

Visual Domain Specific Languages for Actuarial Models: An Industrial Experience Report

Workshop on Domain Specific Languages for Financial Systems

*ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems
(MODELS 2013), Miami, FL*

Aon Benfield Securities, Inc.
Annuity Solutions Group (ASG)

October 1, 2013



This document is the confidential property of Aon Benfield Securities, Inc. (“Aon”), has been prepared by Aon for informational purposes only and is intended only for the designated recipient. As a condition to reviewing this document, the recipient agrees that without the prior written consent of Aon, which may be withheld for any reason, the recipient will not copy the document or any of its contents, and will not disclose or disseminate the document or any of its contents to (i) any third party, or (ii) any person within recipient's organization who does not have a need to know in connection with the express business purpose for which the document is being provided to recipient. If the recipient is legally compelled to disclose this document or any of its contents, it will promptly give notice to Aon and will reasonably cooperate with Aon in any attempts by Aon to obtain a protective order or otherwise limit disclosure. Upon request by Aon, the recipient will promptly return or destroy the document and any copies it has made with Aon's consent (as described above), provided that recipient may maintain, in strict confidentiality, such copy or copies as required by law or regulation. Aon makes no representation of any kind as to the suitability of the products or services described in this document for any entity in any jurisdiction. The recipient is advised to undertake its own review of the legal, regulatory, tax, accounting and actuarial implications of the products and services described in this document, as Aon does not provide legal, regulatory, tax, accounting or actuarial opinions. This document should not be considered an offer to sell or a solicitation of any agreement to purchase any security. All securities advice, products or services are offered solely through Aon Benfield Securities, Inc. or an appropriately licensed affiliate.

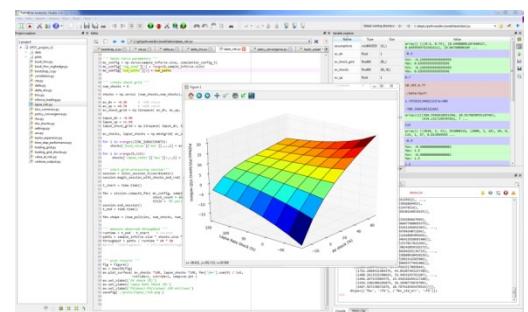
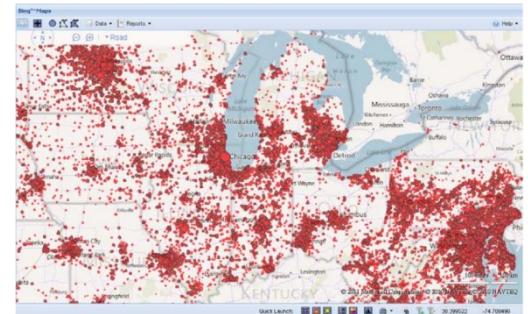


Aon Benfield

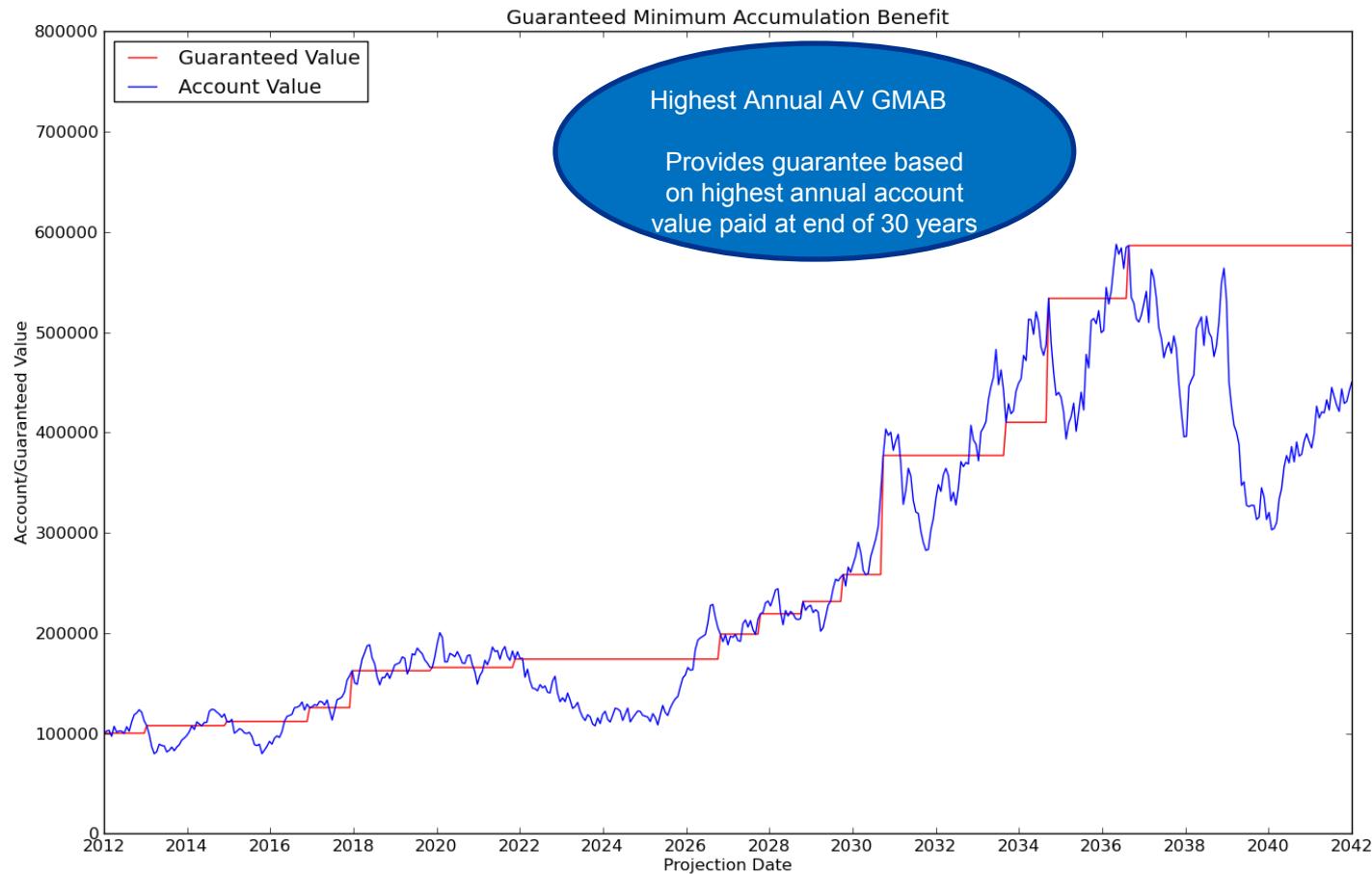
- **Aon Benfield**, a division of Aon plc (NYSE: AON), is the world's leading reinsurance intermediary and full-service capital advisor. We empower our clients to better understand, manage and transfer risk through innovative solutions and personalized access to all forms of global reinsurance capital across treaty, facultative and capital markets. As a trusted advocate, we deliver local reach to the world's markets, and an unparalleled investment in innovative analytics. With more than 80 offices in 50 countries, our worldwide client base has access to the broadest portfolio of integrated capital solutions and services.

- **Aon Benfield Analytics**

- Aon Benfield Analytics offers clients industry-leading catastrophe management, actuarial, rating agency advisory and risk and capital strategy expertise.
- Sample risk analytics products
 - **PathWise**
 - ReMetrica
 - ImpactOnDemand
 - CatScore



Industry Overview – Variable Annuities



Industry Computational Challenges

- **Business end-users focus**
 - ◆ Users are Quantitative Analysts, Actuaries, Traders, Risk Managers, etc
 - ◆ The right tools must focus on the end-user requirements
- **Business logic and systems code must be continually adapted to changes**
 - ◆ Changing models, financial products, market conditions, and regulatory requirements
 - ◆ Changing technologies (Multi-Core, Cell Broadband Engine, GPUs, etc)
- **High Computational Throughput is required**
 - ◆ Large-scale real-time Monte Carlo simulations (Support Hedging Programs)
 - ◆ Nested simulations (Hedging Back Testing, Capital, Valuation)
 - ◆ High end-user productivity (not waiting for huge runs to complete)
- **Mission Critical Operations**
 - ◆ The intended use of such systems is mission critical
 - ◆ System failures or bugs can be catastrophic for business users
 - ◆ Automation and auditability are very important issues

Industry Computational Challenges

- **Business logic and systems code must be continually adapted to changes**
 - Change is constant
 - Financial modeling innovation
 - Financial products innovation
 - Evolving market conditions
 - Changing regulatory requirements
 - Technological innovation
 - Traditional approaches
 - Enterprise IT systems slow to adapt
 - Shadow IT systems fill the gaps –
patchwork of end-user developed, manually
operated spreadsheets (potentially
thousands of interlinked spreadsheets)
 - Slow, costly, error-prone

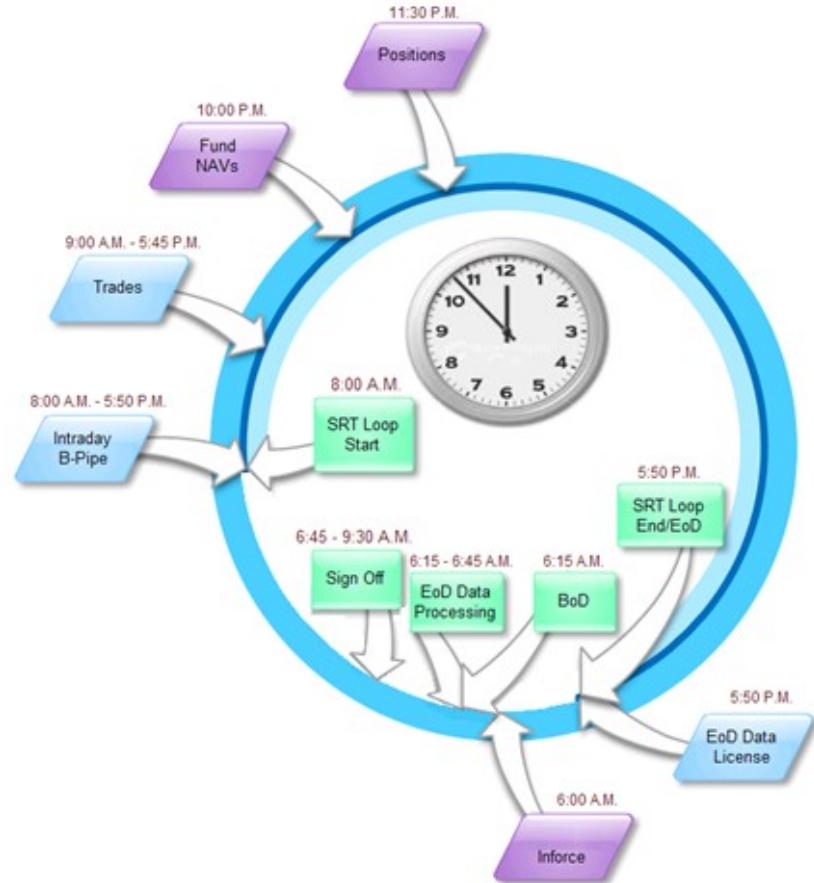


"There it is! I've isolated the origin of the firm's demise."

