginal risk contributions
- Algorithm
 - Monte Carlo simulation
 - Variance reduction through adaptive importance sampling
- Numerical methods
 - Incremental statistics
 - Parallel random numbers
- Software implementation
 - C++ and Python hybrid
 - MPI through Boost.MPI abstraction
 - Aspect oriented design of financial instruments
- Hardware
 - Intel based Linux cluster running GNU/Debian
 - 250 calculation nodes



Performance Gains



Calculation accuracy

- More samples
- More extreme quantiles

Model sophistication

- Risk factor coverage
- Multi-period
- Cash flow details

Transaction volume

- Number of counterparties and contracts
- Portfolio hierarchies

Contents

01 Trends

Computational Finance and Technology

02 Cluster and Grids

Credit Portfolio Analytics, Trading Floor Acceleration

03 GPU

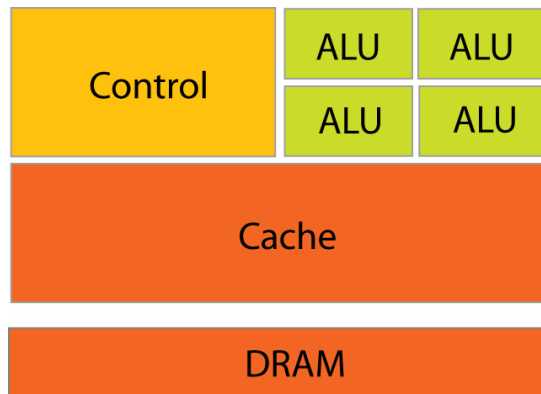
GPUs and CPUs, Taking Advantage of Many-Core GPUs, PricingCatalyst™

04 Wrap Up

CPU and GPU Compared

CPU

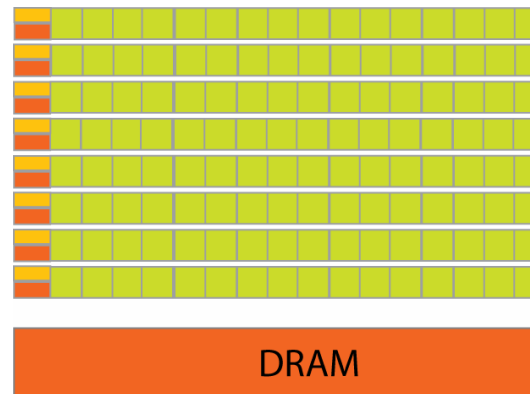
- Hand full of processing cores
- Significant amount of transistors for hardware managed cache
- Extensive control logic for pipelining, branch prediction



CPU

GPU

- Hundreds of processing cores
- Most transistors for ALUs
- Small amount of cache
- Close to ALU memory
- Hardware thread management

