

## NVIDIA CUDA 5.5 on an IBM System X iDataPlex dx360 M4 Server with 2 x NVIDIA K20Xm Co-Processors

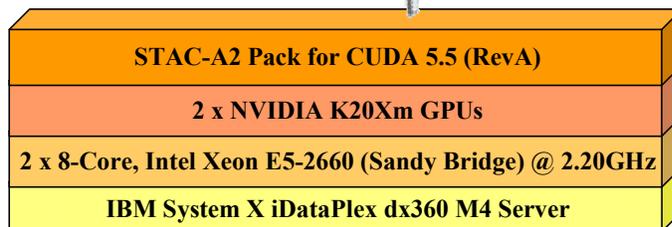
SUT ID: NVDA131118

### STAC-A2™ BENCHMARKS

Test date: November 18, 2013

Draft v1.01, December 3, 2013

Stack under test



Benchmark specs:

STAC-A2  
(beta 2)



THESE TESTS FOLLOWED STAC BENCHMARK SPECIFICATIONS PROPOSED OR APPROVED BY THE STAC BENCHMARK COUNCIL (SEE [WWW.STACRESEARCH.COM](http://WWW.STACRESEARCH.COM)). BE SURE TO CHECK THE VERSION OF ANY SPECIFICATION USED IN A REPORT. DIFFERENT VERSIONS MAY NOT YIELD RESULTS THAT CAN BE COMPARED TO ONE ANOTHER.

## Contents

REFERENCES.....	3
SUMMARY.....	5
1. OVERVIEW OF THE STAC-A2 BENCHMARK SPECIFICATIONS .....	7
1.1 <i>Business context</i> .....	7
1.2 <i>Benchmark construction</i> .....	8
1.3 <i>Interpreting and comparing STAC-A2 results</i> .....	11
2. THE STACK UNDER TEST .....	11
2.1 <i>Understanding this implementation of STAC-A2</i> .....	12
2.2 <i>Understanding the rest of the SUT</i> .....	12
3. PROJECT PARTICIPANTS AND RESPONSIBILITIES .....	13
4. CONTACTS.....	13
5. RESULTS STATUS .....	13
6. SPECIFICATIONS.....	14
7. LIMITATIONS AND CLARIFICATIONS .....	14
8. QUALITY OF THIS IMPLEMENTATION .....	14
8.1 <i>STAC-A2 quality benchmarks</i> .....	14
8.2 <i>Quality visualizations</i> .....	19
9. SPEED BENCHMARKS.....	24
10. SUT SCALING .....	26
11. CAPACITY BENCHMARKS .....	28
12. WORKLOAD SCALING .....	28
13. VENDOR COMMENTARY .....	31
ABOUT STAC .....	32

## References

- [1] Black, Fischer and Scholes, Myron: "The Pricing of Options and Corporate Liabilities", Journal of Political Economy, Vol 81, Issue 3 (May-Jun., 1973, 637-654.
- [2] Heston, Steven L. 1993: "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options", Review of Financial Studies, 6, 327–343.
- [3] Specifications used for this benchmark: STAC-A2 Benchmark Specifications, Rev P. Accessible by qualified members of the STAC Benchmark Council at <http://www.STACresearch.com/a2specs>.
- [4] Andersen, Leif B.G., Efficient Simulation of the Heston Stochastic Volatility Model (January 23, 2007). Available at SSRN : //ssrn.com/abstract=946405.
- [5] F. A. Longstaff and E. S. Schwartz, "Valuing American Options by Simulation: A Simple Least-Squares Approach," Review of Financial Studies, 14(1), 2001 pp. 113–147.
- [6] Jean-Michel Muller, "On the definition of  $ulp(x)$ ", Research Report No 2005-09, École Normale Supérieure de Lyon (February 2005). <http://www.ens-lyon.fr/LIP/Pub/Rapports/RR/RR2005/RR2005-09.pdf>
- [7] Anderson, T. W.; Darling, D. A., "Asymptotic theory of certain 'goodness-of-fit' criteria based on stochastic processes," Annals of Mathematical Statistics 23: 193–212 (1952).
- [8] Andrew Rukhin, et al, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", Special Publication 800-22, National Institute of Standards and Technology (April 2010). <http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf>
- [9] Gretton, A., K. Fukumizu, C.-H. Teo, L. Song, B. Schoelkopf and A. Smola, "A Kernel Statistical Test of Independence," MPI Technical Report 168 (2008).

## Disclaimer

This document was prepared by the Securities Technology Analysis Center (STAC®) at the request of NVIDIA. It is provided for your internal use only and may not be redistributed, retransmitted, or published in any form without the prior written consent of STAC. "STAC" and all STAC names are registered trademarks or trademarks of the Securities Technology Analysis Center LLC. All other trademarks in this document belong to their respective owners.

The test results contained in this report are made available for informational purposes only. Neither STAC nor the vendor(s) supplying the information in this report guarantee similar performance results. All information contained herein is provided on an "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND. STAC explicitly disclaims any liability whatsoever for any errors or otherwise.

## Summary

STAC recently performed STAC-A2 Benchmark tests on a stack consisting of the STAC-A2 Pack for CUDA 5.5 (Rev A) running on an IBM System X iDataPlex dx360 M4 Server with two NVIDIA K20Xm GPUs. This report describes the results of those tests.

NVIDIA authored the STAC Pack using NVIDIA CUDA 5.5.22 Toolkit, a suite for CUDA developers that includes the cuRAND and cuBLAS libraries and the NVIDIA nvcc Compiler. NVIDIA also made use of the CUB library of CUDA utilities and GNU OpenMP technology.

In all, the STAC-A2 specifications deliver nearly 200 test results related to both performance and quality. A few of these are summarized in the STAC Report Card:

*Table 1*

<b>STAC-A2™ (beta 2) Report Card</b> <b>NVIDIA CUDA 5.5 / 2 x NVIDIA K20Xm co-processors / IBM System X iDataPlex dx360 M4</b> <b>(SUT ID: NVDA131118)</b>			
STAC-A2.β2.GREEKS.TIME	Seconds to compute all Greeks with 5 assets, 25K paths, and 252 timesteps.	MEAN	0.770
	(For an explanation of MEAN, WARM, and COLD, see Section 9. Reports prior to Dec 2013 only reported MEAN.)	WARM	0.648
		COLD	1.258
STAC-A2.β2.GREEKS.MAX_ASSETS	Max assets completed in 10 minutes with 25K paths and 252 timesteps (using cold test runs).		41
STAC-A2.β2.GREEKS.MAX_PATHS	Max paths completed in 10 minutes with 5 assets and 252 timesteps (using cold test runs).		8,500,000
STAC-A2.β2.GREEKS.ENERGY_EFFICIENCY	Energy efficiency = GREEKS.MAX_ASSETS / Energy at Capacity		N/A <sup>1</sup>
STAC-A2.β2.PATHGEN.PVSTDERR	Worst case standard error in path generation across 5 assets		0.022
STAC-A2.β2.EARLYEX.ERR	Relative error in Longstaff-Schwartz valuation vs Black-Scholes binomial approximation		0.007

NVIDIA would also like to draw the reader's attention to the following:

*In the end-to-end Greeks benchmark (STAC-A2.β2.GREEKS.TIME), this system was:*

- *Over 9x the average speed of a system with the same class of CPUs but no GPUs (SUT INT130607a)*
- *Over 6x the average speed of the fastest publicly benchmarked system without GPUs (SUT INT130829)*
- *The first system to handle the baseline problem size in "real time" (less than one second)<sup>2</sup>*

<sup>1</sup> Efficiency tests are not required until 2014.

<sup>2</sup> By the response-time thresholds defined in the Workload Scaling analysis of Section 12.

This report describes the solution that was tested, salient aspects of the test project, the quality of the algorithms in the implementation, and the detailed scaling results. We recommend that readers who are not familiar with STAC-A2 first read Section 1 to get a feel for the test cases and metrics.

## Getting the most from these results

Any interested party can analyze public STAC Reports to compare the performance of different systems. However, members of the STAC Benchmark Council are able to put these reports to much greater use. Qualified members may:

- Read the detailed Configuration Disclosure for the system tested in this report
- Read the detailed test specifications
- Obtain the materials to run the STAC-A2 Benchmarks on their own systems
- Access additional reports in the confidential STAC Vault™
- Discuss benchmarks, technologies, and related business issues with their peers.

To join the Council or upgrade your membership, please contact [council@STACresearch.com](mailto:council@STACresearch.com).

## 1. Overview of the STAC-A2 Benchmark specifications

### 1.1 Business context

#### The importance of market risk management

Trading firms devote a tremendous amount of resource to analyzing their market risk—that is, how the value of the positions they hold in various financial instruments would change given potential combinations of price movements in the markets. Understanding market risk is critical to pricing and hedging deals, as well as deciding when to curtail or expand certain types of trading. Proper management of this risk is essential to the financial integrity of the world's trading institutions and the smooth functioning of capital markets. Over the last several years, well-publicized market events have increased the industry's focus on improving the quality of such risk management.

Many of the financial instruments that must be analyzed are derivatives. Understanding the market risk of derivative instruments typically requires a firm to analyze the sensitivity of the derivative values to changes in the behavior of the underlying instruments from which they are derived, as well as changes in the broader market. This analysis requires numerical methods that are computationally intensive. Large firms have datacenters packed with thousands of compute nodes dedicated to this task. The workload becomes more taxing as market volatility increases, portfolios get more complex, and more trading desks incorporate risk information into their decision making, requiring shorter turnaround times for analysis.

A firm's goals with respect to market risk analysis can depend on the extent to which it is focused on cost reduction, revenue expansion, or regulatory compliance. It may wish to reduce the time it takes to get answers or reduce the power and space required for calculations. It may wish to increase the accuracy of calculations or analyze more assets or serve more users. Or it may strive for all of these at once. No matter what the requirement, innovation in technology infrastructure has proven to be a crucial enabler of these business goals. The goal of the STAC-A2 Benchmark suite is to provide some common markers for technologies that firms may deploy in this mission.

#### The importance of options

STAC-A2 specifies computations related to options. Options are fundamental to today's financial markets. They are written on many underlying financial instruments, including equities, equity indices, commodities, bonds, currencies, interest rates, and even derivatives such as futures and swaps. In addition, certain other contracts like warrants and convertible debt implicitly incorporate options. Tens of billions of options contracts are traded every year.

With so much at stake, options traders and those who oversee them need good ways to figure out what options are worth. Options pricing was not well understood until the early 1970s, when Fisher Black and Myron Scholes produced their famous model.[1] Among other variables, the Black-Scholes formula says that the value of an option depends on both the price of the underlying asset and the volatility of that price (roughly, how much it varies over time). While Black-Scholes is still used today, many financial firms use more sophisticated models. One of these developed by Steven Heston in 1993, starts from different assumptions than Black-Scholes.[2] Black-Scholes assumes that while an asset's price is stochastic, the volatility of those prices is constant. By contrast, the Heston model assumes that the volatility is also stochastic. Thus, Heston is an example of a "stochastic volatility model".

Like many models in finance, the Heston model is a partial differential equation that cannot be solved exactly, except in special cases. Using it to price options requires mathematical approximation techniques. One of the most popular techniques is the Monte Carlo method, which essentially discovers a result by running a large number of randomized simulations. Each simulation is a potential path that the underlying asset's price and volatility could take over the lifetime of the option. That lifetime (typically months or years) is divided into timesteps (such as days). At each timestep, the path specifies a price and volatility for the asset, which are obtained from a probability distribution that assumes certain correlations among prices and volatilities across assets. The option's value is computed for each path, and the outcomes of all the paths are analyzed to determine the overall most likely price.

With the ability to compute option prices, risk managers are able to estimate how sensitive those prices are to changes in variables like the prices of the underlying assets, their volatility, and interest rates. These sensitivities,

each of which has been assigned a specific Greek letter by convention, are called "Greeks". Option Greeks are essential because they allow a risk manager to run further scenarios to determine how much value a firm's portfolio of options might lose or gain given various market shocks.