Industry Computational Challenges

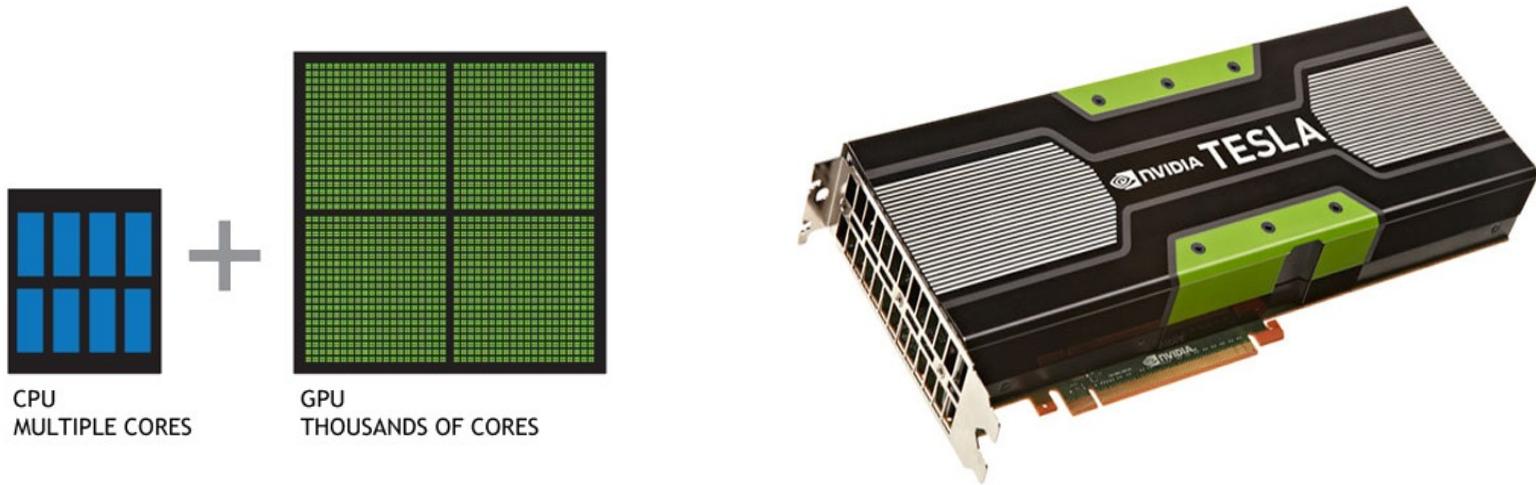
- Mission Critical Operations
 - Requirements

- High performance, integrated real-time analytics
- Complex business data-flow management
- Job scheduling
- Fault tolerance / failover
- Operational workflows
- Reporting presentation layers
- Audit trails
- Monitoring and Error Reporting



GPU Computing

- GPU (Graphics Processing Units) are specialized processors that can be used to speed-up parallel computing problems, such as Monte Carlo simulation



- **Implications for Variable Annuities Modeling:**

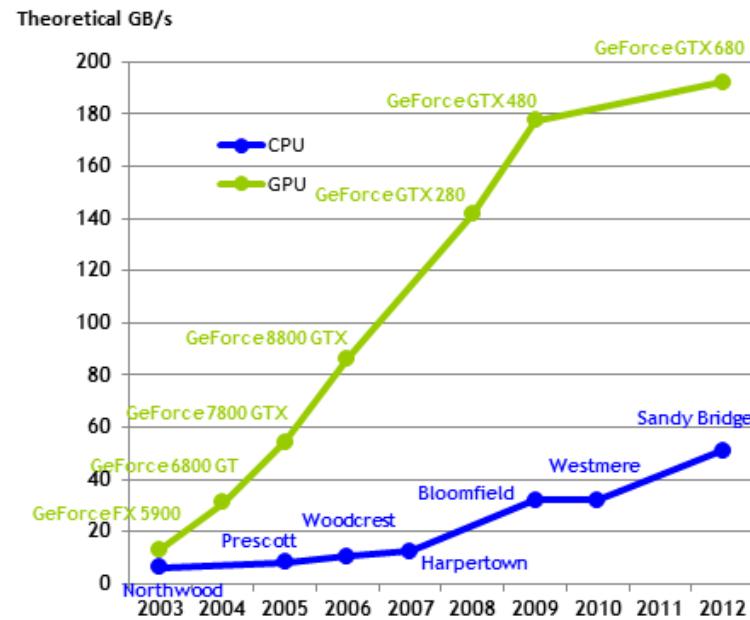
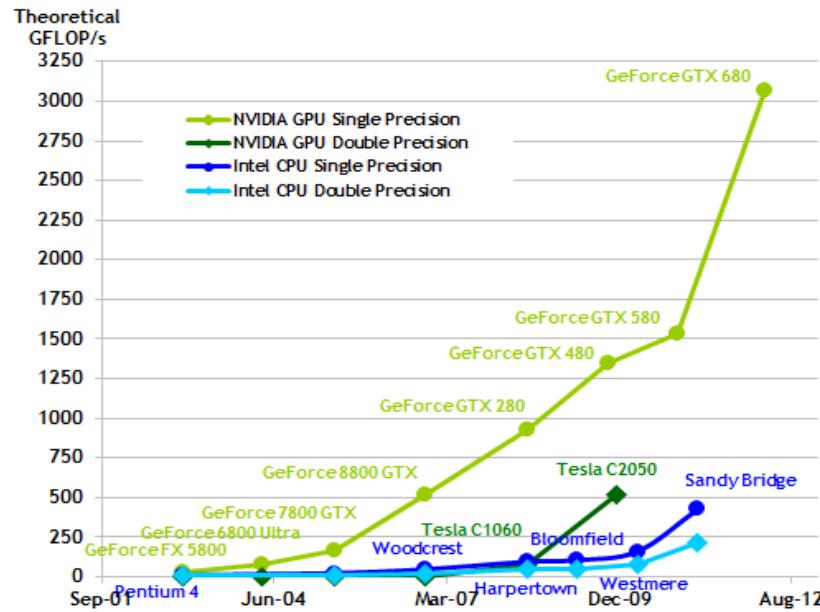
- **50-500x** speed improvements for stochastic models, when compared to equivalent CPU-based software

Above: Tesla K10 GPU module, containing 3,072 cores

Source: <http://www.nvidia.ca/object/what-is-gpu-computing.html>

GPU Computing

- Performance gap between CPUs and GPUs continues to increase rapidly



- Implications for Variable Annuities Modeling:
 - Cost of GPU-based grids is increasingly lower than cost of equivalent CPU-based grids
 - Complex optimizations for GPUs are increasingly important for VA modeling software (not simply a matter of farming out small sections of legacy code to GPUs)

GPU Computing

- General Purpose Computing performance on GPUs continues to increase rapidly

	Tesla M2050	Tesla K10	Improvement
Release Date	May-10	Dec-12	
Cores	448 cores	3072 cores	686%
Memory (GDDR5)	3GB	8GB	267%
Memory Bandwidth	148GB/s	320GB/s	216%
Single Precision Peak Performance	1.04 TFLOPS	4.58 TFLOPS	440%
Power Consumption	225W	235W	

- Tesla M2050 and Tesla K10 have similar hardware and power consumption costs
- Our VA modeling benchmarks show a 200-300% increase in efficiency (scenarios per second, per GPU or per dollar) when comparing Tesla M2050 GPUs in K10 GPUs