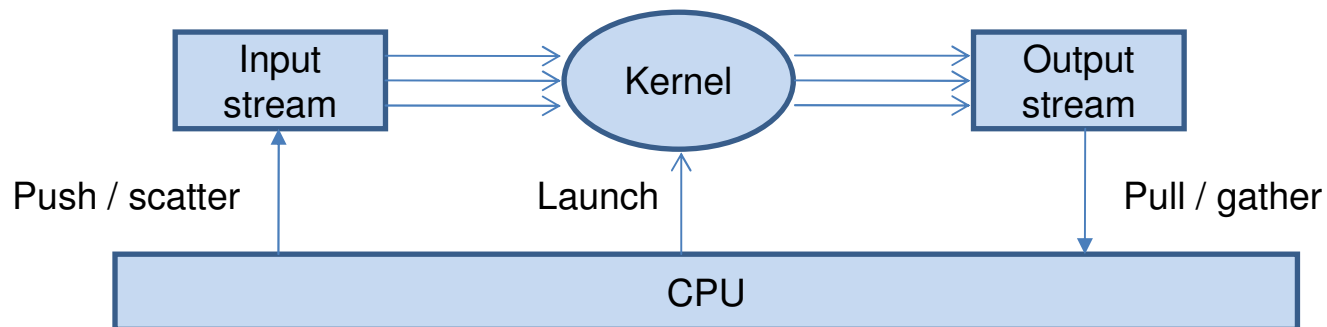


GPU

GPU Details

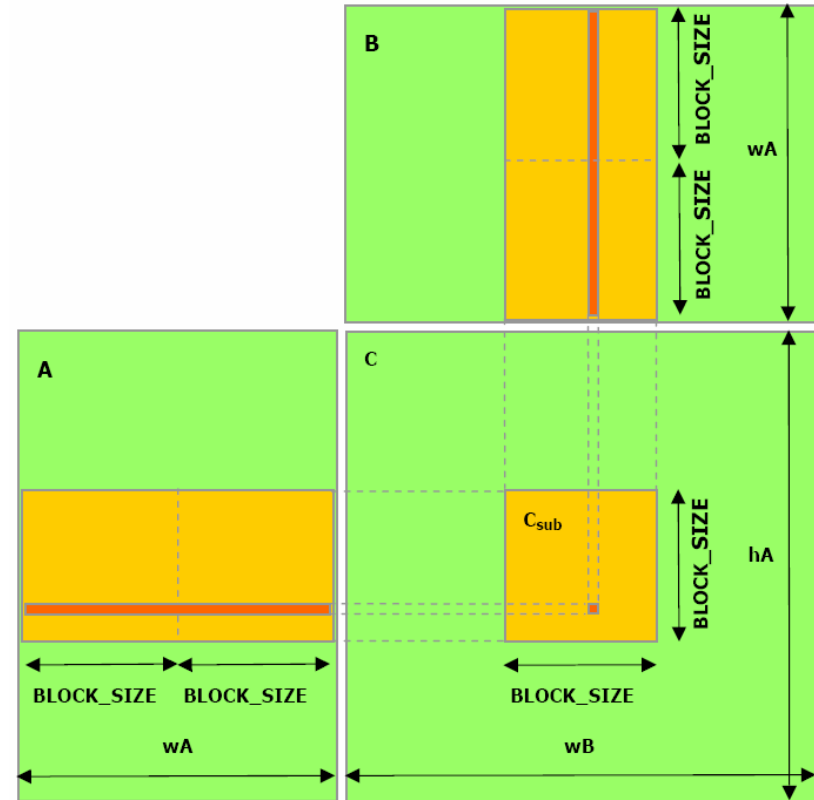
- GPUs support high throughput stream processing
 - Parallel computing with separate input and output data
 - Concurrent execution of thousands of independent threads on multiple processing units
 - Each thread is executing the same code on different data
- Favors workloads with
 - High arithmetic intensity, i.e. large number of arithmetic ops per I/O ops
 - Data parallelism
 - Data locality and low data reuse

Schematic stream computing



GPU Details

- What's about workloads with high data reuse?
- High data reuse requires fast close to ALU memory
 - Like a scratchpad for frequently used data
- Each vendor has its own memory architecture
 - NVIDIA: shared memory per multiprocessor (8 cores), managed by application
 - ATI/AMD: regular L1 and L2 caches
- Application example: matrix multiplication



Example: Nvidia Tesla Architecture

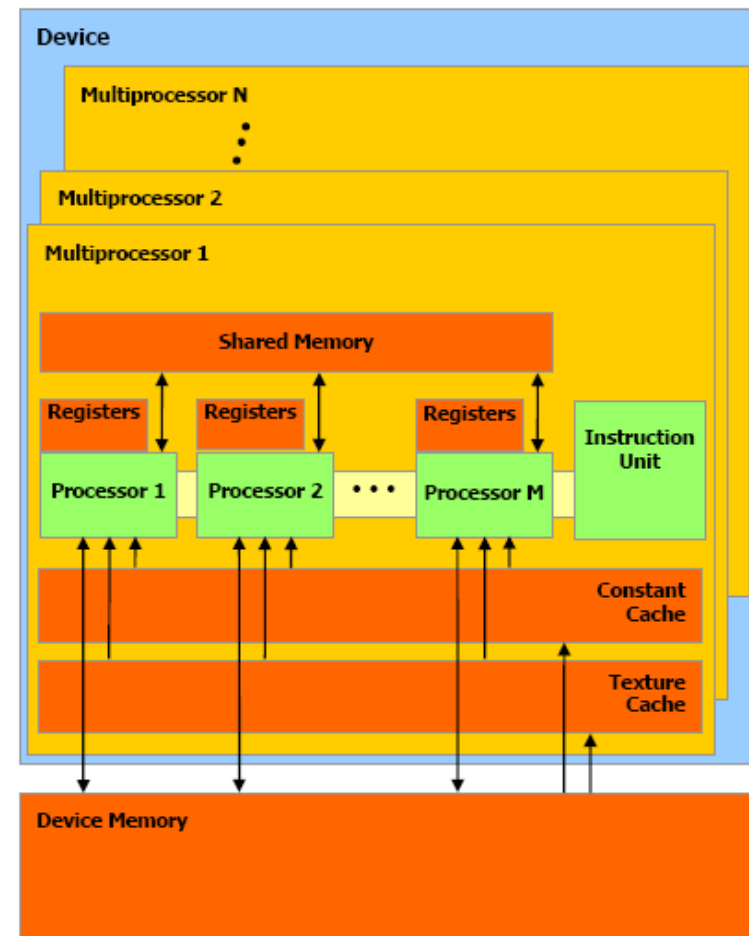
Shared memory for threads running on a multi-processor