## The focus for STAC-A2

Because calculating Greeks is a very common, compute-bound task, the STAC Benchmark Council identified it as a workload that deserved a set of STAC Benchmark standards. STAC Benchmarks define repeatable workloads and meaningful metrics in a business-oriented, technology agnostic way. These allow vendors and end-user firms to generate apples-to-apples comparisons across any number of technology stacks of any type. This is important in light of the ongoing innovation in processors, memory, interconnects, analytic libraries, programming tools, and cluster/grid software that have potential to improve the processing of such workloads. In a rapidly changing market, standard benchmark results help end users decide where to direct their scarce R&D resources.

## 1.2 Benchmark construction

### STAC-A2 Operations

The core of STAC-A2 is a set of algorithms to compute Greeks on a particular type of option using particular mathematical methods to approximate particular theoretical values. These algorithms are expressed in the freely available R programming language.[3] STAC-A2 Implementations can be in any language.

STAC-A2 concerns a best-of option. The payoff gives the holder the right to decide, at any time prior to expiry, to pay a price fixed in advance for the best of N assets. This payoff is multi-asset and path dependent. It is an American option, meaning it can be exercised at any time prior to expiration (i.e., it allows "early exercise"). Such an option represents a class of complex payoff with non-trivial parallelization requirements.

Each firm—or trading desk within a firm—may be concerned with different types of options or use different algorithms to price them, so the degree to which it can draw conclusions solely from end-to-end benchmarks of a given algorithm will vary. For this reason, STAC-A2 also explicitly tests some of the key components of the overall calculation such as random number generation and special math functions. These component algorithms are commonly used in many financial models, including different algorithms for computing Greeks as well as algorithms that have nothing to do with Greeks.

STAC-A2 calls each algorithm an Operation (a more general benchmarking term that could in the future encompass things like memory transfer times, etc.). The current STAC-A2 Operations (with code names in parentheses) are listed below.

Component benchmarks:

- Square roots (SQRT)
- Exponentials (EXP)
- Logs (LOG)
- Unit-normal randoms (UNR)
- Correlated randoms (CORRAND)
- Path generation (PATHGEN)
- Payoff computation with early exercise (EARLYEX)

Greeks:

- Theta (THETA)
- Rho (RHO)
- Delta (DELTA)
- Gamma (GAMMA)
- Cross-gamma (CROSSGAMMA)

- Model vega (MODELVEGA)<sup>3</sup>
- Correlation vega (CORRVEGA)
- All Greeks (GREEKS)

Important things to note:

- The random number generator for UNR and CORRAND must be the same but may be any generator the implementer wishes. A poor-quality generator will suffer in the quality benchmarks.
- PATHGEN simulates the Heston stochastic volatility model [2] using the approach of Anderson.[4]
- EARLYEX uses the approach of Longstaff & Schwartz.[5]

See [3] for complete details.

## Test framework

At a high level, generating STAC-A2 Benchmarks requires four things:

1. A STAC-A2 Implementation. This is the logic that carries out the algorithms and measurements specified by STAC-A2. It is described in Section 2.1.
2. Something on which to run the STAC-A2 Implementation (whatever hardware and software is required). It is described in Section 2.2.
3. The STAC-A2 Run Scripts. These scripts direct the Implementation through all of the performance and quality test sequences required by the STAC-A2 Benchmark specifications, telling it at each step what algorithm to run, what problem size to use, etc.
4. An x86-based server on which to run the STAC-A2 Run Scripts.

#1 and #2, taken together, form the SUT. STAC-A2 places no constraints on the architecture of the SUT, except that it must be capable of being controlled by the STAC-A2 Run Scripts and using disk for input and output data (note that all reads and writes are not included in the benchmark times). The server in #4 may or may not be part of the SUT. STAC-A2 doesn't care.

Figure 1 illustrates the test framework. Conceptually, the Implementation has a control loop that initiates the STAC-A2 Operations as instructed by the STAC-A2 Run Scripts. The execution of Operations may be distributed across cores, cards, servers, or whatever topology is relevant to the Implementation. The control loop obtains timestamps before and after the Operation ( $t_{\text{submit}}$  and  $t_{\text{complete}}$ , respectively) and persists the difference—called "Elapsed Time"—in a performance-results file.

Before the tests, the tester generates input data using the STAC-A2 input data-generation scripts. When executing a given Operation, the Implementation inputs the appropriate file. This input phase may precede  $t_{\text{submit}}$ . The Implementation also persists the output of each Operation (i.e., algorithm results) to an output file. This persistence may take place after  $t_{\text{complete}}$ . The Implementation can also be configured to output certain data used to visualize the quality of its algorithms.

Following the execution of the Operations, the STAC-A2 Quality Assessment Tool produces STAC-A2 quality benchmarks based on the contents of the input and output files. The STAC-A2 analysis and reporting tools then take the performance and quality results to generate STAC Reports.

---

<sup>3</sup> In some of the quality tests, the output of the MODELVEGA Operation is analyzed for each of four Greeks: kappa (KAPPA), the model vega theta (MV\_THETA), khi (KHI), and V0 (V0).

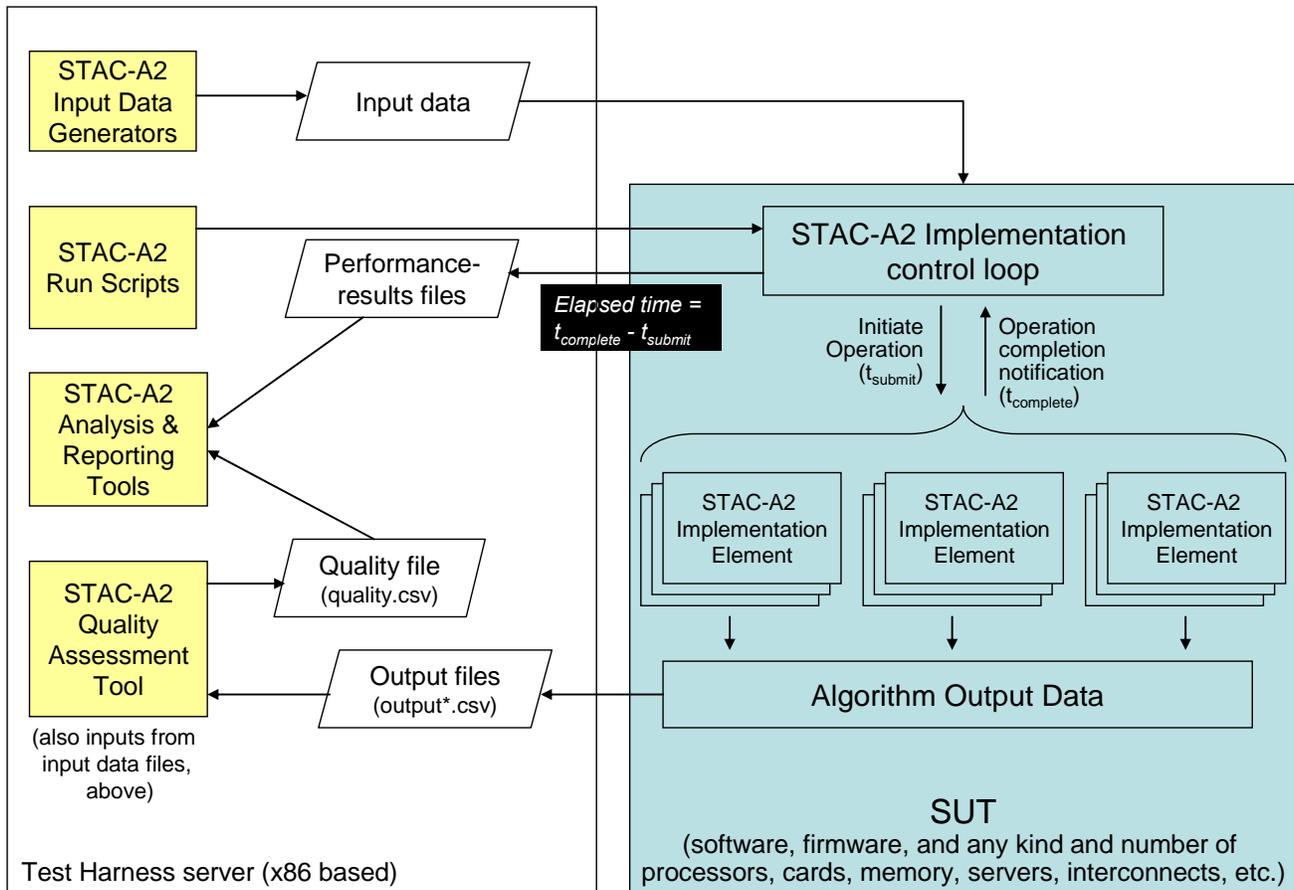


Figure 1 – Conceptual diagram

## STAC-A2 speed, capacity, and scaling tests

The baseline benchmarks in STAC-A2 measure the time to complete a fixed problem size. This enables consistent speed comparisons across multiple technology stacks. These "time per workload" benchmarks are meaningful to a user who cares about response times.

STAC-A2 also includes "workload per time" benchmarks—i.e., where the question is how much work the SUT can complete in a fixed unit of time. These benchmarks are important to capacity planners who need to know the capabilities of their system building blocks. STAC-A2 defines two capacity benchmarks: the maximum assets that can be handled while holding the rest of the problem size at its baseline values, and the maximum paths that can be processed while holding the rest of the problem size at baseline.

STAC-A2 also measures how well the "stack under test" (SUT) scales, in two ways:

1. Workload scaling, in which the SUT must try to handle problem sizes that increase along multiple dimensions.
2. SUT scaling, in which successively larger subsets of the SUT are tested against the same workload in order to demonstrate the incremental benefit of additional SUT resources.

Because each SUT will generally be tested at different scales from one another, scale tests are not considered benchmarks. But they are very useful data points.

Each of these performance tests is described further in Section 9, along with its results.

## STAC-A2 quality tests

A key influence on the speed of analytics is the quality that is required (whether the relevant dimension of quality is precision, closeness of observed to theoretical values, or other characteristics.). All other things being equal, higher quality generally requires more processing time. Clear measurement of algorithm quality was therefore a first-order design consideration for STAC-A2. In fact, STAC-A2 specifies more quality benchmarks than performance benchmarks.

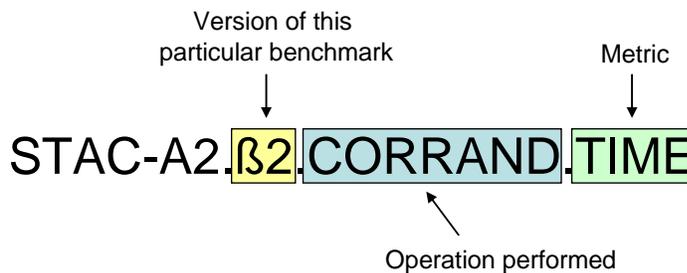
As discussed in Section 8, STAC-A2 does not impose minimum quality standards on an implementation but rather measures and reports its quality. This reflects the fact that an end user may have legitimate reasons to sacrifice quality for performance, depending on the use case. The STAC-A2 Working Group did not feel it was in a position to dictate a point on those tradeoff curves.

## STAC-A2 efficiency tests

STAC-A2 defines ways to measure the efficiency of the SUT with respect to power and space. However, these benchmarks are not required until 2014.

## 1.3 Interpreting and comparing STAC-A2 results

STAC-A2 is a suite of benchmarks. The tables and charts in this report identify each benchmark unambiguously, as follows:



In charts, the ID is sometimes decomposed, with part of it in the chart title or labels. Each individual STAC Benchmark™ specification has its own version number. The same version of a given spec may appear in multiple benchmark suites. Thus, the code names of the suites are irrelevant when making comparisons. Versioning individual specs enables the reader to compare a discrete result from this SUT to the corresponding result from another SUT. When making comparisons, be sure that the identifiers match exactly. If they do not, the benchmark results cannot be fairly compared.