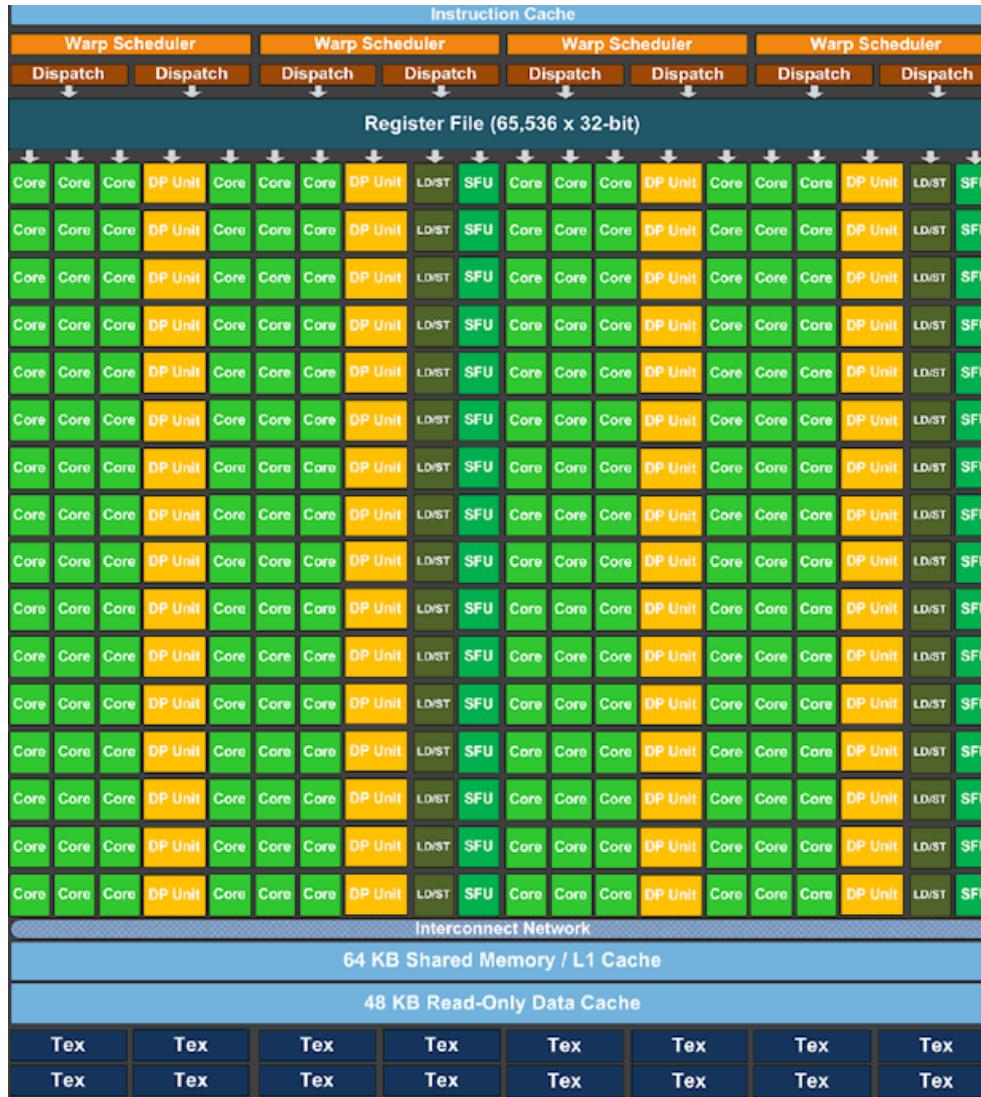
GPU Computing

NVIDIA Kepler GK110 processor



GPU Computing

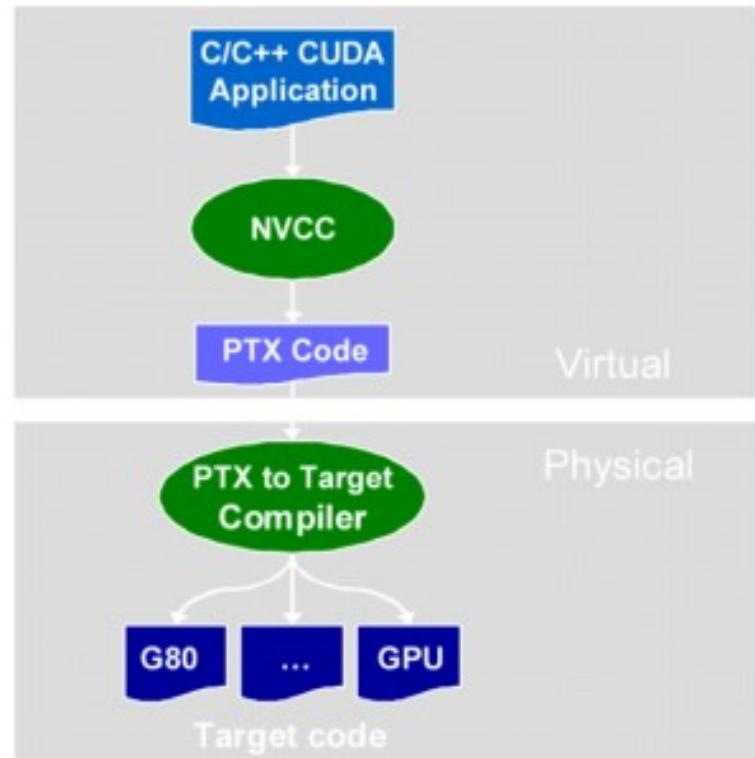
GK110 processor SMX



GPU Computing

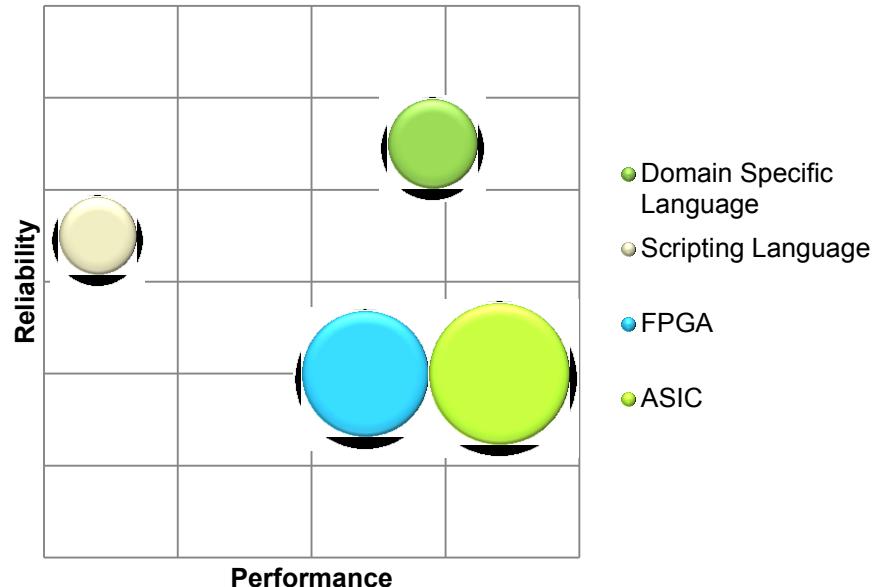
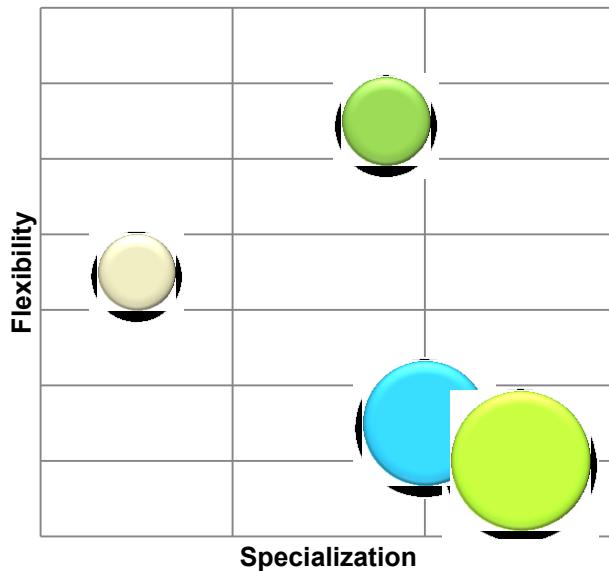
NVIDIA CUDA programming model

```
//Vector size in elements  
const int N = 1048576;  
//Vector size in bytes  
const int dataSize = N * sizeof(float);  
  
//CPU memory allocation  
float *h_A = (float *)malloc(dataSize);  
float *h_B = (float *)malloc(dataSize);  
float *h_C = (float *)malloc(dataSize);  
  
//GPU memory allocation  
float *d_A, *d_B, *d_C;  
cudaMalloc((void **)&d_A, dataSize));  
cudaMalloc((void **)&d_B, dataSize));  
cudaMalloc((void **)&d_C, dataSize));  
  
//Initialize h_A[], h_B[]...  
  
//Copy input data to GPU for processing  
cudaMemcpy(d_A, h_A, dataSize, cudaMemcpyHostToDevice) );  
cudaMemcpy(d_B, h_B, dataSize, cudaMemcpyHostToDevice) );  
  
//Run the core of N / 256 units, 256 streams each  
//Assuming that N is multiple of 256  
vectorAdd<<<N / 256, 256>>>(d_C, d_A, d_B);  
  
//Read GPU results  
cudaMemcpy(h_C, d_C, dataSize, cudaMemcpyDeviceToHost) );
```



Domain Specific Languages

Example HPC Solution Trade-Offs

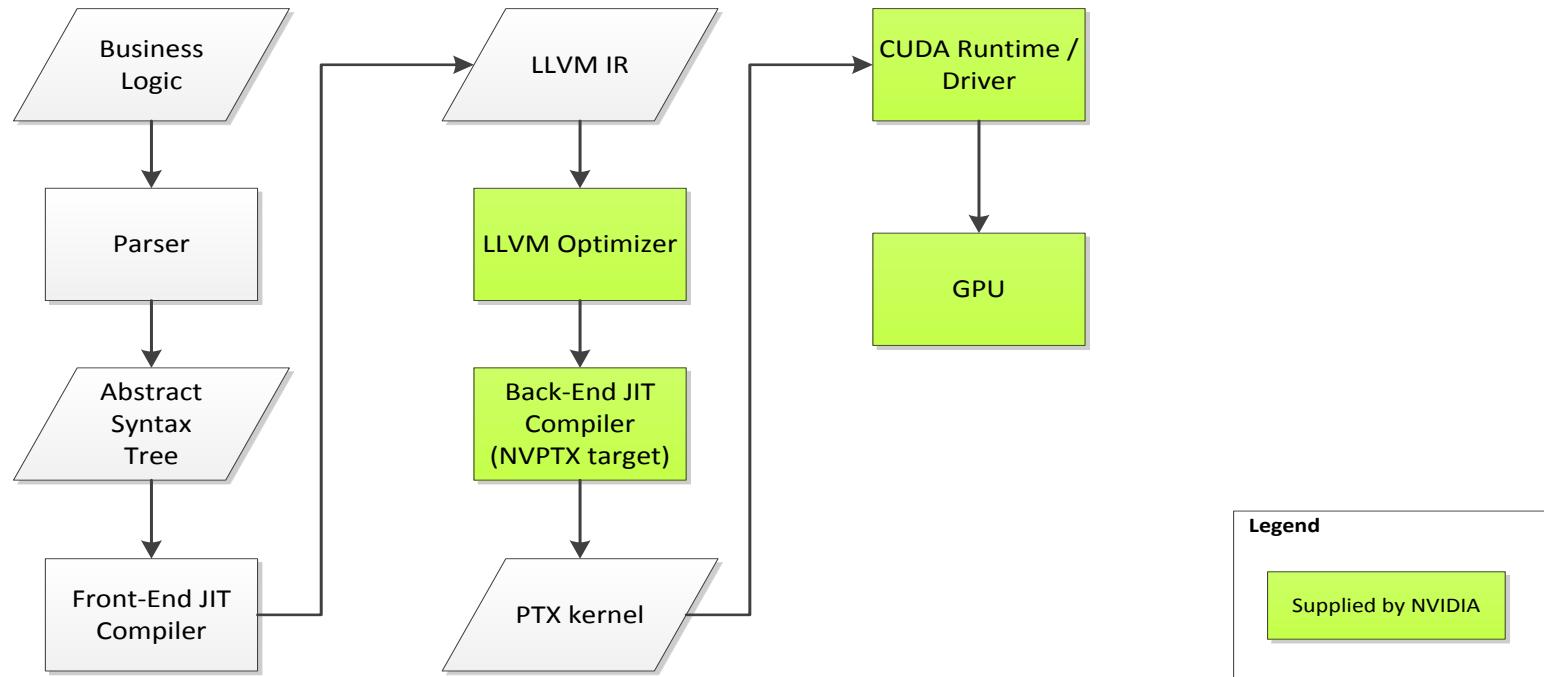


Size of bubble indicates cost (in terms of time and money) of solution

- **Flexibility** – ability to rapidly make changes
- **Specialization** – code specialized to specific hardware
- **Performance** – run-time performance of the solution
- **Reliability** – probable number of bugs in a large system

Domain Specific Languages

GPU DSL compiler architecture



Domain Specific Languages

GPU DSL compiler architecture

DSL

```
foo(x0, x1, x2)
{
    return x0 + x1 * (x2 + 1.0)
}
```

LLVM IR

```
; ModuleID = 'module1'
target triple = "nvptx64"

define double @foo(double %x0, double %x1, double %x2)
{
entry:
    %x23 = alloca double
    %x12 = alloca double
    %x01 = alloca double
    store double %x0, double* %x01
    store double %x1, double* %x12
    store double %x2, double* %x23
    %x04 = load double* %x01
    %x15 = load double* %x12
    %x26 = load double* %x23
    %faddtmp = fadd double %x26, 1.000000e+00
    %fmultmp = fmul double %x15, %faddtmp
    %faddtmp7 = fadd double %x04, %fmultmp
    ret double %faddtmp7
}
```

Domain Specific Languages

GPU DSL compiler architecture

PTX

```
//  
// Generated by LLVM NVPTX Back-End  
  
.version 3.1  
.target sm_20, texmode_independent  
.address_size 64  
  
// .globl      foo  
.entry foo(  
    .param .f64 foo_param_0,  
    .param .f64 foo_param_1,  
    .param .f64 foo_param_2  
)  
{  
    .local .align 8 .b8      __local_depot0[24];  
    .reg .b64      %SP;  
    .reg .b64      %SPL;  
    .reg .pred %p<396>;  
    .reg .s16 %rc<396>;  
  
    ...
```

...

```
.reg .s16 %rs<396>;  
.reg .s32 %r<396>;  
.reg .s64 %rl<396>;  
.reg .f32 %f<396>;  
.reg .f64 %f1<396>;  
  
mov.u64      %SPL, __local_depot0;  
cvta.local.u64 %SP, %SPL;  
ld.param.f64  %f10, [foo_param_0];  
st.f64  [%SP+16], %f10;  
ld.param.f64  %f10, [foo_param_1];  
st.f64  [%SP+8], %f10;  
ld.param.f64  %f10, [foo_param_2];  
st.f64  [%SP+0], %f10;  
ld.f64  %f11, [%SP+16];  
ld.f64  %f12, [%SP+8];  
add.f64      %f10, %f10, 0d3FF000000000000;  
fma.rn.f64   %f10, %f12, %f10, %f11;  
st.param.f64  [func_retval0+0], %f10;  
ret;
```

}

PathWise Platform



PathWise Industry Recognition

The screenshot shows the homepage of Insurance & Technology magazine. At the top, there's a green header bar with the magazine's name. Below it is a navigation menu with five categories: Architecture / Infrastructure, Policy Administration & Management, Claims, Management Strategies, and Security / Risk Management. To the left of the main content area is a sidebar with links to Home, News, Blog / Opinion, Executive / Carrier Profiles, and Case Studies. The main content area features a large red title: "Standard Life's Traditional Prudence Drives Adoption of Aon Benfield's State-of-the-Art Annuity Risk Management Solution". Above the title is a small image of a magazine cover titled "INSURANCE & TECHNOLOGY" with a subtitle "January 2012 • Business Innovation Through Technology". To the right of the title is a call-to-action button: "Sign Up! Our Free Subscription Now".