Pros

- Explicitly managed by programmer, hence useful for memory intensive applications with complex access pattern
- Multiple levels of parallelism add flexibility and unlock additional performance potential
- Fast concurrent access from each thread in a block
- Synchronization and communication between threads in same block

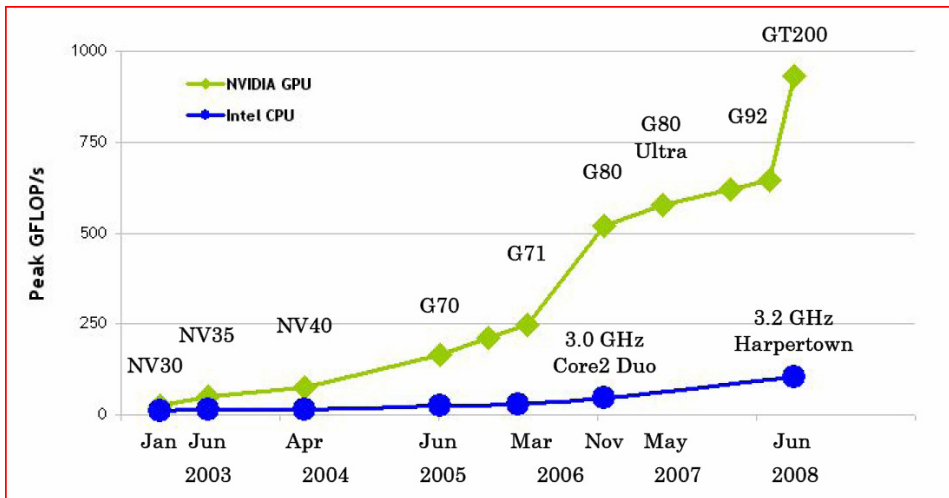
Cons

- New technology
 - CUDA 1.0 released in February 2007
- Explicit memory management and multiple parallelization levels complicate application development significantly
- Fast ALU memory relatively small, just 16K per multi-processor