## 2. The stack under test

As explained in Section 1.2, any SUT to be assessed by STAC-A2 Benchmarks has two layers:

1. An implementation of the benchmarks. This is software and/or firmware specifically written to implement the quantitative algorithms and other logic required by the STAC-A2 Benchmark specifications. For example, this could be a software program written in C++, VHDL code for an FPGA, scripts, etc. It is delivered with documentation in a "STAC Pack."
2. Other hardware and software required to carry out the operations defined by STAC-A2. This includes processors and memory and often includes a server, operating system, compiler, math libraries, etc.

The next two sections discuss each layer in turn.

## 2.1 Understanding this implementation of STAC-A2

This project used a STAC-A2 Implementation developed by NVIDIA, called the STAC-A2 Pack for CUDA 5.5. NVIDIA provided the following description:

*The NVIDIA implementation of STAC-A2 takes advantage of the throughput optimized NVIDIA GPU and the latency optimized Intel Xeon CPU, allocating each task to the component most suited for it and building an overall system that combines speed and efficiency.*

*This implementation is based on CUDA 5.5 and uses the highly optimized libraries delivered with CUDA: cuBLAS, the GPU-enabled implementation of the linear algebra package BLAS, and cuRAND, a parallel and efficient GPU implementation of random number generators.*

*The different components of the implementation are designed in a modular and maintainable framework using object-oriented programming in CUDA/C++. Highly efficient implementations of the key algorithms were developed using the fine-tuned library: CUB (<http://nvlabs.github.io/cub/>).*

*The implementation uses a fast approach that fits in GPU memory to solve smaller cases. For larger cases, the solution switches to an “out-of-core” method which stores additional information in the CPU memory and moves data between CPU and GPU when needed. The building blocks to compute the Greeks are written as simple algorithms that work on a single GPU. The Greeks implementation uses as many GPUs as available to schedule the different scenarios and to enable linear scalability.*

*The implementation was developed using the different tools provided by NVIDIA to help debug and profile CUDA code. Those tools include NVIDIA Nsight for Visual Studio Edition and NVIDIA Nsight for Eclipse Edition.*

## 2.2 Understanding the rest of the SUT

The implementation relied on the following other components in this project:

- NVIDIA CUDA 5.5
- NVIDIA cuRAND and cuBLAS 5.5
- GNU OpenMP library v4.4.7
- CUB library v1.0.2
- Eigen 3 library v3.1.3
- 2 x NVIDIA K20Xm Co-Processor @ 2600MHz (memory), 784MHz (graphics) (with ECC enabled)
- IBM System X iDataPlex dx360 M4 Server
- 2 x Intel Xeon E5-2660 @ 2.20GHz (SandyBridge)
- CentOS Linux 6.4
- 96GB DDR3 RAM

NVIDIA provided the following information about its products used in this SUT:

*The K20Xm GPU was launched in 2012 as part of NVIDIA’s Tesla Scientific and Technical computing line of accelerators. The Kepler generation of GPUs combine class leading performance, programmability features, and energy efficiency in a commodity package to delivery over 1 TFLOP of IEEE compliant double precision math per card. It has 2688 cores running at 784MHz standard, 6GB of high speed ECC protected GDDR5 RAM, and connects with the CPU via 16 lane bi-directional PCIe Gen 2 links.*

*CUDA 5.5. is the latest production version of the CUDA C/C++ toolchain (toolkit, libraries, debugger and profilers). CUDA languages and tools are rapidly becoming the de facto standard for accelerated high performance computing among the scientific, engineering, and financial sectors. The cuRand and cuBLAS libraries are delivered as part of the CUDA platform and provide a wide range of high performance functionality for developers to leverage.*

Using *cuRand*, developers can generate identical pseudo or quasi-random sequences on the CPU or GPU, in blocks or one at a time, using a variety of quality RNGs. Full details of the RNGs supported and results from quality testing are in the documentation (<http://docs.nvidia.com/cuda/curand/index.html>). The *cuBLAS* library provides a full range of GPU accelerated level 1, 2, and 3 BLAS linear algebra routines.

*CUB* is an open source project that provides high performance primitives for common GPU programming tasks such as sorting and reduction at all levels of the programming model (device, block, warp, and thread).

IBM provided the following information about its products used in this SUT:

*The IBM System x® iDataPlex® dx360 M4 is designed to optimize density and performance within typical data center infrastructure limits. The unique half-depth form factor is designed to help you improve compute density in your space-constrained data center while also providing outstanding performance with outstanding power and cooling efficiencies. The latest generation servers support up to two 2.7 GHz 12-core Intel Xeon E5-2600 v2 Series processors, up to 512 GB of DDR3 memory per server, with two dedicated x16 PCIe 3.0 slots per server, and optional expansion for two double-width PCIe slots for GPUs or coprocessors. For more information, please see: <http://www-03.ibm.com/systems/x/hardware/highdensity/dx360m4/>*

A detailed STAC Configuration Disclosure for the SUT in this report is available to premium members of the STAC Benchmark Council at <http://www.stacresearch.com/node/15806>. If you get an access-denied message and would like to request access, use the request form at: <http://www.stacresearch.com/node/15808>

## 3. Project participants and responsibilities

The following firms participated in the project:

- NVIDIA
- IBM
- STAC

The Project Participants had the following responsibilities:

- NVIDIA wrote the STAC-A2 Pack for CUDA 5.5 in accordance with the STAC-A2 Benchmark specifications, configured the hardware and software to be tested, and sponsored the report.
- IBM provided the test server.
- STAC inspected and tested the STAC-A2 Pack for conformance to specifications, inspected the test system, executed the tests, and prepared the STAC Report and STAC Configuration Disclosure.

## 4. Contacts

- NVIDIA: Alastair Houston ([ahouston@nvidia.com](mailto:ahouston@nvidia.com)), or John Ashley ([jashley@nvidia.com](mailto:jashley@nvidia.com)).
- IBM: Joshua Blumert, Storage Solutions Engineering, [blumert@us.ibm.com](mailto:blumert@us.ibm.com), tel: +1-732-926-2793. See also the IBM Wall Street Center of Excellence: <http://www-03.ibm.com/systems/services/briefingcenter/wscoe/>

## 5. Results status

- These benchmark specifications were developed by the STAC-A2 Working Group of the STAC Benchmark Council. The specifications are still in beta testing, but the results can be fairly compared to other STAC-A2 Benchmark results with the same identifier (including version number). If the specifications change before they are finalized, the version number will also change. Comparisons of these results to results with a different version number in the identifier are not valid.

- These test results were audited by STAC, as indicated in the Responsibilities section above. As such, they are official results. For details, see [www.STACresearch.com/reporting](http://www.STACresearch.com/reporting).
- The vendors attest that they did not modify the SUT during the Audit.

## 6. Specifications

This project followed the STAC-A2 Benchmark specifications Rev P. Qualified members of the STAC Benchmark Council can access these specifications and accompanying STAC Test Harness Software, as well as request vendor STAC Packs for STAC-A2 at [www.STACresearch.com/a2](http://www.STACresearch.com/a2).

Note that the revision number of the specifications is irrelevant to whether these results can be compared to other STAC-A2 results. Only the ID for each benchmark (including its version number) can be used to determine results compatibility. See Section 1.3 for details.

## 7. Limitations and clarifications

- The rules regarding where an implementation must insert the  $T_{\text{submit}}$  and  $T_{\text{complete}}$  timestamps varies by operation. The philosophy behind these rules is to reflect the measurement that a user would care about in the real world. For example, component operations like square root normally execute in the context of a larger algorithm, whereas end-to-end operations such as Greeks are meaningful in isolation. This affects what preparation steps (e.g., memory allocation, transfer of code to a co-processor, etc.) may occur outside the boundaries of test timing. These details can also differ for cold and warm runs (described in Section 9). For details, see the Policies section of the specifications workbook in [3].
- The quality tests of UNR and CORRAND only use a subset of the output of these Operations, in order to limit the time it takes to execute the quality tests. However, the subset is still a large sample and is not known to the vendor in advance.
- In this version of STAC-A2, there are certain situations in which it is possible for the Implementation to infer that it is not being tested for quality and to execute an alternate (lower quality) code path. The only check for this violation of STAC-A2 policy is eventual discovery by STAC Benchmark Council members who have access to the Implementation source code, exposing the vendor to reputational risk.
- The quality visualizations in Section 8.2 are for indicative purposes only. The code that produces them is not rigorously analyzed. Rather than these charts, the quality benchmarks in Section 8.1 are the definitive measures of the quality of the Implementation with respect to a given Operation.

## 8. Quality of this implementation

### 8.1 STAC-A2 quality benchmarks

Table 2 compiles the quality scores for this implementation with respect to each STAC-A2 Operation. Note that some Operations have multiple quality metrics. Each quality benchmark compares an algorithm's results to those obtained from a more reliable source. The nature of that source depends on the algorithm.

Below is a summary of each quality benchmark. Note that these are only summaries. The ultimate expression of each quality benchmark is in the source code to the STAC-A2 Quality Assessment (QA) tool, which is available to qualified members of the STAC Benchmark Council via [www.STACresearch.com/a2](http://www.STACresearch.com/a2).

#### Special functions (SQRT, EXP, LOG)

Each quality benchmark for the special functions (SQRT.ERR, EXP.ERR, LOG.ERR) represents the error in the units-in-the-last-place (ULP). ULP, which (roughly speaking) measures the gap between the last digit of a calculated result and its theoretical value, is a common way to measure the accuracy of floating-point calculations. The STAC-A2 QA tool makes use of a library with greater than 64 bits of precision in order to calculate reference values. The closer each benchmark is to 0.5, the better. For more background on ULP calculations, see [6].

## Randoms

As explained above, STAC-A2 allows an implementer to use any random-number generator (RNG) for the unit-normal random and correlated-random performance benchmarks. (Note that when benchmark results are reported, the RNG must be disclosed in the STAC Configuration Disclosure.)

Two aspects of the numbers generated by the RNG are essential to quality: their normality and their correlation (or lack thereof).

STAC-A2 measures normality by applying the Anderson-Darling test.[7] This test transforms the generated numbers to a uniform distribution using the assumption that the data are from a normal distribution, then it tests the uniformity of that distribution. The STAC QA tool generates Anderson-Darling statistics for multiple subsets of the generated values, then uses a p-value from those statistics to determine uniformity, hence normality of the underlying values (see [8] for more background). The ideal score of this benchmark (UNR.NORM) is 1, but any score greater than 0.01 is considered aligned with theory. The normality benchmark for correlated randoms (CORRAND.NORM) is the same except that the matrix of correlated random numbers first undergoes an orthogonal transformation.

STAC-A2 uses the Hilbert-Schmidt Independence Criterion [9] to assess the independence of values produced in the UNR tests. Hilbert-Schmidt works by breaking the output data into segments, which it tests against each other for independence. The benchmark (UNR.INDEP) is the percent of randomly chosen sub-sequences that fail a Hilbert-Schmidt threshold. Thus, the ideal score is zero.

By contrast, correlated randoms are not meant to be independent but rather should be correlated as specified by the correlation matrix provided to the implementation at run time. STAC-A2 measures the actual correlation of the generated numbers, compares those correlations to the input correlations, and calculates the root-mean-square error. Again, a lower score on this benchmark (CORRAND.CORR) is better, with the limit being 0.

## Path generation

STAC-A2 applies several quality tests to the Monte Carlo paths produced by the benchmark implementation.

PATHGEN.PVSTDERR represents the worst-case standard error when measured across five different assets. (This metric is determined empirically, not estimated from the number of paths.) Lower is better, with 0 as the limit.

PATHGEN.ICORR assesses how well the implied correlations from the paths correspond to the input correlation matrix. This benchmark represents the smallest significance level for which all observed correlations are contained in confidence intervals constructed around the input correlations. Lower is better, and 0 is the limit.

PATHGEN.ERR1 is the worst-case root-mean-squared error of the asset prices across the 5 assets. Again, lower is better, with a limit of 0.

PATHGEN.ERR2 is the same as PATHGEN.ERR1 except with respect to the error in the variance in the log of the price.