Standard Life's Traditional Prudence Drives Adoption of Aon Benfield's State-of-the-Art Annuity Risk Management Solution

- Insurance and Technology Magazine

"We had ready access to risk information on a regular basis before PathWise, but now the information is refreshed frequently and we're able to make more timely decisions," says Ettles. **"Many calculations that we would have done in hours or days are now done every few minutes** — our information is up-to-date on a real-time basis and we're not taking decisions on information that is stale." Martin Ettles is a senior actuary, finance and risk management, Standard Life

PathWise Industry Recognition



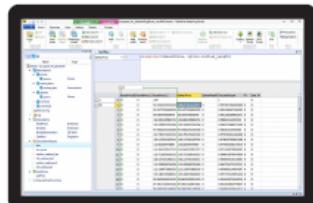
- PathWise won IDC's HPC Innovation Excellence Award in June 2012
“The new award winners and project leaders announced at ISC'12 are as follows (contact IDC for additional details about the projects):
- **GE Global Research (U.S.).**
- **Department of Defense High Performance Computing Modernization Program (U.S.).**
- **Mary Bird Perkins Cancer Center and Louisiana State University (U.S.).**
- **BGI Shenzhen (China).**
- **Aon Benfield Securities, Inc. (Canada).** Aon has developed the PathWise platform, which uses GPU-based high performance computing to enable quantitative analysts to quickly and easily express financial application kernels such as Monte Carlo simulations using domain-specific interfaces. The computational capabilities offered by the GPU-driven HPC enabled quantitative analysts to accelerate financial computations from days to minutes, with 50-100 times throughput over conventional techniques. The PathWise platform from Aon Benfield achieved an average 90% cost savings both in terms of HPC infrastructure costs and time-to-market, translating to several millions of dollars in savings. Project leader: Peter Phillips, Aamir Mohammad”

PathWise Modeling Studio

- End-user tools for High Productivity Computing

GPU Development Workflow

PathWise™ Modeling Studio



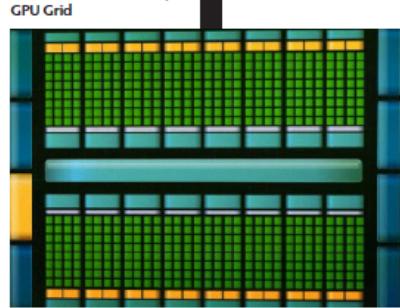
Implement Scenario Generators and Cashflow Models

Deploy Model Library and Interfaces to GPU Grid and PWAS

Generate and Compile C++ GPU Code

PathWise™ GPU Coordinator

Run Model on Grid, Collect and Analyze Results using Python Scripts



Load Data

b1010
0101010
0101010
0101010
0101010

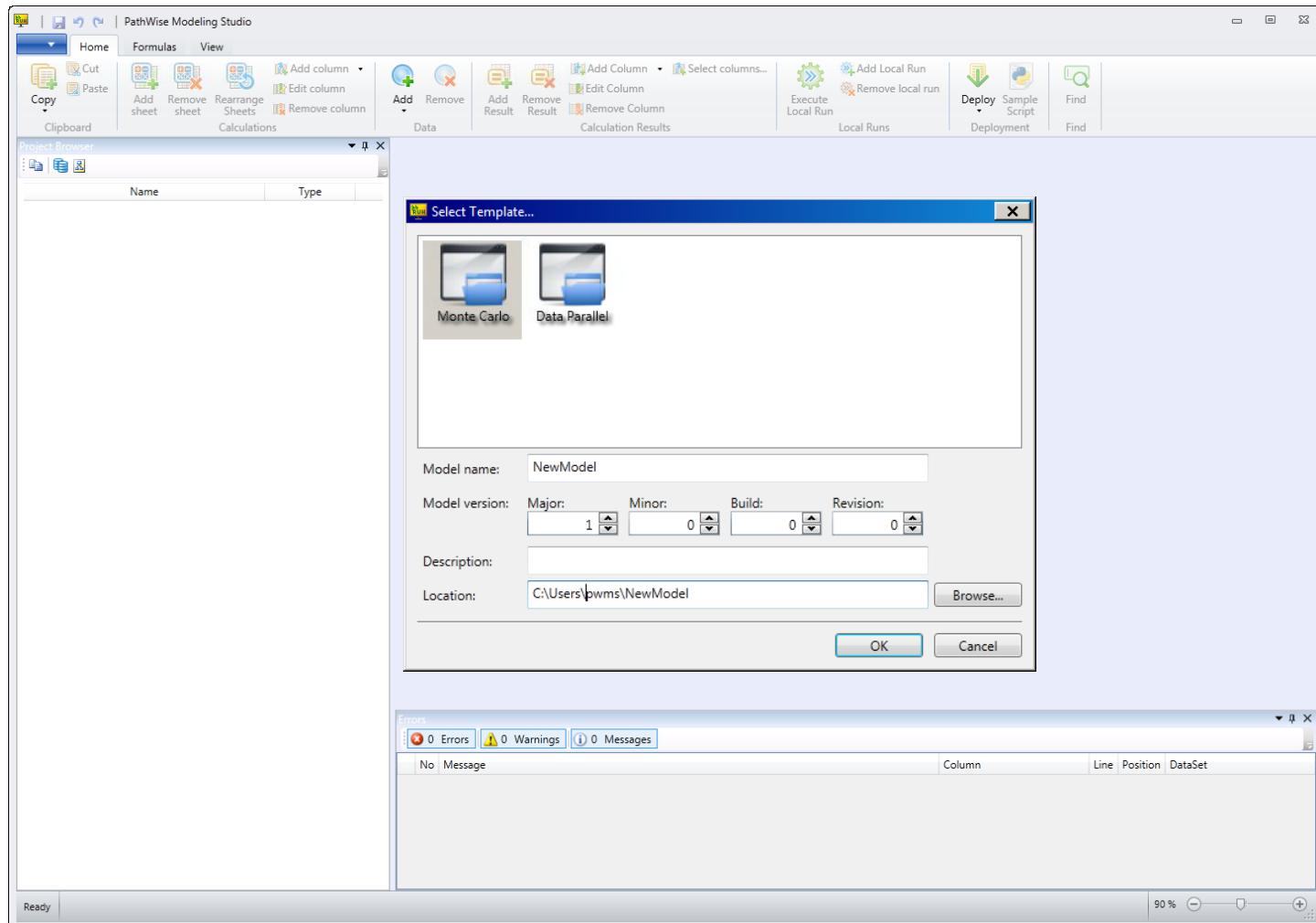


Export Data

b1010
0101010
0101010
0101010
0101010

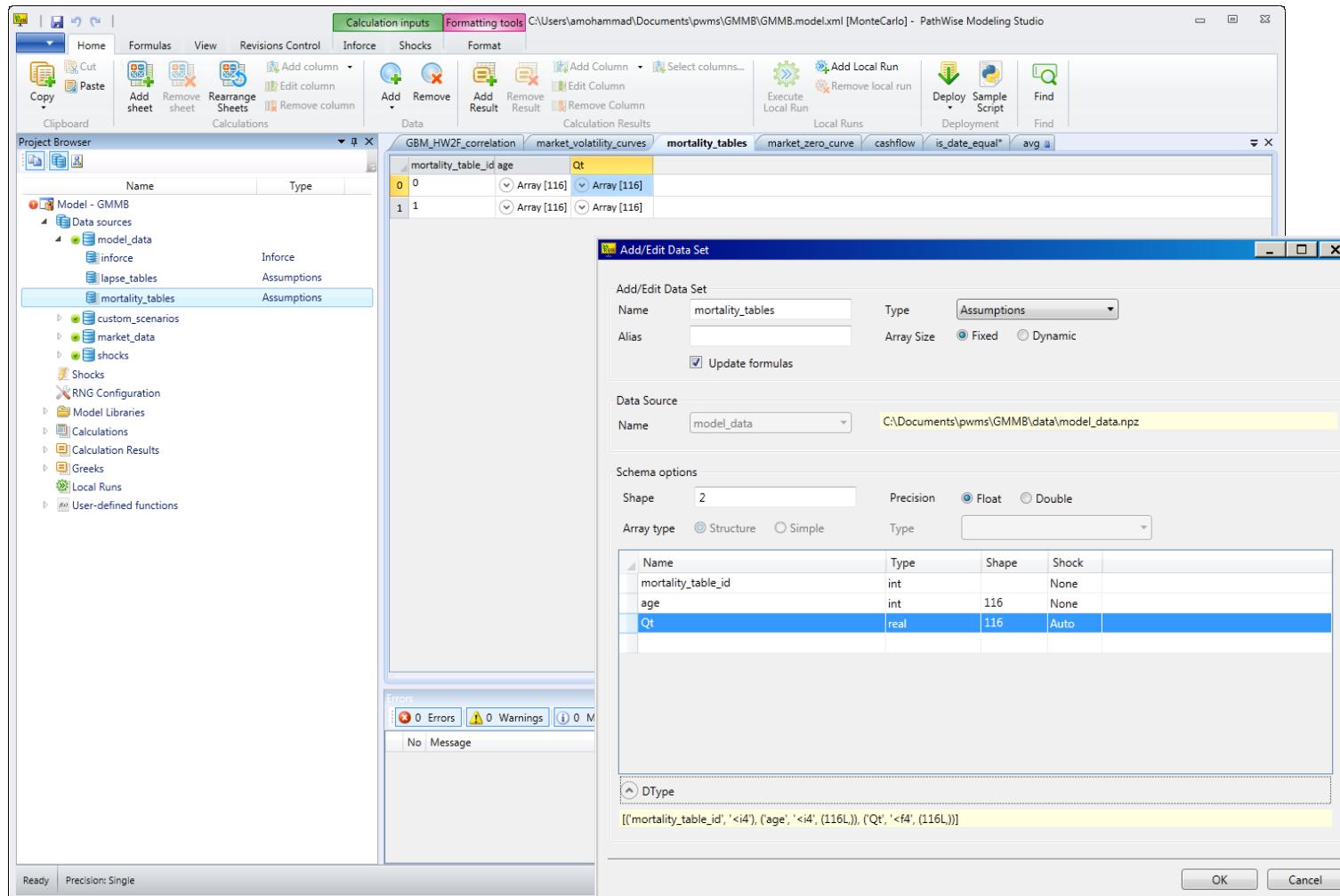
PathWise Modeling Studio

- Create a new model



PathWise Modeling Studio

- Define input data structures (customized NumPy data structures)