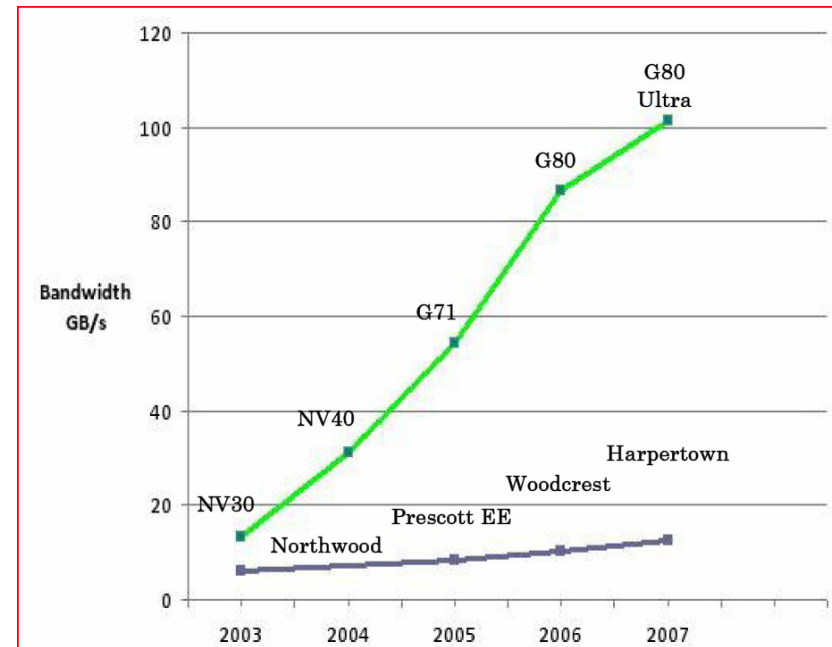


Performance, Bandwidth and Power

- Massively parallel architecture
 - Over 300 GFLOPS/GPU on 128 Cores @ 1.3 GHz
 - On board memory 1.5 GB @ 80 GB/s
 - Power efficient: 4 Tesla deliver 1.2 TFLOPS @ 550 – 800 Watt
 - GT200 with 240 Cores, 4 GB, double precision support



GT200 = GeForce GTX 280 G71 = GeForce 7900 GTX NV35 = GeForce FX 5950 Ultra
 G92 = GeForce 9800 GTX G70 = GeForce 7800 GTX NV30 = GeForce FX 5800
 G80 = GeForce 8800 GTX NV40 = GeForce 6800 Ultra



Nvidia's and "going parallel"

- Specialized manufacturer of graphics processor technology, first dual-core GPU released in 1998
- Many-core technology developed for graphics rendering is accessible for high-performance computing in finance
 - CUDA effectively turns NVIDIA GPUs into open architectures
 - Best software stack for general purpose GPU programming
- Consumer market success drives investments
 - Over 50m CUDA-enabled GPUs in the market, selling a further 10-15m per month
 - Guaranteed future technology development
 - Sustained performance increase