## European option pricer

To assess the quality of the implementation before layering on the complexity of early execution, STAC-A2 calculates five prices for single-asset European options using a Heston closed form formula and compares those to the values for those same options obtained from the implementation. This benchmark (OPTIONPV.ERR) represents the worst-case relative error across all these options, so the closer to zero the better.

## American option pricer

As explained earlier, STAC-A2 specifies Longstaff-Schwartz as the method for pricing options with early-execution. The STAC-A2 QA tool tests the quality of the Longstaff-Schwartz implementation by computing prices for a single-

asset option via a Black-Scholes binomial approximation and comparing this to the values output by the implementation. A lower score on this benchmark (EARLYEX.ERR) is better, with the limit being 0.

## Greeks

STAC-A2 subjects an implementation's Greek results to several quality tests. Each quality benchmark represents the absolute difference between the theoretical and observed values. (The input data are constant across projects, making absolute error a comparable metric.) Lower is better, with 0 being the limit.

The \*.1ASSET.ERR benchmarks compare Greeks on European options produced by the benchmark implementation to those calculated using a closed form Heston formula (the input parameters are chosen to allow a closed form solution). The ID "MV\_THETA" refers to theta from the model vega (as opposed to the basic Greek, theta).

Note that the STAC-A2 performance benchmarks involve Greeks on American options (using Longstaff-Schwartz to price the options in the presence of early exercise), while the quality benchmarks involve Greeks on European options. To compute the latter, the benchmark implementation is required to use the same Monte Carlo and finite-difference code that it uses for the American option case, while using the European option pricer created for the OPTIONPV.ERR quality benchmark.

The \*.2ASSET.ERR benchmarks are comparisons of Greeks on exchange options with a payoff described by  $[\max(S_1(T) - S_2(T), 0)]$  produced by the benchmark implementation to those calculated using the Margrabe formula, an alternate formula for pricing options. A Greek that differs by asset has two error measurements: one with respect to the first asset (ERR1), and one with respect to the second (ERR2).

The MODELVEGA error for each asset is the dot product of the model vega vector and vector of the partial derivatives of the model vegas with respect to implied volatility.

**NOTE: Greeks quality benchmarks have changed**

In October 2013, the STAC-A2 Working Group decided that the formulation of the quality benchmarks for Greeks was incorrect, because the observed values were generated by the standard GREEKS operation (which are for American options), while the theoretical values used for comparison were for Greeks on European options. Those quality benchmarks have now been removed from the suite. They have been replaced by corresponding quality benchmarks for Greeks on European options. Because they are different benchmarks, they carry different identifiers, each of which begins with "EUR\_".

Table 2

STAC-A2 (beta 2) Quality Benchmarks NVIDIA CUDA 5.5 / 2 x NVIDIA K20Xm co-processors / IBM System X iDataPlex dx360 M4 SUT ID: NVDA131118 Tested at the baseline problem size: 5 assets, 25K paths, 252 timesteps					
	SPECIFICATION ID	OPERATION	METRIC	SCALE	RESULT
Special functions	STAC-A2 . $\beta 2$ . SQRT . ERR	Square roots	p-value related to ULP error	Closer to 0.5 is better	0.50000000
	STAC-A2 . $\beta 2$ . EXP . ERR	Exponentials	p-value related to ULP error	Same as above	0.91400000
	STAC-A2 . $\beta 2$ . LOG . ERR	Logs	p-value related to ULP error	Same as above	0.53500000
Random number generation	STAC-A2 . $\beta 2$ . UNR . NORM	Unit-normal randoms	Normality. NIST p-value of uniformity of Anderson-Darling p-values.	Higher is better. Perfect score is 1. A score over 0.01 is considered aligned with theory.	0.69931257
	STAC-A2 . $\beta 2$ . UNR . INDEP	Unit-normal randoms	Independence. Percent of randomly chosen sub-sequences that fail a Hilbert-Schmidt threshold.	Lower is better, with a limit of zero.	0.04444444
	STAC-A2 . $\beta 2$ . CORRAND . NORM	Correlated randoms	Normality after orthogonal transformation. NIST p-value of uniformity of Anderson-Darling p-values.	Higher is better. Perfect score is 1. A score over 0.01 is considered aligned with theory.	0.65793331
	STAC-A2 . $\beta 2$ . CORRAND . CORR	Correlated randoms	Correlation error. RMS error of observed correlations vs input correlations.	Lower is better, with a limit of zero.	0.00513411
Path generation and valuation	STAC-A2 . $\beta 2$ . PATHGEN . PVSTDERR	Path generation	Worst case standard error across 5 assets	Lower is better, with a limit of zero.	0.02164937
	STAC-A2 . $\beta 2$ . PATHGEN . ICORR	Path generation	Implied correlation vs input correlation	Same as above	0.03116798
	STAC-A2 . $\beta 2$ . PATHGEN . ERR1	Path generation	Error in price mean	Same as above	0.01081228
	STAC-A2 . $\beta 2$ . PATHGEN . ERR2	Path generation	Error in price variance	Same as above	0.01948263
	STAC-A2 . $\beta 2$ . OPTIONPV . ERR	Option present value	Error in European option price	Same as above	0.02521222
	STAC-A2 . $\beta 2$ . EARLYEX . ERR	Early exercise	Error vs Black-Scholes binomial approximation	Same as above	0.00700000

Table 3

<b>STAC-A2 (beta 2) Quality Benchmarks</b> <b>NVIDIA CUDA 5.5 / 2 x NVIDIA K20Xm co-processors / IBM System X iDataPlex dx360 M4</b> <b>SUT ID: NVDA131118</b> <b>Tested at the baseline paths (25K) and timesteps (252), with assets as noted</b>					
SPECIFICATION ID	OPERATION	METRIC	SCALE	RESULT	
<b>Single-asset tests</b>					
STAC-A2.β2.EUR_THETA.1ASSET.ERR	Greeks	Error in observed Greek for European option vs theoretical computed from closed form Heston.	Lower is better. Limit is 0.	0.00182239	
STAC-A2.β2.EUR_RHO.1ASSET.ERR	Greeks	Same as above	Same as above	0.00246826	
STAC-A2.β2.EUR_DELTA.1ASSET.ERR	Greeks	Same as above	Same as above	0.00324071	
STAC-A2.β2.EUR_GAMMA.1ASSET.ERR	Greeks	Same as above	Same as above	0.21375827	
STAC-A2.β2.EUR_KAPPA.1ASSET.ERR	Greeks	Same as above	Same as above	0.00000095	
STAC-A2.β2.EUR_MV_THETA.1ASSET.ERR	Greeks	Same as above	Same as above	0.01269887	
STAC-A2.β2.EUR_KHI.1ASSET.ERR	Greeks	Same as above	Same as above	0.00017362	
STAC-A2.β2.EUR_V0.1ASSET.ERR	Greeks	Same as above	Same as above	0.00160416	
<b>Two-asset tests</b>					
<b>Greeks</b>	STAC-A2.β2.EUR_THETA.2ASSET.ERR	Greeks	Error in observed Greek for exchange option vs theoretical computed via Margrabe formula.	Same as above	0.01895833
	STAC-A2.β2.EUR_RHO.2ASSET.ERR	Greeks	Same as above	Same as above	0.06058583
	STAC-A2.β2.EUR_DELTA.2ASSET.ERR1	Greeks	Same as above	Same as above	0.03719969
	STAC-A2.β2.EUR_DELTA.2ASSET.ERR2	Greeks	Same as above	Same as above	0.03485827
	STAC-A2.β2.EUR_GAMMA.2ASSET.ERR1	Greeks	Same as above	Same as above	0.17716143
	STAC-A2.β2.EUR_GAMMA.2ASSET.ERR2	Greeks	Same as above	Same as above	0.17515517
	STAC-A2.β2.EUR_CROSSGAMMA.2ASSET.ERR	Greeks	Same as above	Same as above	0.21969644
	STAC-A2.β2.EUR_MODELVEGA.2ASSET.ERR1	Greeks	Same as above	Same as above	0.09142919
	STAC-A2.β2.EUR_MODELVEGA.2ASSET.ERR2	Greeks	Same as above	Same as above	0.10205244
	STAC-A2.β2.EUR_CORRVEGA.2ASSET.ERR	Greeks	Same as above	Same as above	0.01918897

## 8.2 Quality visualizations

While the quality benchmarks above are the definitive expression of algorithm quality in STAC-A2, visualizations are also helpful. Several of these are provided below. Similar to the quality benchmarks, these charts compare the algorithm's results to those obtained from a more reliable source. For example, Figures 2 and 3 compare the normality of the output of the unit-normal randoms (UNR) Operation to values obtained by theory.<sup>4</sup>

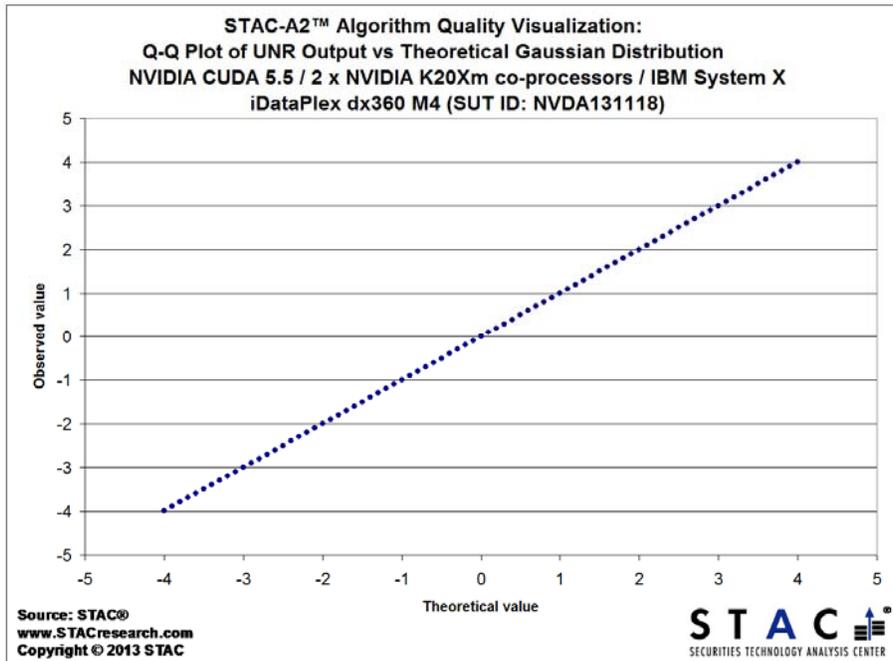


Figure 2

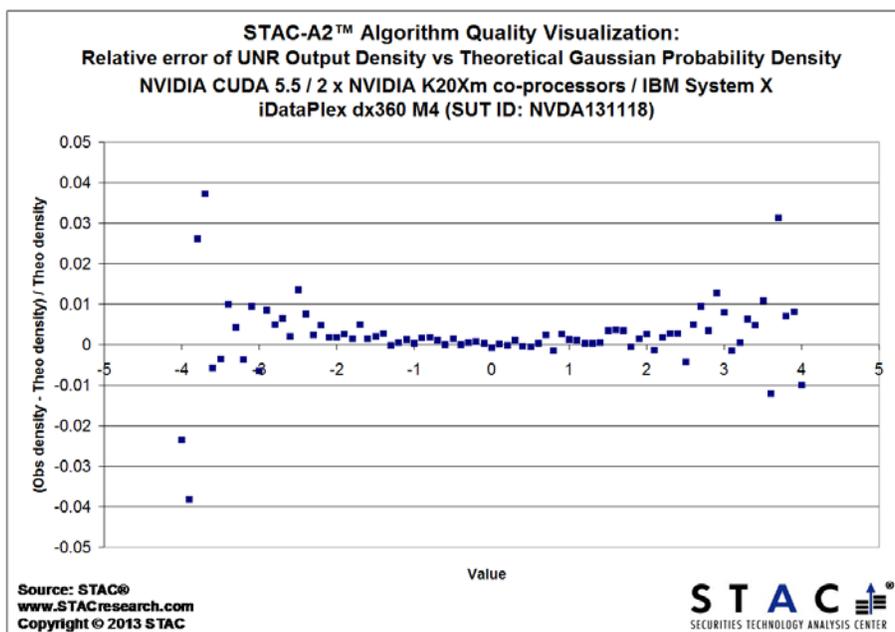


Figure 3

<sup>4</sup> Note that many of the charts in this section are different from those in STAC Reports for STAC-A2 prior to December 2013. Working group members asked to convert the charts of observed and theoretical values into charts of relative error.