PathWise Modeling Studio

■ Setup Random Number Generator options

The screenshot shows the PathWise Modeling Studio interface with the 'RNG Configuration' dialog open. The dialog is titled 'RNG Configuration' and contains several sections:

- Configuration:** Num paths =inforce.num_paths, Skip rows =0, Current path index < 0.
- Uniform Generator Params:** Uniform Generator set to 'XORWOW Pseudo-random Generator', Seed =inforce.rng_seed.
- Distribution Algorithm:** Non-Central Chi Square, with a dropdown menu showing options: Mersenne Twister (607) Pseudo-random Generator, Sobol (32) Quasi-random Generator, Sobol (64) Quasi-random Generator, Minimal Random Generator, CURAND MTGP11213 Generator, and CURAND MRG32K3A Generator.
- Correlation Params:** Model library RNG combinations set to 'Default', Correlation set to 'Linear Correlation', Correlation Matrix =GBM_HW2F_correlation.cor_matrix.
- Correlation Matrix Preview:** A table showing the correlation matrix for 'Equity_ESG' and 'Interest_Rate_ESG'.

	corr_whitenoise[0] (Normal)	corr_whitenoise[1] (Normal)	corr_whitenoise[2] (Normal)	corr_whitenoise[3] (Normal)	corr_whitenoise[4] (Normal)	corr_wn[0] (Normal)	corr_wn[1] (Normal)
EquityESG	1 0.822 0.822 0.681 0.971 0.733	0.822 1 0.686 0.806 0.66 0	0.681 0.686 1 0.651 0.904 0	0.971 0.806 0.651 1 0.68 0	0.733 0.66 0.904 0.68 1 0	0 0 0 0 0 1	0 0 0 0 0 1
InterestRate_ESG	corr_wn[0] (Normal) corr_wn[1] (Normal)	0 0	0 0	0 0	0 0	1 0	0 1

PathWise Modeling Studio

- Import and configure Model Libraries (e.g. pre-built Economic Scenario Generators)

The screenshot shows the PathWise Modeling Studio application window. The top menu bar includes Home, Formulas, View, Revisions Control, Inforce, and Shocks. Below the menu is a toolbar with various icons for operations like Cut, Paste, Add sheet, Remove sheet, Rearrange Sheets, Add column, Edit column, Add Result, Remove Result, Add Local Run, Remove local run, Execute Local Run, Deploy, Sample Script, Find, and Deployment.

The left side features a Project Browser pane displaying a hierarchical tree structure of the model. Nodes include Model - GMMB, Data sources, Shocks, RNG Configuration, Model Libraries, Equity_ESG, Geometric_Brownian_Motion, Inputs, Calculations, simu_params, scenarios, Output, User-defined functions, Interest_Rate_ESG, Volatility_ESG, Calculations, Calculation Results, Greeks, Local Runs, and User-defined functions.

The main workspace contains a scenario configuration window titled "Geometric_Brownian_Motion.scenarios" and an RNG Configuration window. The RNG Configuration window lists several distribution types: chisquare, ncchisquare, normal, poisson, and uniform. The main workspace displays a data grid with columns labeled "dt_in_years" and "corr_whitenoise[0]". The data grid contains numerous rows of numerical values, such as:

	dt_in_years	corr_whitenoise[0]
1	0.06301	0
2	0.08219	-0.1548973
3	0.08219	0.04265733
4	0.08219	-1.006912
5	0.08219	-1.465682
6	0.08219	-0.2940714
7	0.08219	-0.8779107
8	0.08219	0.3820293
9	0.08219	-0.423768
10	0.08219	0.2174918
11	0.08219	-0.4242617
12	0.08219	0.4300877
13	0.08219	-1.285436
14	0.08219	0.2424345
15	0.08219	-5.8794 %
16	0.08219	-12.7533 %
17	0.08219	2.3601 %
18	0.08219	-7.4764 %
19	0.08219	0.0000 %
20	0.08219	0.0000 %
21	0.08219	0.0000 %
22	0.08219	0.0000 %
23	0.08219	0.0000 %
24	0.08219	0.0000 %
25	0.08219	0.0000 %

The bottom status bar indicates "Ready" and "Precision: Single".

PathWise Modeling Studio

- Calculate number of time-steps to simulate

The screenshot shows the PathWise Modeling Studio interface. The main window displays a data grid titled "time_step_logic.num_steps". The grid contains columns for "prev_projection_date", "current_projection_date", "next_projection_date", "prev_projection_date_id", "current_projection_date_id", "projection_time_in_days", and "projection_time_in_years". A specific row is highlighted with a red circle, and a red arrow points from the text "Time" to the "projection_time_in_days" column. The "Project Browser" on the left shows the model structure, including "Model - GMMB", "Data sources", "RNG Configuration", "Model Libraries", and "Calculations". The "cashflow" calculation is selected.

	prev_projection_date	current_projection_date	next_projection_date	prev_projection_date_id	current_projection_date_id	projection_time_in_days	projection_time_in_years
1	2012-01-01	2012-01-01	2012-01-24	2455928	2455928	0	0.00000
2	2012-01-01	2012-01-24	2012-02-23	2455928	2455951	23	0.06301
3	2012-01-24	2012-02-23	2012-03-24	2455951	2455981	53	0.14521
4	2012-02-23	2012-03-24	2012-04-23	2455981	2456011	83	0.22740
5	2012-03-24	2012-04-23	2012-05-23	2456011	2456041	113	0.30959
6	2012-04-23	2012-05-23	2012-06-22	2456041	2456071	143	0.39178
7	2012-05-23	2012-06-22	2012-07-22	2456071	2456101	173	0.47397
8	2012-06-22	2012-07-22	2012-08-21	2456101	2456131	203	0.55616
9	2012-07-22	2012-08-21	2012-09-20	2456131	2456161	233	0.63836
10	2012-08-21	2012-09-20	2012-10-20	2456161	2456191	263	0.72055
11	2012-09-20	2012-10-20	2012-11-19	2456191	2456221	293	0.80274
12	2012-10-20	2012-11-19	2012-12-19	2456221	2456251	323	0.88493
13	2012-11-19	2012-12-19	2013-01-18	2456251	2456281	353	0.96712
14	2012-12-19	2013-01-18	2013-02-17	2456281	2456311	383	1.04932
15	2013-01-18	2013-02-17	2013-03-19	2456311	2456341	413	1.13151
16	2013-02-17	2013-03-19	2013-04-18	2456341	2456371	443	1.21370
17	2013-03-19	2013-04-18	2013-05-18	2456371	2456401	473	1.29589
18	2013-04-18	2013-05-18	2013-06-17	2456401	2456431	503	1.37808
19	2013-05-18	2013-06-17	2013-07-17	2456431	2456461	533	1.46027
20	2013-06-17	2013-07-17	2013-08-16	2456461	2456491	563	1.54247
21	2013-07-17	2013-08-16	2013-09-15	2456491	2456521	593	1.62466
22	2013-08-16	2013-09-15	2013-10-15	2456521	2456551	623	1.70685
23	2013-09-15	2013-10-15	2013-11-14	2456551	2456581	653	1.78904
24	2013-10-15	2013-11-14	2013-12-14	2456581	2456611	683	1.87123
25	2013-11-14	2013-12-14	2014-01-13	2456611	2456641	713	1.95343
26	2013-12-14	2014-01-13	2014-02-12	2456641	2456671	743	2.03562

PathWise Modeling Studio

■ Define simulation columns and formulas

The screenshot shows the PathWise Modeling Studio interface. The top menu bar includes Home, Formulas, View, Revisions Control, Inforce, and Shocks. The main window has tabs for cashflow, Geometric_Brownian_Motion.scenarios, and RNG Configuration. The cashflow tab is active, displaying a code editor with the following pseudocode:

```
guaranteed_value real
if is_ratchet_step then
    gv = max(total_account_value, prev(guaranteed_value))
else
    gv = prev(guaranteed_value)
endif
return gv
```

Below the code editor is a data grid with columns: _return[2], account_value[2], log_r, _return[4], account_value[4], total_account_value, guaranteed_value, itm_pct, and base_la. The data grid contains numerous rows of financial data. A context menu is open over the data grid, showing options like \$root, abs, account_value, acos, age, asin, atan, avg, avg_range, base_lapse_lt, base_Qt, business_rules, ceil, check_date, check_julian, cholesky, claims, compare, etc.

PathWise Modeling Studio

- Encapsulate re-usable logic in UDFs and UDF libraries

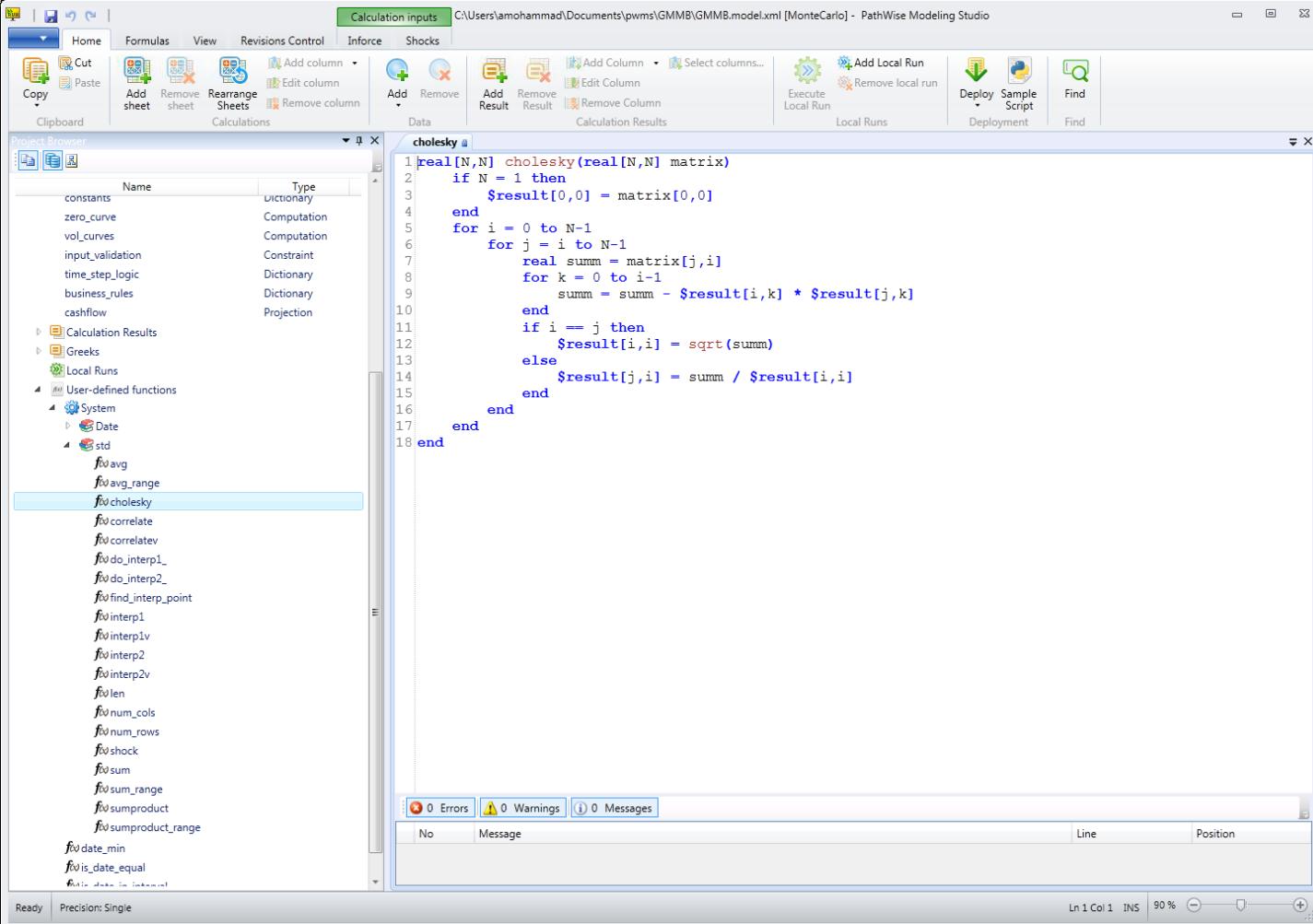
The screenshot shows the PathWise Modeling Studio interface. The main window displays a code editor with C++ code for the Cholesky decomposition algorithm. The code is as follows:

```
real[N,N] cholesky(real[N,N] matrix)
{
    if N == 1 then
        $result[0,0] = matrix[0,0]
    end
    for i = 0 to N-1
        for j = i to N-1
            real summ = matrix[j,i]
            for k = 0 to i-1
                summ = summ - $result[i,k] * $result[j,k]
            end
            if i == j then
                $result[i,i] = sqrt(summ)
            else
                $result[j,i] = summ / $result[i,i]
            end
        end
    end
}
```