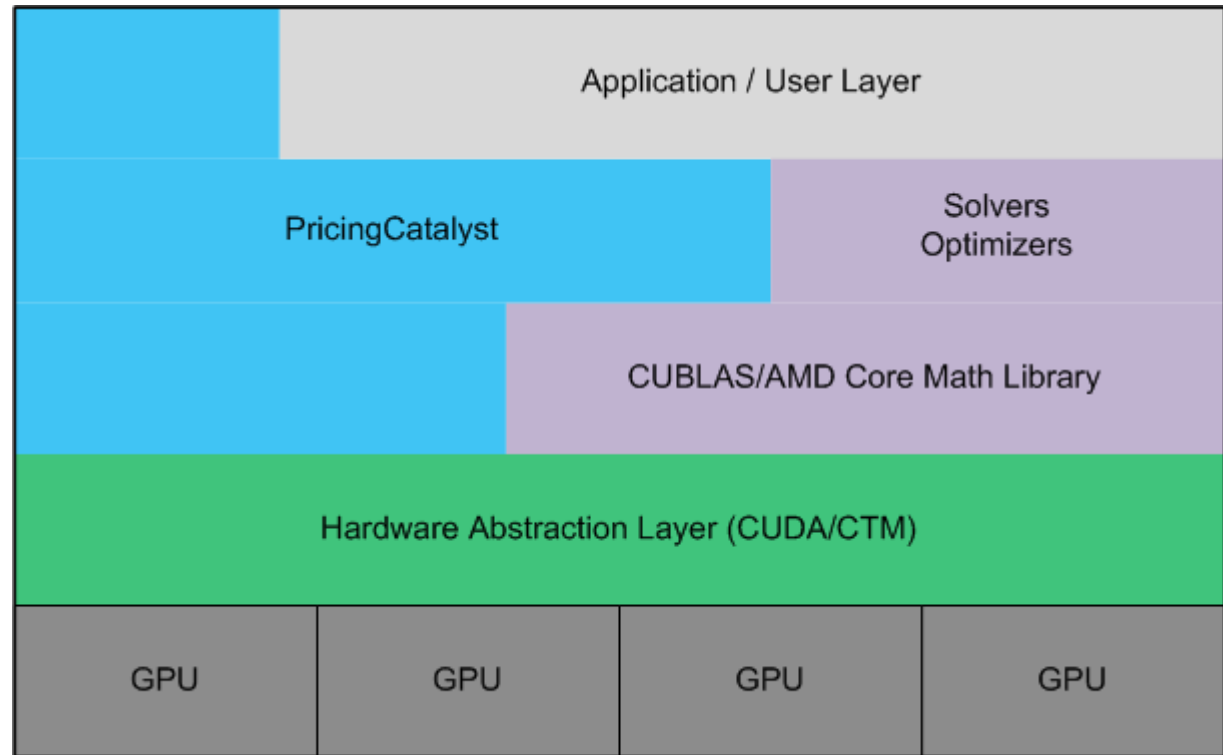
Taking advantage of the many-cores

The industry is searching for the right abstraction of the many-cores

- Requirements
 - Utilization of available bandwidth to bring data to the processors
 - Optimally manage the close to ALU memory
 - Exploit all compile time data access patterns
 - Handle dynamic data access pattern with shared memory and device memory
- Long-term solution: Cross platform (GPU, Cell, Larabee) tool chain with programming language that can express
 - parallelism
 - data locality
- Short-term solution:
 - Performance primitive libraries taking advantage of the new hardware
 - Scripting languages with interpreter using performance-optimized runtime environments
 - Programming frameworks

PricingCatalyst™ Vertical Approach

- Integrate all layers from hardware management to financial engineering building blocks
- Consistent modelling and application framework

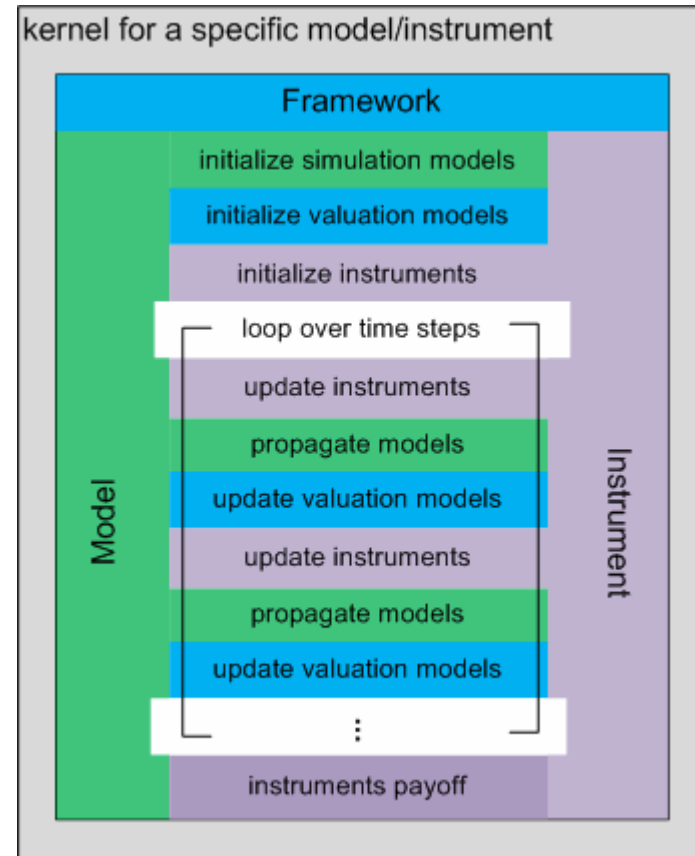


PricingCatalyst™ Example

PricingCatalyst framework

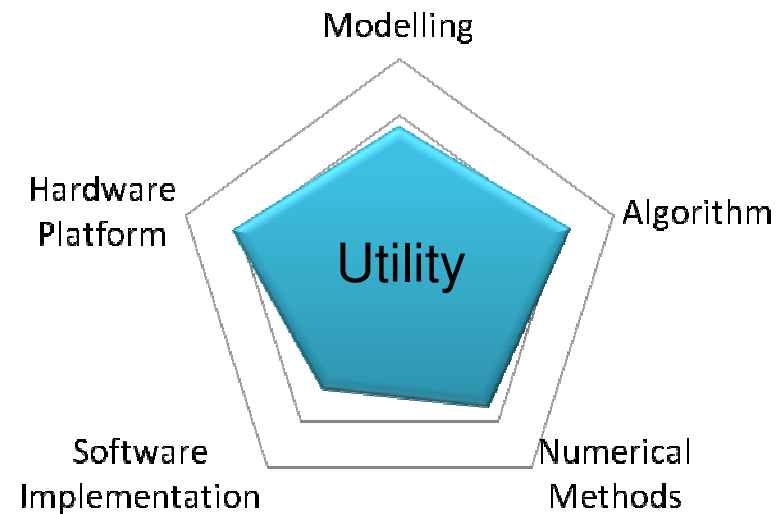
- Hardware abstraction by software layers provided as source code
- Enforces optimal data access pattern at compile time
- Allows the developer to infuse code fragments in a structured way