Figure 4 plots the error in the present value for a single-asset European call option provided by this Monte Carlo implementation relative to that derived from the closed form solution.

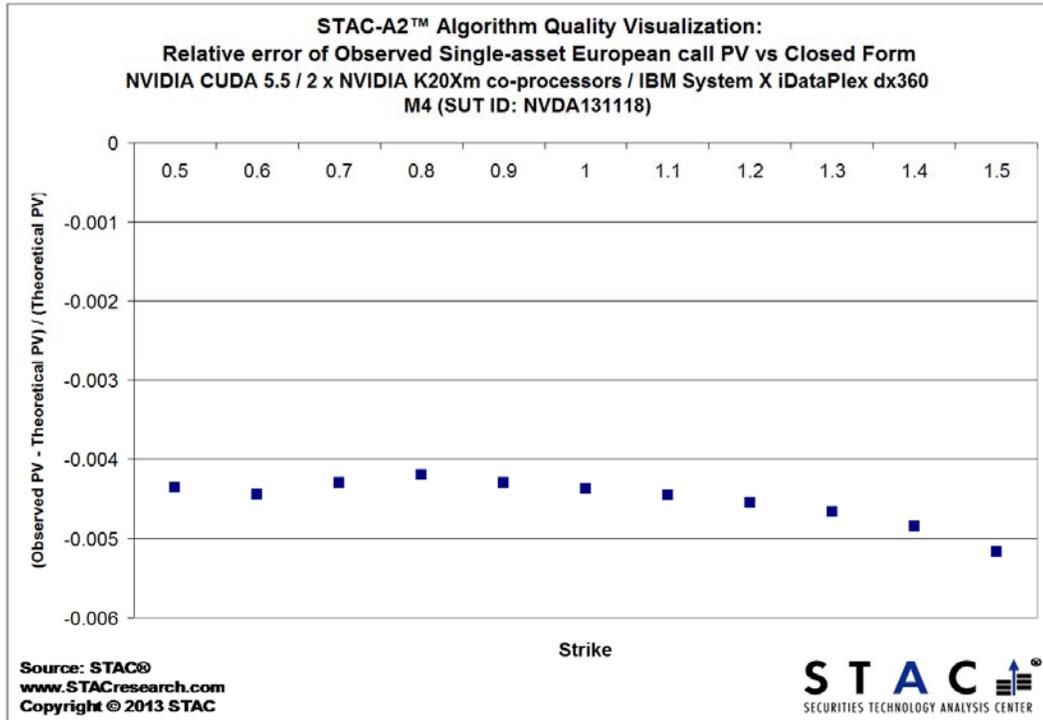


Figure 4

Figure 5 plots the relative error of the mean of the variance path.

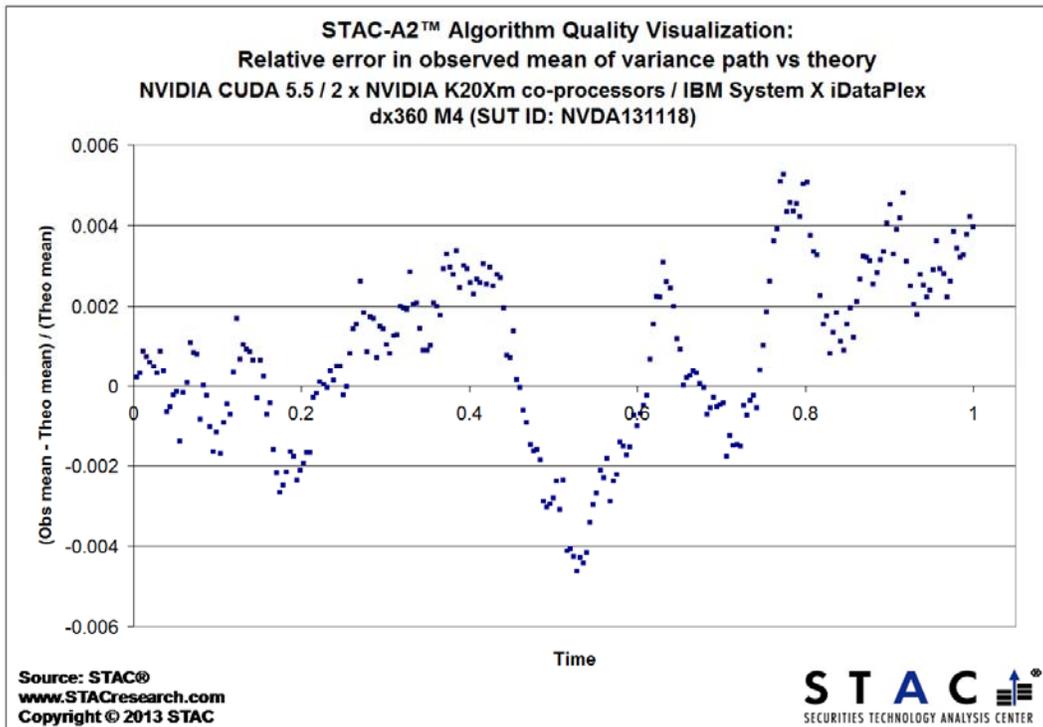


Figure 5

Figure 6 plots the relative error of the variance of the variance path.

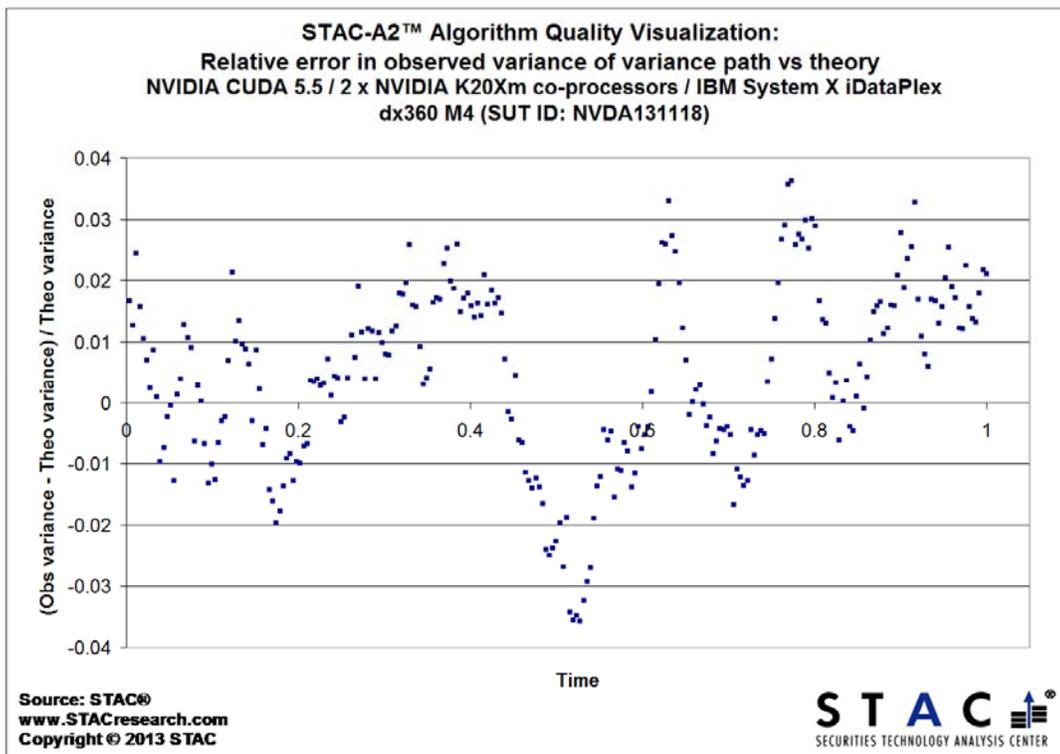


Figure 6

Figure 7 plots the relative error of the mean of the path of log prices.

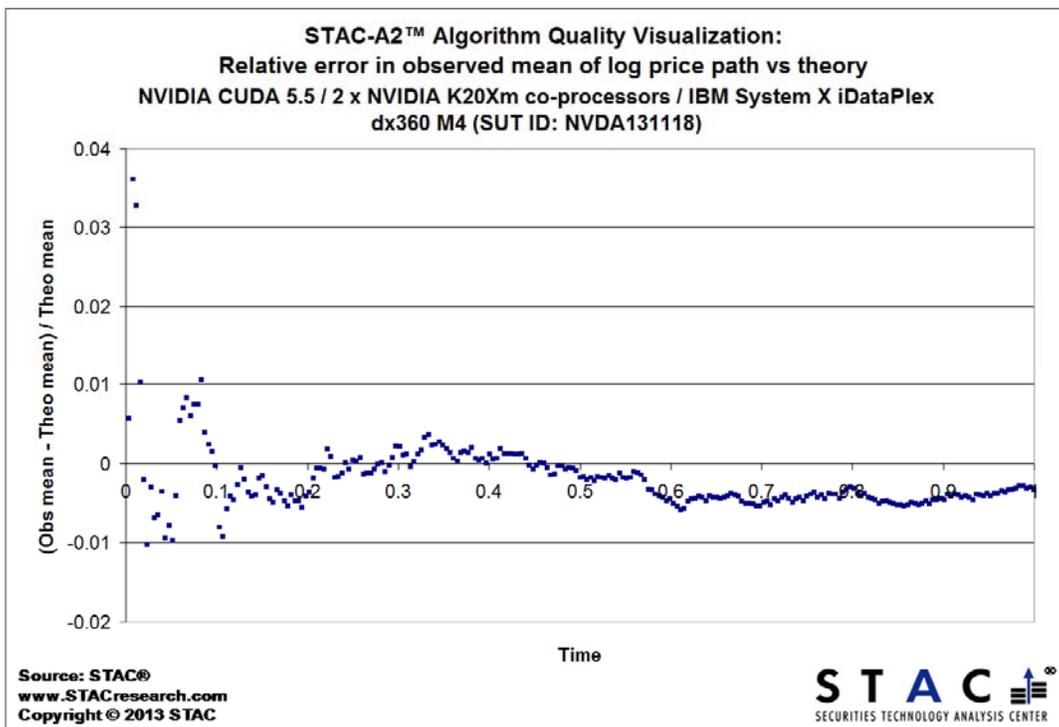


Figure 7

Figure 8 plots the relative error of the variance of the log price path.