The code editor has syntax highlighting for C++ and includes status bars for errors, warnings, and messages. The project browser on the left shows various files and functions, with `fvocholesky` selected. The toolbar at the top provides various modeling and deployment tools.

PathWise Modeling Studio

- Encapsulate re-usable logic in UDFs and UDF libraries



The screenshot shows the PathWise Modeling Studio interface. The main window displays a code editor with C++ code for performing Cholesky decomposition. The code is as follows:

```
real[N,N] cholesky(real[N,N] matrix)
{
    if N == 1 then
        $result[0,0] = matrix[0,0]
    end
    for i = 0 to N-1
        for j = i to N-1
            real summ = matrix[j,i]
            for k = 0 to i-1
                summ = summ - $result[i,k] * $result[j,k]
            end
            if i == j then
                $result[i,i] = sqrt(summ)
            else
                $result[j,i] = summ / $result[i,i]
            end
        end
    end
}
```

The code editor has syntax highlighting for C++ and includes status bars at the bottom showing '0 Errors', '0 Warnings', and '0 Messages'. The project browser on the left lists various components like 'User-defined functions' and 'System'.

PathWise Modeling Studio

- Define model outputs (e.g. Greeks)

The screenshot shows the PathWise Modeling Studio interface. The left sidebar displays the project structure under 'Model - GMMB'. The 'Greeks' section is selected, showing a table with columns 'Name' and 'Type'. One row is highlighted, showing 'shocks' as a 'Greeks' type calculation. The main workspace shows a hierarchical tree for 'delta_gamma.shocks' under 'cholesky a'. The tree has 11 nodes, each labeled 'Array [5]' with sub-nodes 'add' and 'mul'. The first node's 'add' sub-node contains values: 0 0 -0.01; 1 0 0; 2 0 0; 3 0 0; 4 0 0. The second node's 'add' sub-node contains values: 0 0 0.01; 1 0 0; 2 0 0; 3 0 0; 4 0 0. The third node's 'add' sub-node contains values: 0 0 0; 1 0 -0.01; 2 0 0; 3 0 0; 4 0 0. The fourth node's 'add' sub-node contains values: 0 0 0; 1 0 0.01; 2 0 0; 3 0 0; 4 0 0. The fifth node's 'add' sub-node contains values: 0 0 0; 1 0 0; 2 0 0; 3 0 0; 4 0 0. The sixth node's 'add' sub-node contains values: 0 0 0; 1 0 0; 2 0 0; 3 0 0; 4 0 0. The seventh node's 'add' sub-node contains values: 0 0 0; 1 0 0; 2 0 0; 3 0 0; 4 0 0. The eighth node's 'add' sub-node contains values: 0 0 0; 1 0 0; 2 0 0; 3 0 0; 4 0 0. The ninth node's 'add' sub-node contains values: 0 0 0; 1 0 0; 2 0 0; 3 0 0; 4 0 0. The tenth node's 'add' sub-node contains values: 0 0 0; 1 0 0; 2 0 0; 3 0 0; 4 0 0.

PathWise Modeling Studio

- Define model outputs (e.g. Greeks)

The screenshot shows the PathWise Modeling Studio interface. The top menu bar includes Home, Formulas, View, Revisions Control, Inforce, and Shocks. The main area has tabs for Calculation inputs, Project Browser, and a detailed view of a calculation named 'delta_gamma.Calculation'.

Project Browser:

- Model - GMMB
 - Data sources
 - Shocks
 - RNG Configuration
 - Model Libraries
 - Calculations
 - Calculation Results
 - Greeks
 - delta_gamma
 - Greeks

delta_gamma.Calculation:

```

delta real[5] > idx_dn = $index*2 + 1
          idx_up = $index*2 + 2
          fmv_up = fmv[idx_up].avg_fmv
          fmv_dn = fmv[idx_dn].avg_fmv
          av_up = shock( inforce.account_value[$index], shocks[idx_up].inforce_account_value[$index] )
          av_dn = shock( inforce.account_value[$index], shocks[idx_dn].inforce_account_value[$index] )
          return (fmv_up - fmv_dn) / (av_up - av_dn)
    
```

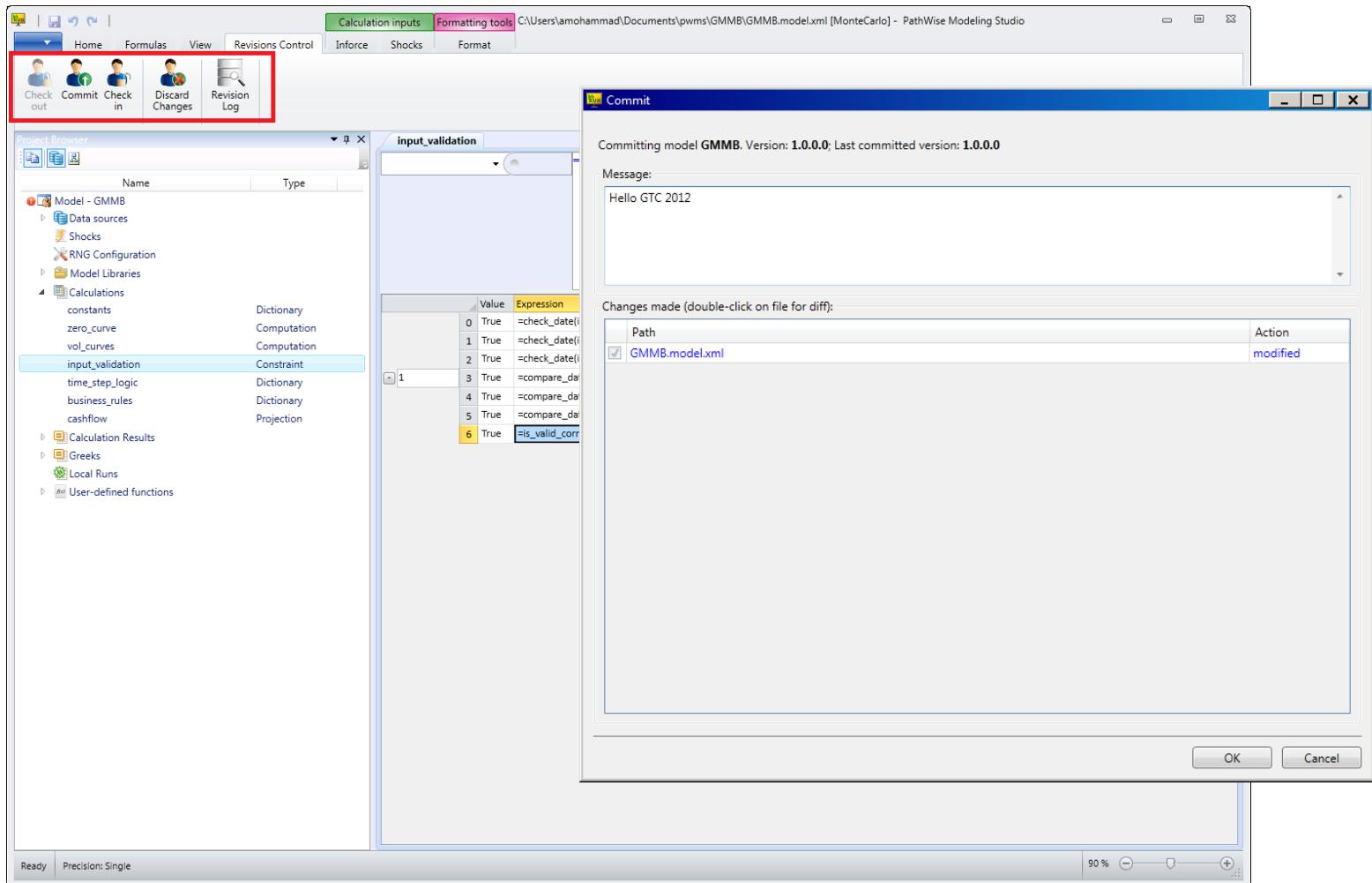
Results Table:

	formula	calculated value
base_fmv	=fmv[0].avg_fmv	-72.0284
delta[0]	$\begin{aligned} \text{idx_dn} &= \$\text{index}*2 + 1 \\ \text{idx_up} &= \$\text{index}*2 + 2 \\ \text{fmv_up} &= \text{fmv}[\text{idx_up}].\text{avg_fmv} \\ \text{fmv_dn} &= \text{fmv}[\text{idx_dn}].\text{avg_fmv} \\ \text{av_up} &= \text{shock}(\text{inforce.account_value}[\$index], \text{shocks}[\text{idx_up}].\text{inforce_account_value}[\$index]) \\ \text{av_dn} &= \text{shock}(\text{inforce.account_value}[\$index], \text{shocks}[\text{idx_dn}].\text{inforce_account_value}[\$index]) \\ \text{return} &= (\text{fmv_up} - \text{fmv_dn}) / (\text{av_up} - \text{av_dn}) \end{aligned}$	0.00425906
dollar_delta[0]	$\begin{aligned} \text{idx_dn} &= \$\text{index}*2 + 1 \\ \text{idx_up} &= \$\text{index}*2 + 2 \\ \text{fmv_up} &= \text{fmv}[\text{idx_up}].\text{avg_fmv} \\ \text{fmv_dn} &= \text{fmv}[\text{idx_dn}].\text{avg_fmv} \\ \text{av_up} &= \text{shock}(\text{inforce.account_value}[\$index], \text{shocks}[\text{idx_up}].\text{inforce_account_value}[\$index]) \\ \text{av_dn} &= \text{shock}(\text{inforce.account_value}[\$index], \text{shocks}[\text{idx_dn}].\text{inforce_account_value}[\$index]) \\ \text{av_base} &= \text{inforce.account_value}[\$index] \\ \text{return} &= (\text{fmv_up} - \text{fmv_dn}) / (\text{av_up} - \text{av_dn}) * \text{av_base} \end{aligned}$	425.906
gamma[0]	$\begin{aligned} \text{idx_dn} &= \$\text{index}*2 + 1 \\ \text{idx_up} &= \$\text{index}*2 + 2 \\ \text{fmv_up} &= \text{fmv}[\text{idx_up}].\text{avg_fmv} \\ \text{fmv_dn} &= \text{fmv}[\text{idx_dn}].\text{avg_fmv} \\ \text{fmv_base} &= \text{fmv}[0].\text{avg_fmv} \\ \text{av_up} &= \text{shock}(\text{inforce.account_value}[\$index], \text{shocks}[\text{idx_up}].\text{inforce_account_value}[\$index]) \\ \text{av_dn} &= \text{shock}(\text{inforce.account_value}[\$index], \text{shocks}[\text{idx_dn}].\text{inforce_account_value}[\$index]) \\ \text{return} &= (\text{fmv_up} - 2*\text{fmv_base} + \text{fmv_dn}) / ((\text{av_up} - \text{av_dn})^2) * \text{av_base}^2 \end{aligned}$	-2.40879E-08
dollar_gamma[0]	$\begin{aligned} \text{idx_dn} &= \$\text{index}*2 + 1 \\ \text{idx_up} &= \$\text{index}*2 + 2 \\ \text{fmv_up} &= \text{fmv}[\text{idx_up}].\text{avg_fmv} \\ \text{fmv_dn} &= \text{fmv}[\text{idx_dn}].\text{avg_fmv} \\ \text{fmv_base} &= \text{fmv}[0].\text{avg_fmv} \\ \text{av_up} &= \text{shock}(\text{inforce.account_value}[\$index], \text{shocks}[\text{idx_up}].\text{inforce_account_value}[\$index]) \\ \text{av_dn} &= \text{shock}(\text{inforce.account_value}[\$index], \text{shocks}[\text{idx_dn}].\text{inforce_account_value}[\$index]) \\ \text{av_base} &= \text{inforce.account_value}[\$index] \\ \text{return} &= (\text{fmv_up} - 2*\text{fmv_base} + \text{fmv_dn}) / ((\text{av_up} - \text{av_dn})^2) * \text{av_base}^2 \end{aligned}$	-240.879

Bottom status bar: Ready | Precision: Single | 90% | zoom controls

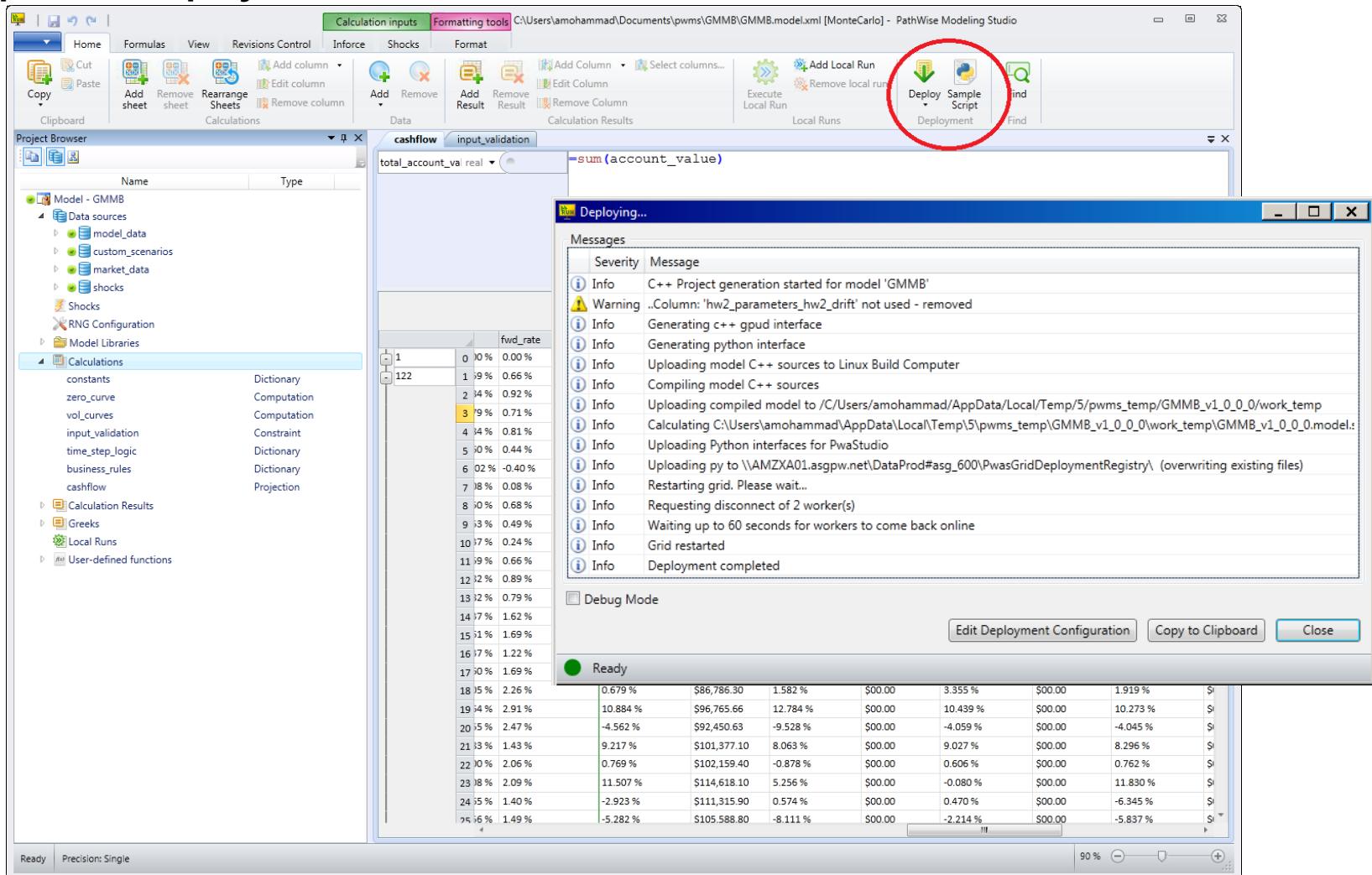
PathWise Modeling Studio

- Commit model to SVN source code repo



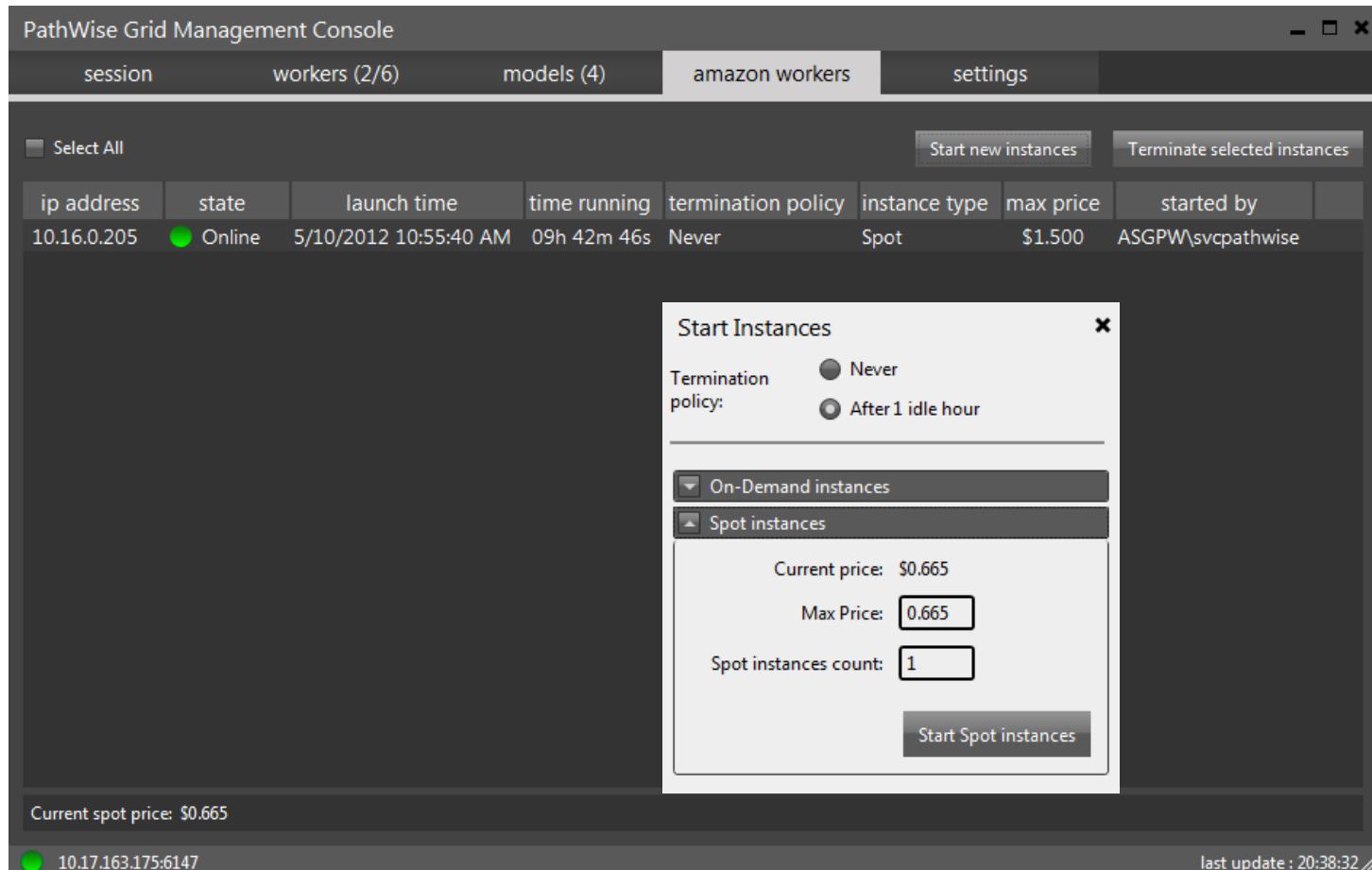
PathWise Modeling Studio

▪ Compile and deploy model to GPUs



PathWise Modeling Studio

- Add GPU grid workers from the Cloud



PathWise Modeling Studio

- Generate sample Python script

The screenshot shows a window titled "Sample Script". On the left, there is a sidebar with a tree view of available options:

- Options
 - Precision
 - Single (selected)
 - Double
 - Begin session
 - Default_begin_session (selected)
 - Default_begin_session_with_shocks
 - Computation
 - compute_aging
 - compute_delta_gamma
 - compute_esg_verbose
 - compute_greeks
 - compute_option_value (selected)
 - compute_output_by_path
 - compute_policy_verbose
 - compute_test_verbose
 - For Version
 - Latest (selected)
 - Current

The main area is titled "Script" and contains the following Python code:

```
import global_config as cfg
import etl
import numpy as np

# Import model
import GMMB_model_t_float as model

# Load inputs
DATA_DIR = 'C:\\\\Users\\\\amohammad\\\\Documents\\\\pwms\\\\GMMB\\\\'

model_data_npz = etl.load(DATA_DIR + 'data/model_data.npz', model.PRECISION)
market_data_npz = etl.load(DATA_DIR + 'data/market_data.npz', model.PRECISION)
inforce = model_data_npz['inforce']

# Compute
session = model.GMMB_session_t(cfg.COORDINATOR_ENDPOINT)
session.begin_session(model_data_npz, market_data_npz, modelType = model.ModelType.Default)
fmvs = session.compute_option_value(inforce)
session.end_session()

# Results
etl.savez('output.npz', fmvs = fmvs)
print 'DONE'
```

At the bottom of the window are three buttons: "Copy to Clipboard", "Save As ...", and "Close".