PricingCatalyst Monte Carlo GPU Kernel Framework



Boosting Multiple Dimensions

- Modelling and representation
 - Financial engineering
 - Pre-processing
 - Dependency evaluation
- Algorithm
 - Path exchanging for basket Greeks
 - Variance reduction, e.g. through Importance sampling
- Numerical methods
 - Optimized matrix multiplications
 - Parallel equation solvers
 - Fast inverse problems with SOR
- Software implementation
 - Expression templates
 - Loop unrolling
 - Compile-time vs. run-time dispatching
- Hardware
 - Mapping to many-core architecture
 - Exploiting data locality and data parallel execution
 - Constant and texture memory usage
 - Optimal management of close-to-ALU memory



Advantages of Vertical Approach

- Access to all layers
 - No black-box
 - Prioritize performance critical aspects with best cost / benefit ratio
 - No restrictions on implementing non-standard financial instruments and models
- Based on commonly available hardware and software components
- No third-party software vendor lock-in due to proprietary layer
 - Cilk++, non-standard language keywords, Hyperobject Library, Cilk++ runtime
 - QuIC, non-standard QuIC-script and proprietary runtime environment
 - Rapidmind, non-standard language keywords and proprietary just-in-time backend compiler

Performance Gains

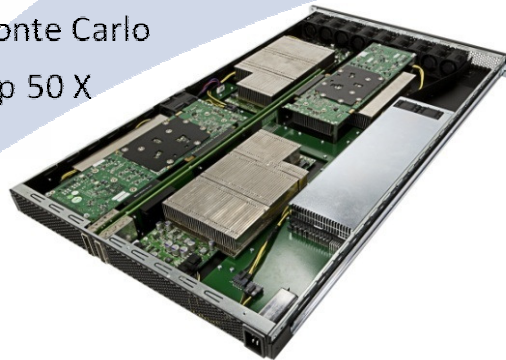
Model representation and variance reduction

Speedup 4 X – 10 X



GPU Monte Carlo
Speedup 50 X

Multi-GPU support
Speedup 4X – 8 X



S-up	$S_0 + dS_0$	$S_1 + dS_1$	$S_2 + dS_2$	} 2 baskets simulated
Price	S_0	S_1	S_2	
dP/dS0	$S_0 + dS_0$	S_1	S_2	} 4 baskets valued
dP/dS1	S_0	$S_1 + dS_1$	S_2	
dP/dS2	S_0	S_1	$S_2 + dS_2$	