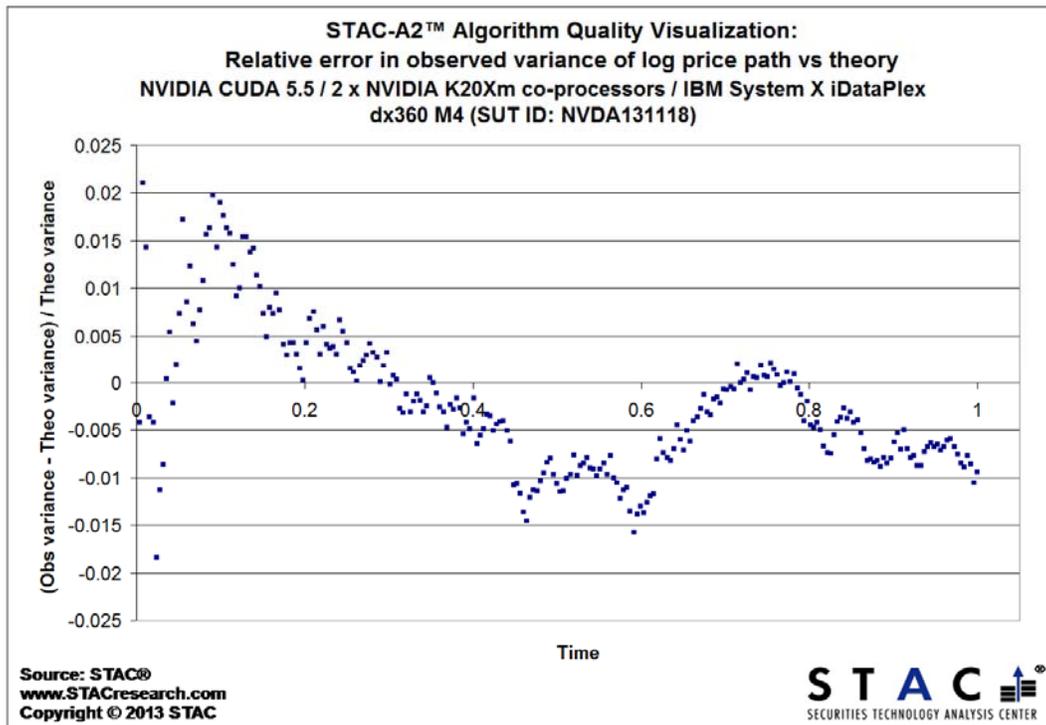


Figure 8

Figure 9 plots the relative error of the covariance of the log price path and variance path.

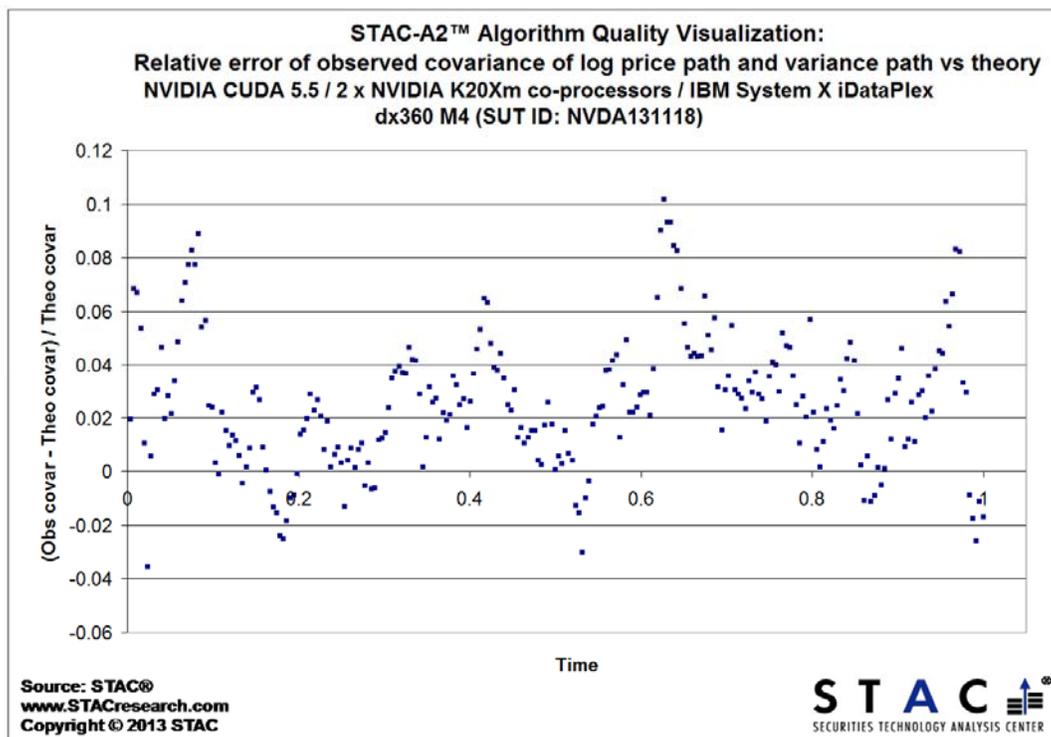


Figure 9

Figure 10 plots the relative error of present values for five multi-asset cases.

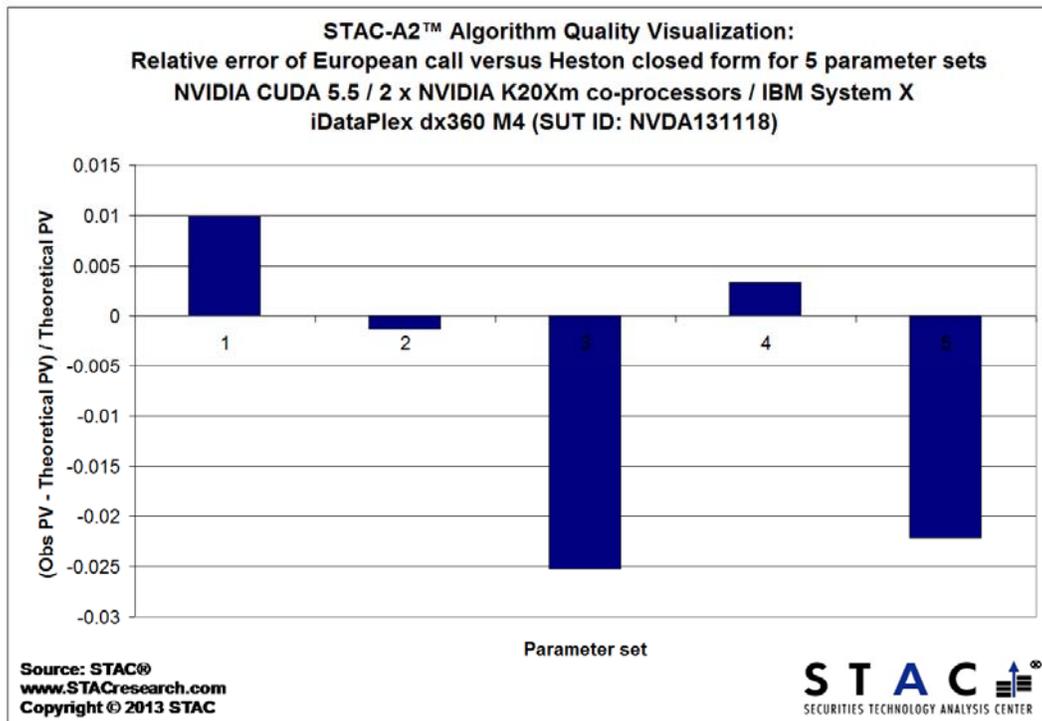


Figure 10

Figure 11 shows the observed European option price for a best-of-5 option over its life, versus the strike.

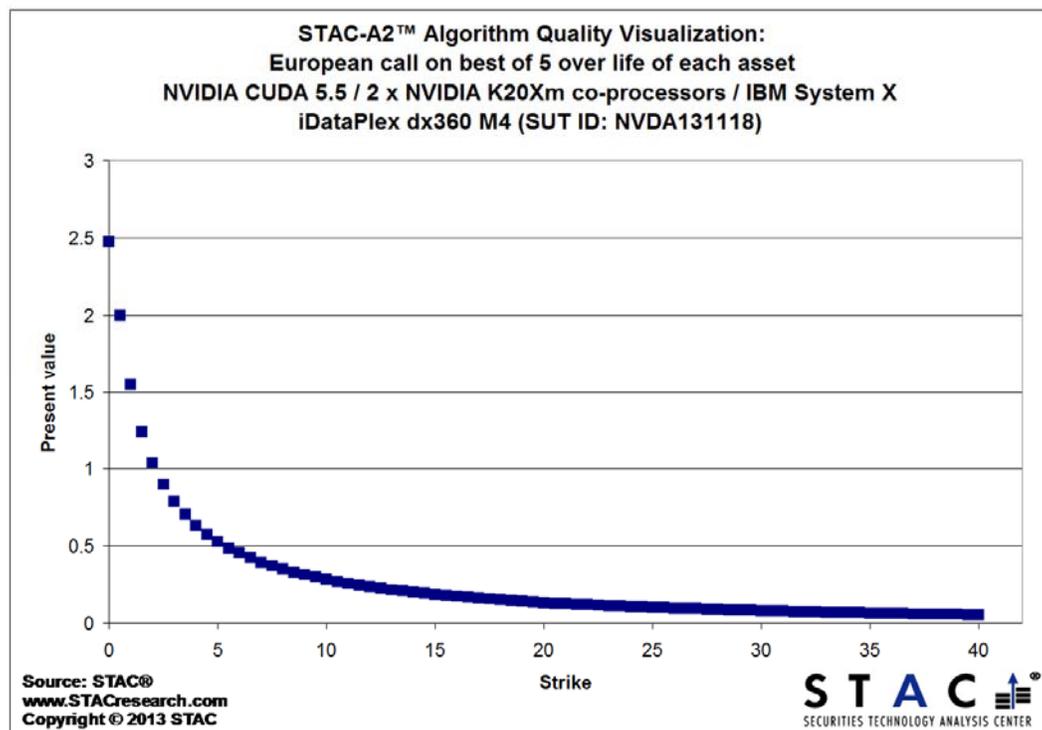


Figure 11

## 9. Speed benchmarks

STAC-A2 requires the implementation to report the Elapsed Time ( $t_{\text{complete}} - t_{\text{submit}}$ ) for component Operations as well as the end-to-end GREEKS Operation. These tests involve repeated execution (five times) of a given Operation without restarting the STAC-A2 Implementation. Thus, each Operation has one "cold" run and four "warm" runs, the latter of which benefit from pre-allocated memory and other efficiencies. Whether a cold run or warm run is more realistic depends on the use case one has in mind. A cold run simulates a deployment situation in which a risk engine starts up in response to a request. A warm run simulates a case in which an engine is already running, with sufficient memory allocated to handle the request.

Table 3 contains three results for each operation: the mean of all five runs, the result for the first (cold) run, and the mean of results for the subsequent four runs. All benchmarks are at the baseline problem size.