PathWise Analytics Studio

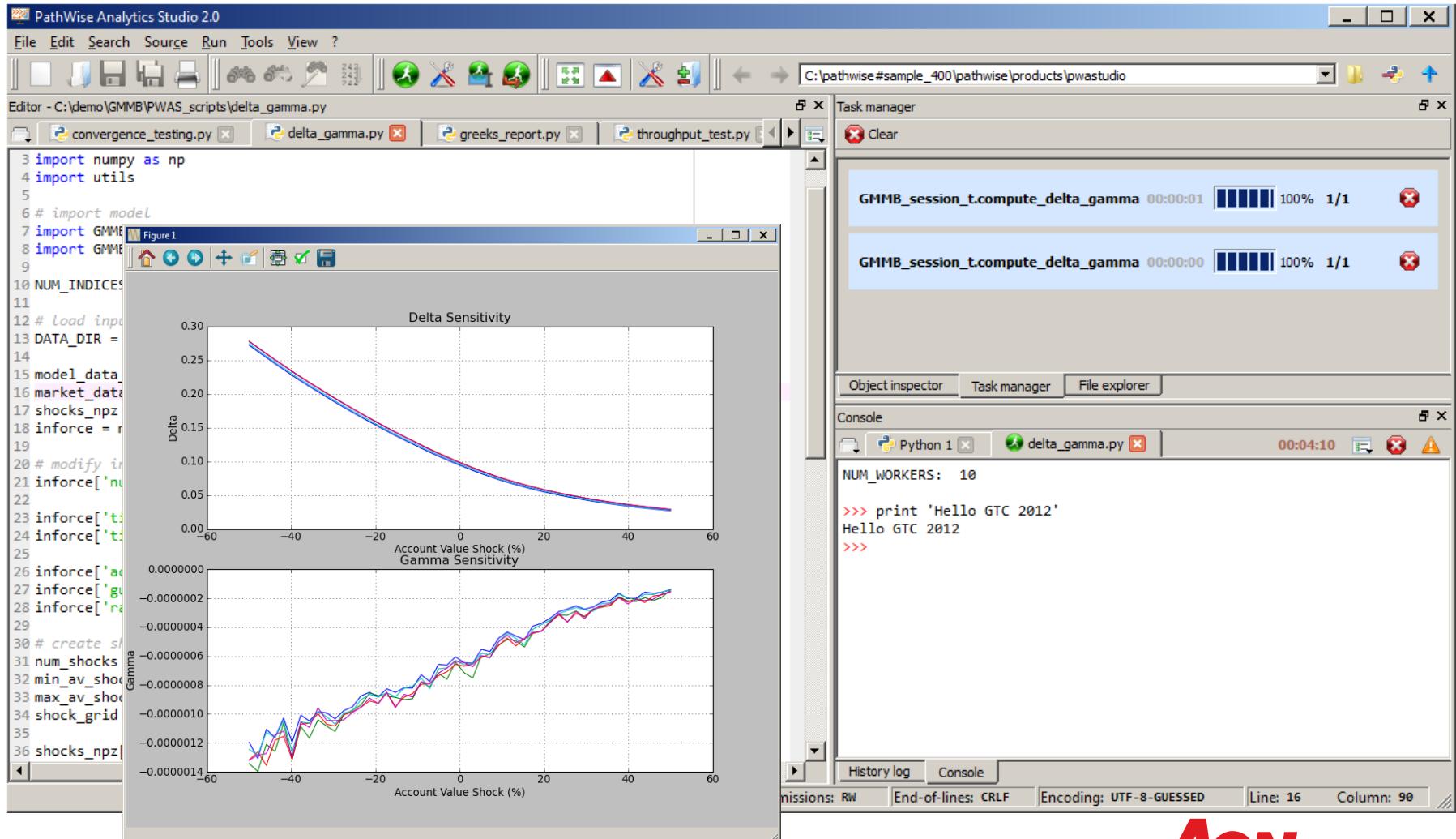
- Run Python scripts from PathWise Analytics Studio (customized Python IDE)

The screenshot shows the PathWise Analytics Studio interface. On the left is a code editor window titled "Editor - C:\Users\joel\Documents\PathWise_Training\scripts\Demo3.py" containing Python script code. The script performs various financial calculations, including loading data, creating shock arrays, computing Greeks, and plotting P&L attribution. On the right side, there is a "Task manager" window showing two tasks: "Compute Greeks" and "Compute FMV", both at 100% completion. Below the Task manager is a "Console" window with the title "Figure1" and a 3D bar chart titled "P&L Attribution, 2nd Order Greeks". The chart displays P&L values across different index shocks (SPX, TSX, DEX, EAFE) and market movements (MM). The Z-axis represents P&L values ranging from -30,000 to 10,000. The X and Y axes represent the index shocks and market movements.

```
1 ## Copyright (C) Aon Benfield Securities 2012, All Rights Reserved
2 from settings import *
3
4 # Load data (inforce, assumptions, market)
5 mapped_inforce_npz = etl.load(DATA_DIR+'signed_mapped_inforce-20120222.npz',precision)
6 inforce = scale_inforce(mapped_inforce_npz, 20000, 41632 )
7 assumptions_npz = demo.get_best_assumptions(mapped_inforce_npz)
8 market_data = mapped_inforce_npz['market_data']
9
10 # Create shock array
11 n = 5
12 multiplier_shock = np.array([ 0.01, 0.01, 0.01, 0.01, 0.01 ])
13 num_shocks = 2 * n * n + 1
14 shocks = np.zeros( num_shocks, dtype=demo_model.shock_data_t )
15 shocks['av']['mul'] = shk.createShockArray( multiplier_shock )
16
17 # Compute Delta & Gamma
18 fmv = computeFMV( assumptions_npz, market_data, shocks, inforce, 'Compute Greeks' )
19 (jacobian, hessian) = shk.taylor_expansion( multiplier_shock, fmv )
20
21 # Compute FMV under market movement
22 newshocks = np.zeros( 2, dtype = demo.model.shock_data_t )
23 dx = np.array([[ 0.007, 0.005, -0.005, 0.01, 0.008 ]])
24 newshocks['av']['mul'][1,:] = dx
25 marketfmv = computeFMV( assumptions_npz, market_data, newshocks, inforce, 'Compute FMV' )
26
27 # Compare actual PnL versus approximated PnL
28 actual_pnl = marketfmv[1] - marketfmv[0]
29 full_2ndorder_pnl = (np.dot(jacobian,np.transpose(dx)) + \
30 0.5 * np.dot(dx,np.dot(hessian,np.transpose(dx))))[0][0]
31 part_2ndorder_pnl = (np.dot(jacobian,np.transpose(dx)) + \
32 0.5 * np.dot( dx*dx, np.diag(hessian) ))[0][0]
33
34 # Plot PnL
35 data = 0.5 * ( np.transpose(dx) * hessian ) * dx
36 createplot( data )
```

PathWise Analytics Studio

- Run Python scripts from PathWise Analytics Studio (customized Python IDE)



PathWise Seriatim Real-Time Risk System

AON BENFIELD PathWise CALCULATION: 67% ONLINE |

DATA Delta Rho FX Trade Blotters Market Data Updates

Liability Delta 12:24:22

Mngr Group	Risk Factor	Dollar Delta	Currency
36	TSE	114,846,800.00	CAD
35	S&P 500	17,544,200.00	CAD
34	EAFE	11,752,600.00	CAD
37	DEX	66,200,000.00	CAD

Asset Equity Delta 12:24:22

Mngr Group	Risk Factor	Dollar Delta	Currency
36	TSE	-122,840,060.00	CAD
35	S&P500	-17,574,724.44	CAD
34	EAFE	-16,111,707.07	CAD
37	DEX	-66,094,560.00	CAD

Net Delta 12:24:22

Mngr Group	Risk Factor	Liability Dollar Delta	Asset Dollar Delta	Net Dollar Delta	Currency
36	TSE	114,846,800.00	-122,840,060.00	-7,993,260.00	CAD
35	S&P500	17,544,200.00	-17,574,724.44	-30,524.44	CAD
34	EAFE	11,752,600.00	-16,111,707.07	-4,359,107.07	CAD
37	DEX	66,200,000.00	-66,094,560.00	105,440.00	CAD

Liability Gamma 12:24:22

Mngr Group	Risk Factor	Dollar Gamma	Currency
36	TSE	-185,920,000.00	CAD
35	S&P 500	360,000.00	CAD
34	EAFE	1,320,000.00	CAD
37	DEX	-141,200,000.00	CAD

Liability Gamma srt_valuation_result/liability_gamma histor...

Liability Delta for column=Dollar Delta 12:24:22

	Liability Delta
TSE	114,846,800.00
S&P 500	17,544,200.00
EAFE	11,752,600.00
DEX	66,200,000.00

Asset Equity Delta history for Mngr Group=35 and column=Dollar Delta 12:24:22

2/15/2011 12:11:15 PM • Dollar Delta -17,557,680.65

Net Delta history for Mngr Group=36 and column=Net Dollar Delta 12:24:22

2/15/2011 8:04:54 AM • Net Dollar Delta 1,207,360.00

Tier 1 Delta Risk Limits 12:24:22

Mngr Group	Risk Factor	Net Dollar Delta	Lower Limit	Upper Limit	Currency	Limit Breach
36	TSE	-7,993,260.00	-10,723,249.36	10,723,249.36	CAD	False
35	S&P500	-30,524.44	-1,648,966.27	1,648,966.27	CAD	False
34	EAFE	-4,359,107.07	-1,083,626.63	1,083,626.63	CAD	True
37	DEX	105,440.00	-6,122,165.47	6,122,165.47	CAD	False

Tier 2 Delta Risk Limits 12:24:22

Mngr Group	Risk Factor	Net Dollar Delta	Lower Limit	Upper Limit	Currency	Limit Breach
36	TSE	-7,993,260.00	-16,084,873.42	16,084,873.42	CAD	False
35	S&P500	-30,524.44	-2,473,449.30	2,473,449.30	CAD	False
34	EAFE	-4,359,107.07	-1,625,439.89	1,625,439.89	CAD	True
37	DEX	105,440.00	-9,183,247.84	9,183,247.84	CAD	False

Tier 3 Delta Risk Limits 12:24:22

Mngr Group	Risk Factor	Net Dollar Delta	Lower Limit	Upper Limit	Currency	Limit Breach
36	TSE	-7,993,260.00	-21,446,498.72	21,446,498.72	CAD	False
35	S&P500	-30,524.44	-3,297,932.53	3,297,932.53	CAD	False
34	EAFE	-4,359,107.07	-2,167,253.27	2,167,253.27	CAD	True
37	DEX	105,440.00	-12,244,330.93	12,244,330.93	CAD	False

2/15/2011 8:26:33 AM SRT Loop Started

Thank You