Calculation accuracy

- More samples
- Stabilized sensitivities

Model sophistication

- Next generation models
- More sensitivities

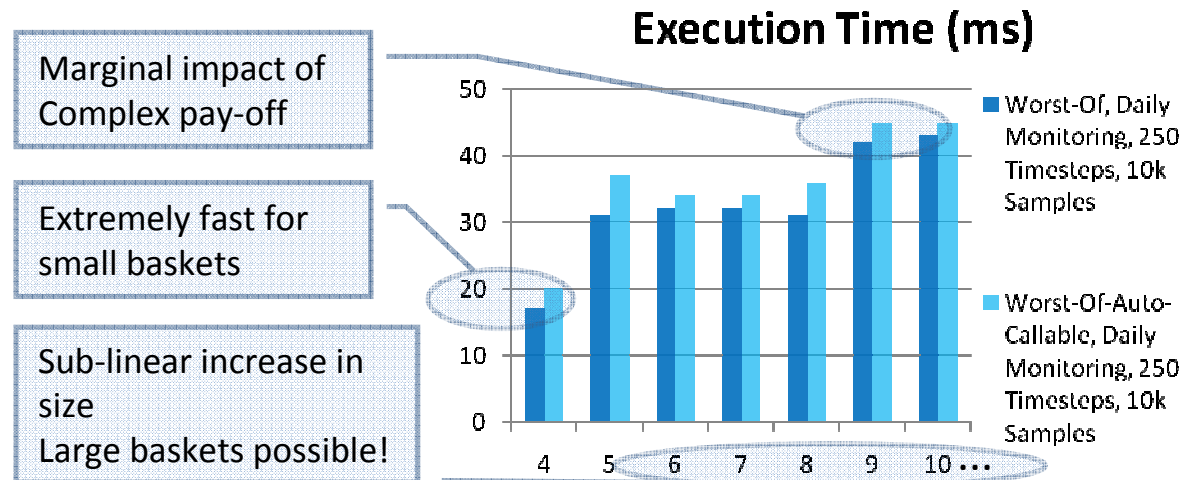
Calculation latency

- Real time pricing
- Full revaluation risk management

Valuation of Multi Asset Option

- Product valuation: Barrier reverse convertible bond on worst of multi-equity basket
 - Two different pay-off variants: without / with auto-callability
 - Basket varied in size: from 4 to 10 underlying equities
 - Including cash dividends
- Correlated local volatility model, inhomogeneous time grid, state dependent drift & volatility
- **10'000** Monte Carlo samples x 250 time steps (1Y daily monitoring)
- **Industry timing benchmark**: 2 – 10 seconds (**2000-10'000 ms**) for only 4 equities

Results on single GPU:



And with **100'000** MC samples (10x): Only 7x run time!

Contents

01 Trends

Computational Finance and Technology

02 Cluster and Grids

Credit Portfolio Analytics, Trading Floor Acceleration

03 GPU

GPUs and CPUs,

04 Wrap Up

Contact

Dr Daniel Egloff

daniel.egloff@quantcatalyst.com

Phone: +41 44 520 01 17

Mobile: +41 79 430 03 61

Michael Gauckler

michael.gauckler@quantcatalyst.com

Phone: +41 44 520 01 18

Mobile: +41 76 440 37 28

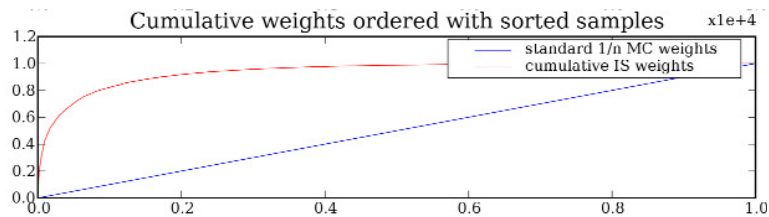
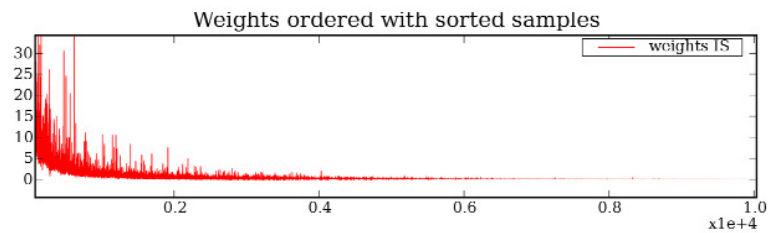
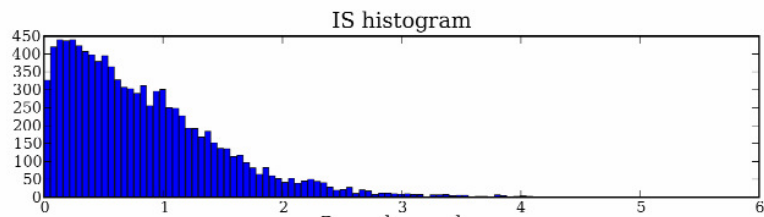
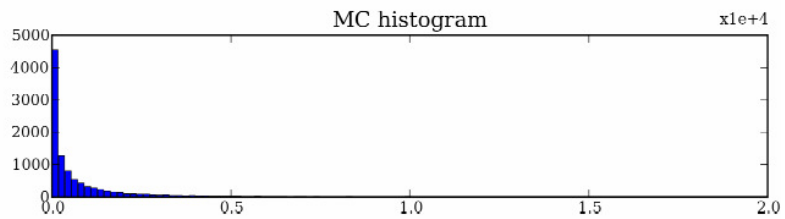