**Table 4**

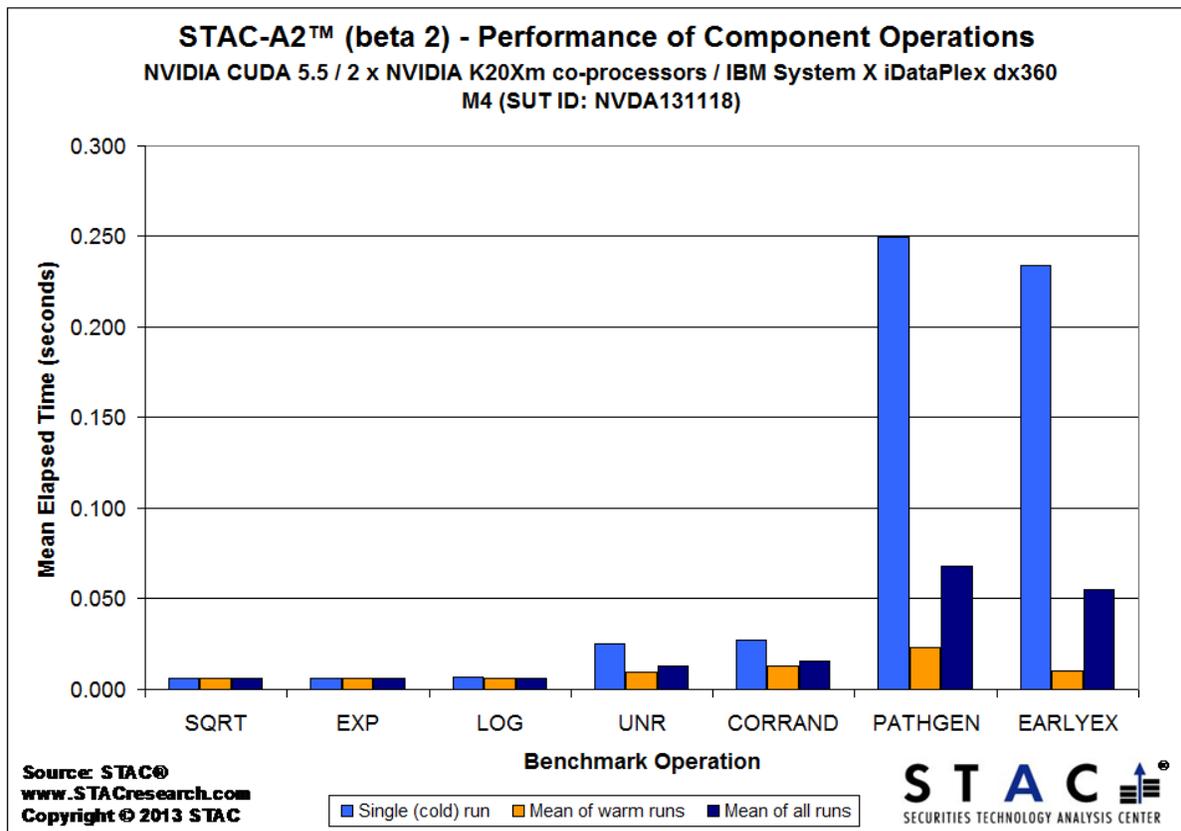
<b>STAC-A2™ (beta2) Performance Decomposition of Baseline Workload</b> <b>NVIDIA CUDA 5.5 / 2 x NVIDIA K20Xm co-processors / IBM System X iDataPlex dx360 M4</b> <b>(SUT ID: NVDA131118)</b> <b>Problem size: 5 Assets, 25K paths, 252 timesteps (63 million values in path cube)</b> <b>(Elapsed times in seconds)</b>				
Performance benchmark	Description	Mean, all runs	Cold run	Mean of warm runs
STAC-A2.β2.SQRT.TIME	Time to compute square roots	0.006	0.006	0.006
STAC-A2.β2.EXP.TIME	Time to compute exponentials	0.006	0.006	0.006
STAC-A2.β2.LOG.TIME	Time to compute logs	0.006	0.006	0.006
STAC-A2.β2.UNR.TIME	Time to compute unit-normal randoms	0.013	0.025	0.010
STAC-A2.β2.CORRAND.TIME	Time to compute correlated randoms	0.016	0.027	0.013
STAC-A2.β2.PATHGEN.TIME	Time to generate paths	0.068	0.249	0.023
STAC-A2.β2.EARLYEX.TIME	Time to compute payoff with early exercise	0.055	0.234	0.011
STAC-A2.β2.THETA.TIME	Time to compute theta	0.168	0.662	0.044
STAC-A2.β2.RHO.TIME	Time to compute rho	0.150	0.612	0.034
STAC-A2.β2.DELTA.TIME	Time to compute delta	0.239	0.726	0.117
STAC-A2.β2.GAMMA.TIME	Time to compute gamma	0.262	0.764	0.137
STAC-A2.β2.CROSSGAMMA.TIME	Time to compute cross-gamma	0.546	1.028	0.425
STAC-A2.β2.MODELVEGA.TIME	Time to compute model vega	0.355	0.818	0.239
STAC-A2.β2.CORRVEGA.TIME	Time to compute correlation vega	0.277	0.769	0.153
STAC-A2.β2.GREEKS.TIME	End-to-end time to compute all Greeks	0.770	1.258	0.648

For several of the component operations, Table 5 divides the problem size by the elapsed time in Table 4 to derive a speed. It does this for both cold and warm runs.

**Table 5**

Component Benchmarks Expressed as Speeds (Computed from the corresponding elapsed-time benchmarks above)		Cold run	Mean of warm runs
		STAC-A2.β2.SQRT.SPEED	Square roots per second
STAC-A2.β2.EXP.SPEED	Exponentials per second	9,849,906,191	10,007,942,812
STAC-A2.β2.LOG.SPEED	Logs per second	9,731,232,623	9,776,915,616
STAC-A2.β2.UNR.SPEED	Unit-normal randoms per second	2,517,080,187	6,592,716,618
STAC-A2.β2.CORRAND.SPEED	Correlated randoms per second	2,327,729,540	4,900,435,594

Figures 12 and 13 plot the mean Elapsed Time for each of the benchmarks to provide some insight into the relative time consumption of each Operation. Figure 12 looks just at the component Operations, while Figure 13 looks at each of the Greeks, as well as the GREEKS Operation, which generates all of the Greeks. Note that the Elapsed Time for GREEKS may be less than the sum of the Elapsed Time for the individual Greeks, because certain interim algorithm results can be re-used for multiple Greeks.



**Figure 12**

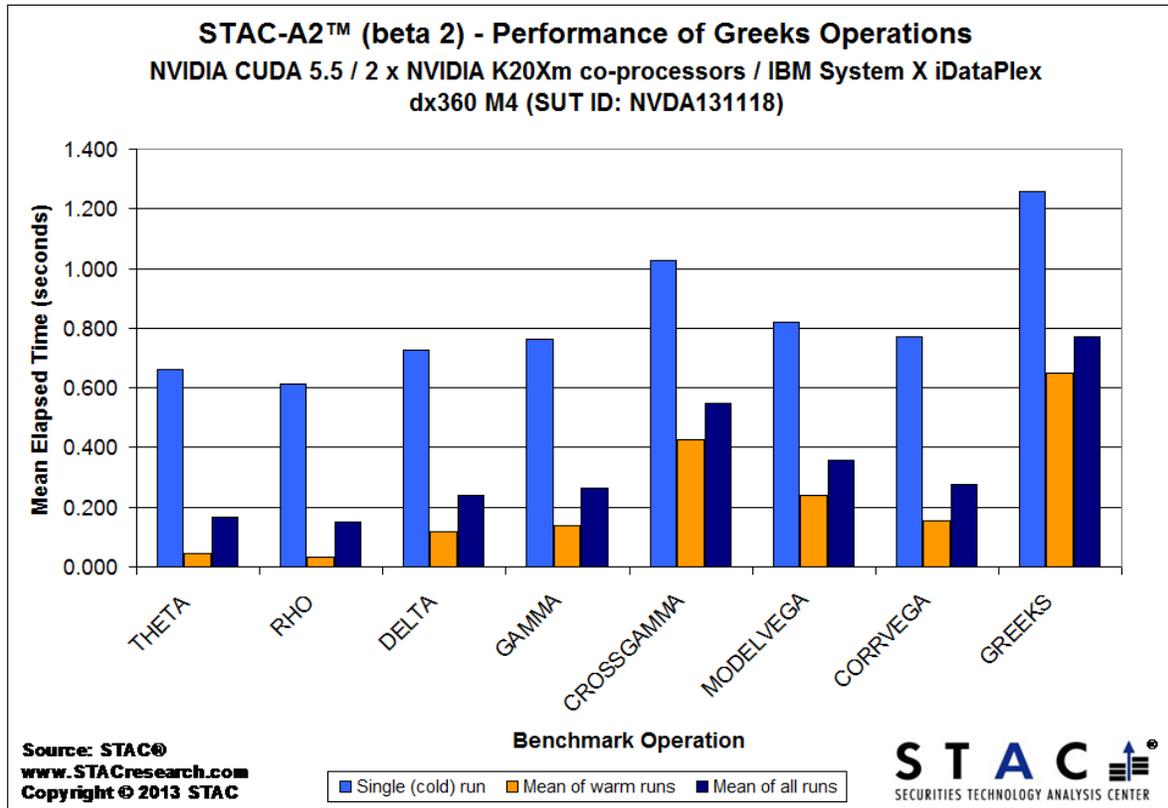


Figure 13

## 10. SUT Scaling

It is often instructive to see how the performance results for a given SUT depend on the scale of the SUT, by running the same fixed problem size on subsets of the SUT. In STAC-A2, a SUT Scale is defined as a subset of the SUT that is capable of independently processing the end-to-end Greeks operation. For example, in a SUT consisting of a grid of 8 servers each with 2 sockets and 4 cores per socket, SUT Scales might be defined as a single socket, 2 sockets, 2 servers, 4 servers, and 8 servers. STAC-A2 tests the baseline problem size at each of these SUT Scales and measures the Elapsed Time.

A vendor wishing to publish results is responsible for defining SUT Scales at which it wishes to demonstrate comparative performance. Since SUT Scales are architecture specific, STAC-A2 does not establish hard and fast definitions for them. However, the STAC-A2 Working Group does maintain best-practice recommendations. In the extreme, if a vendor does not believe that performance testing of subset of the SUT will provide any meaningful information, it may skip the SUT Scaling tests.

Table 5 lists the results, while Figure 14 plots the performance of progressively larger scales on the same chart, yielding a curve that describes the architecture’s scalability. Again, the results for cold and warm runs are shown separately.

In this case, NVIDIA chose to define SUT Scales in terms of the number of GPUs used and the number of CPU cores dedicated to serving those GPUs (cores being defined implicitly by the number of OpenMP threads). The full SUT had 2 GPUs with 4 CPU threads serving each.

Table 6

STAC-A2 SUT Scaling with Baseline Problem Size					
NVIDIA CUDA 5.5 / 2 x NVIDIA K20Xm co-processors / IBM System X iDataPlex dx360 M4					
SUT ID: NVDA131118					
Problem size: 5 Assets, 25K paths, 252 timesteps					
		COLD RUN		WARM RUNS	
SUT Scale	Description	Time to complete workload (seconds)	Speedup relative to 1 GPU, 1 CPU thread	Mean time to complete workload (seconds)	Speedup relative to 1 GPU, 1 CPU thread
1	1xGPU, 1x1 CPU thread per device	1.528	1.0	1.276	1.0
2	1xGPU, 2x1 CPU threads per device	1.525	1.0	1.254	1.0
3	1xGPU, 2x4 CPU threads per device	1.478	1.0	1.226	1.0
4	2xGPUs, 1x1 CPU threads per device (2 CPU threads)	1.277	1.2	0.648	2.0
5	2xGPUs, 1x4 CPU threads per device (8 CPU threads)	1.275	1.2	0.650	2.0

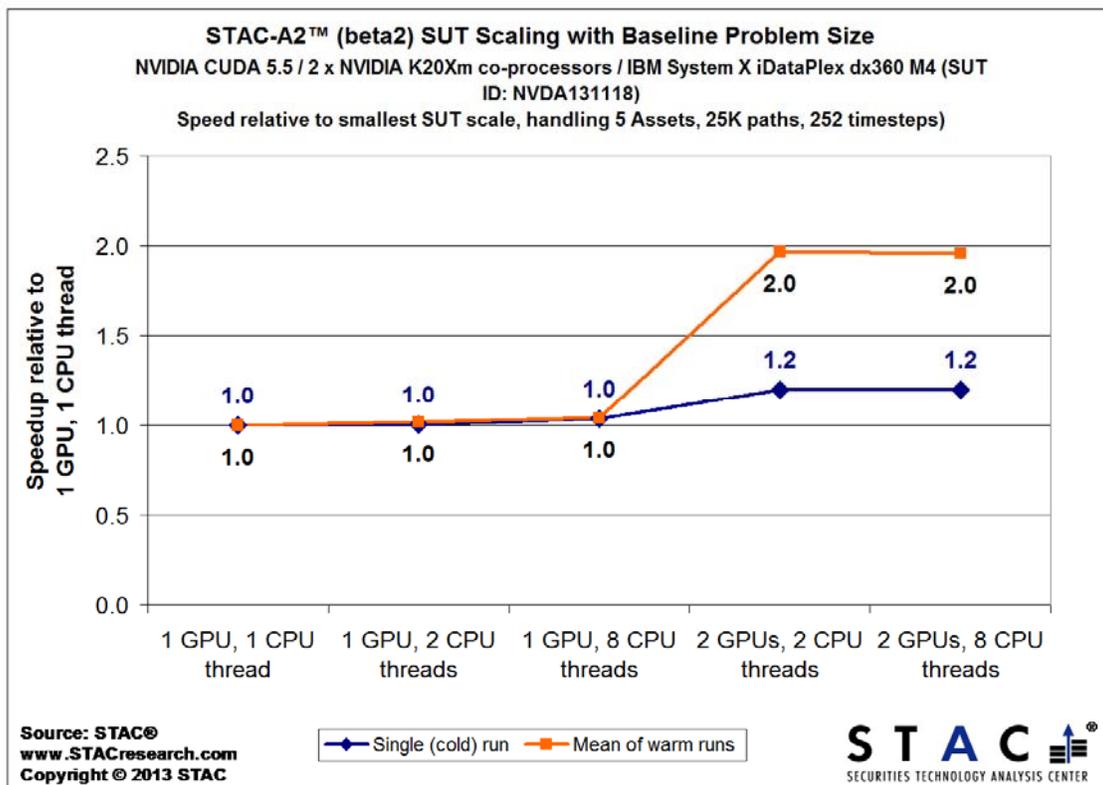


Figure 14

## 11. Capacity benchmarks

In addition to measuring the time it takes the SUT to process a fixed workload (the baseline problem size), STAC-A2 also measures the maximum workloads that can be processed in a fixed unit of time. Such benchmarks are a measure of capacity. The STAC-A2 workload has three dimensions: the number of assets to which the option applies, the number of Monte Carlo paths the implementation must use, and the number of timesteps each path must contain. STAC-A2 defines capacity benchmarks by allowing just one of these parameters to vary at a time:

1. GREEKS.MAX\_ASSETS: the highest number of assets that can be processed within the test window while holding paths and timesteps constant at their baseline values. This indicates the SUT's capacity to handle increasing basket sizes for the multi-asset option of STAC-A2 and perhaps other real-world operations that require simulations for many correlated assets.
2. GREEKS.MAX\_PATHS: the maximum number of paths in the Monte Carlo simulations that can be processed within the test window while holding assets and timesteps constant at their baseline values. Increasing the number of paths increases the accuracy of the resulting Greeks.