Quant Catalyst

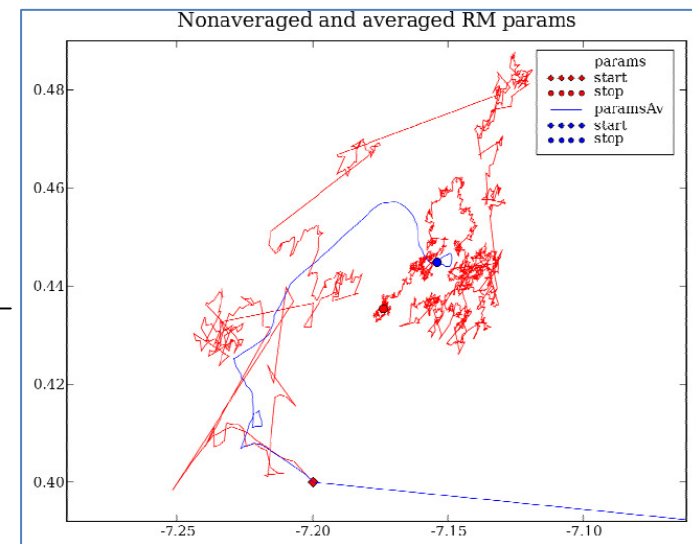
Hardturmstrasse 101

CH-8005 Zürich

Adaptive IS



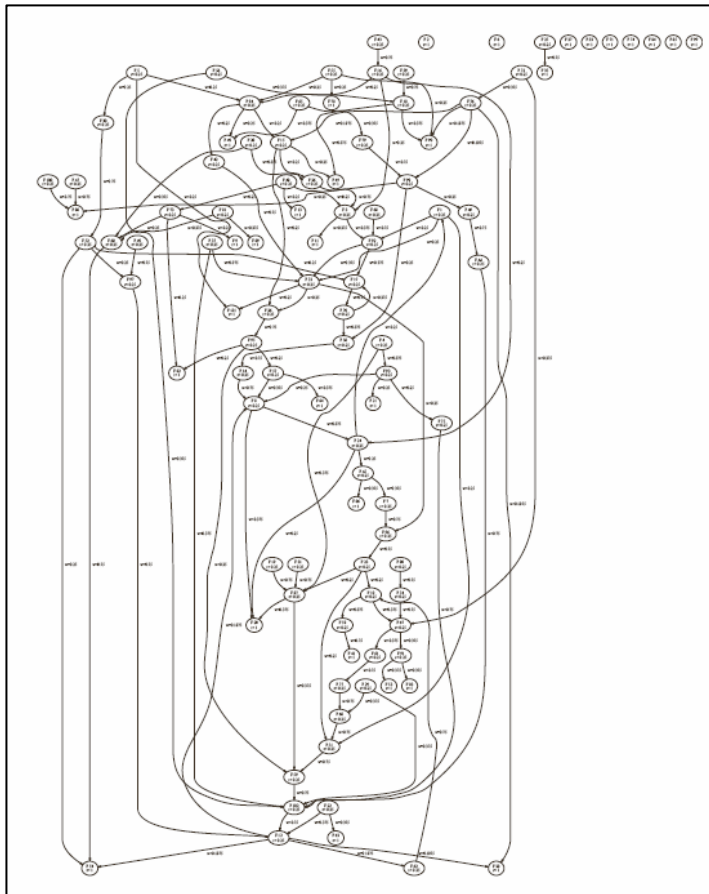
Stochastic adaptive search of optimal sampling parameters



Dimension reduction by PCA
Polyak averaging

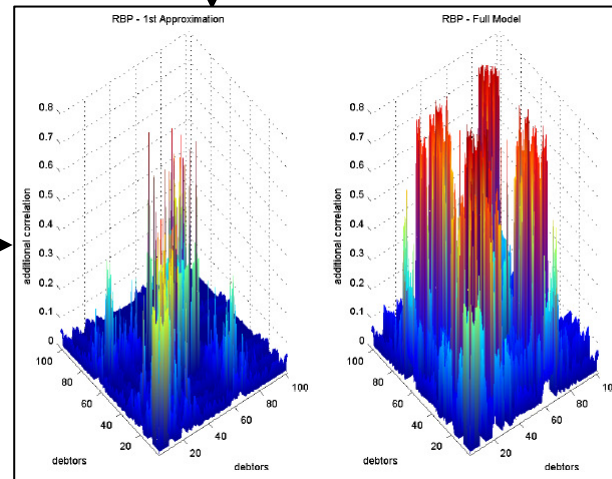
Model Quality

Dependency through microstructure effects



Graph modelling dependence and weights

Asset returns with higher correlations



Reference: Egloff et al. A Simple Model of Credit Contagion, JBF, 2006