Table 7 contains the capacity benchmarks for this SUT. It bears noting that the STAC-A2 workload is roughly quadratic in assets and linear in paths. Hence, a SUT that shows a large increase in maximum paths relative to another may show a more modest increase in maximum assets.

Note that because of the number of problem sizes that must be tested in order to find the maxes (with unsuccessful runs taking a full 10 minutes), these results are based on cold runs. That is, each time the problem size changes, the implementation is restarted.

*Table 7*

<b>STAC-A2™ (beta 2) Capacity Benchmarks</b> <b>NVIDIA CUDA 5.5 / 2 x NVIDIA K20Xm co-processors / IBM System X iDataPlex dx360 M4</b> <b>(SUT ID: NVDA131118)</b>		
STAC-A2.β2.GREEKS.MAX_ASSETS	Max assets completed in 10 minutes with 25K paths and 252 timesteps (using cold test runs).	41
STAC-A2.β2.GREEKS.MAX_PATHS	Max paths completed in 10 minutes with 5 assets and 252 timesteps (using cold test runs).	8,500,000

## 12. Workload scaling

STAC-A2 capacity benchmarks are just two points on a "cube" of potential problem sizes. To provide a more complete picture of the capacity of the SUT, the STAC-A2 Test Harness subjects it to 125 combinations of assets, paths, and timesteps in order to reveal any cross-over points in the ability of the SUT to scale with the workload—and in order to provide meaningful reference points for as many end users as possible. (Again, these are based on single—i.e., cold--test runs.)

Note that the value of assets can be any integer between 1 and GREEKS.MAX\_ASSETS, while the paths value is one of the discrete values 10K, 25K, 50K, or 100K. Timesteps is a discrete value that corresponds to an option tenure (with one timestep per day: 6 months (126 timesteps), 1 year (252 timesteps), 2 years (504 timesteps), 5 years (1260 timesteps), and 10 years (2520 timesteps).

Figures 15-18 summarize the workload scaling results by plotting the execution time versus the basket size for options of different tenors. Each chart is for a fixed number of paths, which maps to a fixed Monte Carlo standard error. Figure 15 plots the curves when paths are set to 10,000. Figure 16 increases the paths to 25,000, while Figure 17 increases them to 50,000. Finally, Figure 18 plots the performance at 100,000 paths.

These charts use fixed log scales in order to enable comparisons to future SUTs, since performance is expected to improve over time. Each chart has markers on the far right y-axis to signify salient points for a human end user. For these purposes, any result set provided in 1 second or less is considered to be "real time". This scale is, of course, arbitrary. Each use case or user will have its own thresholds.

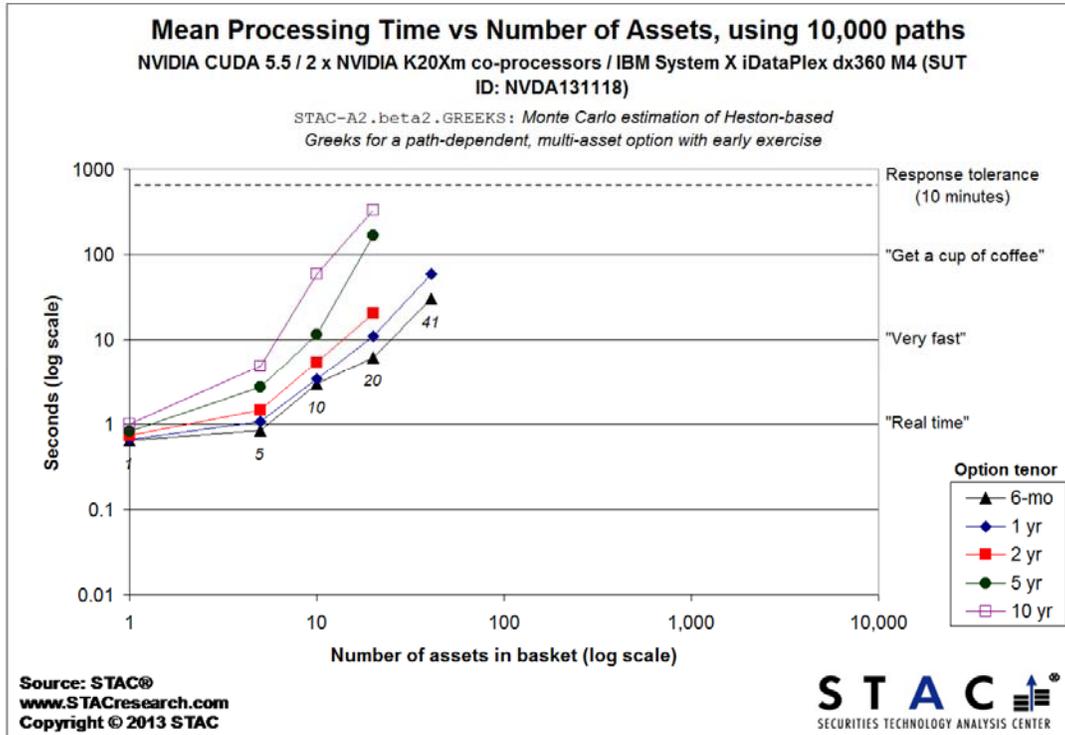


Figure 15

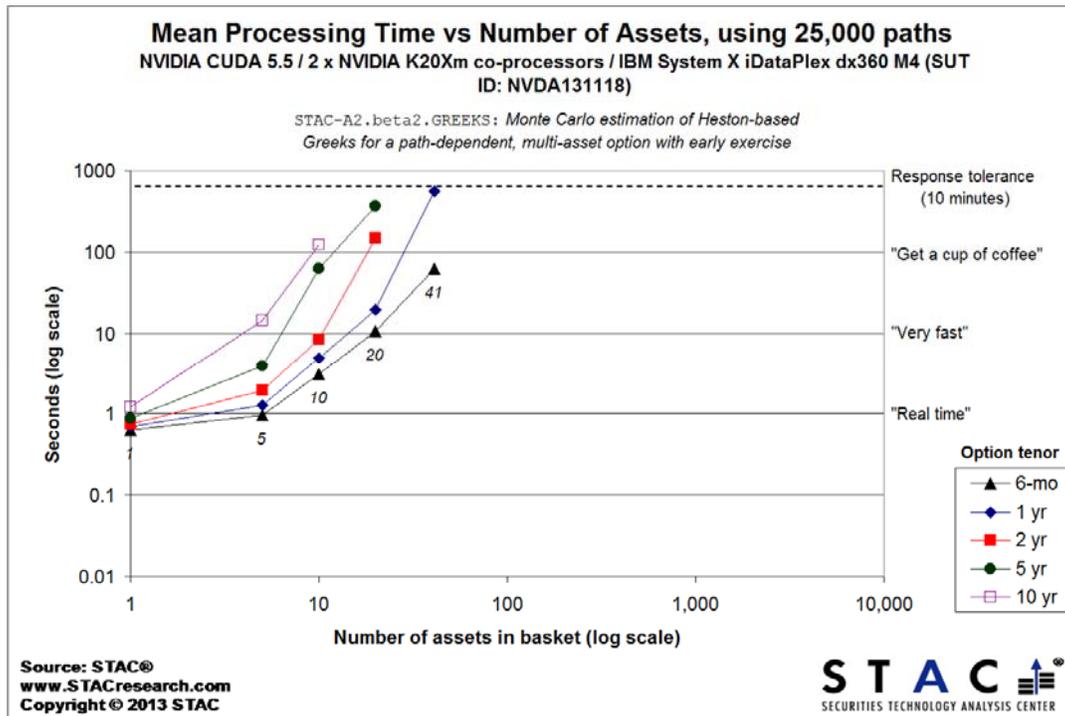


Figure 16

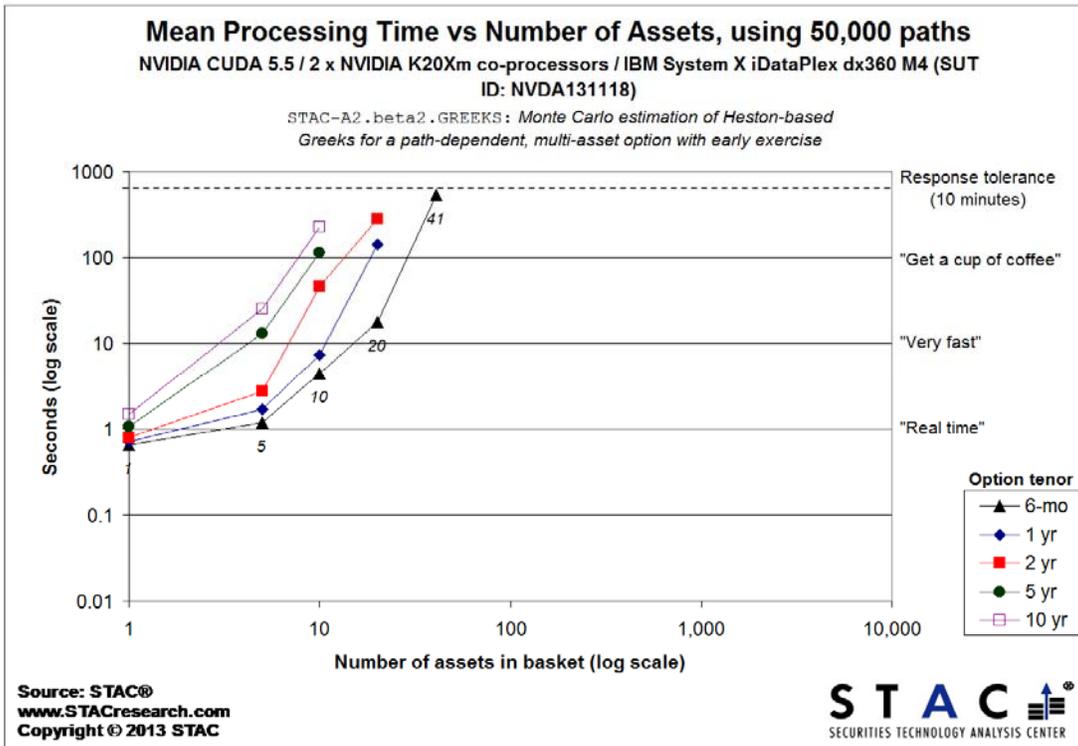


Figure 17

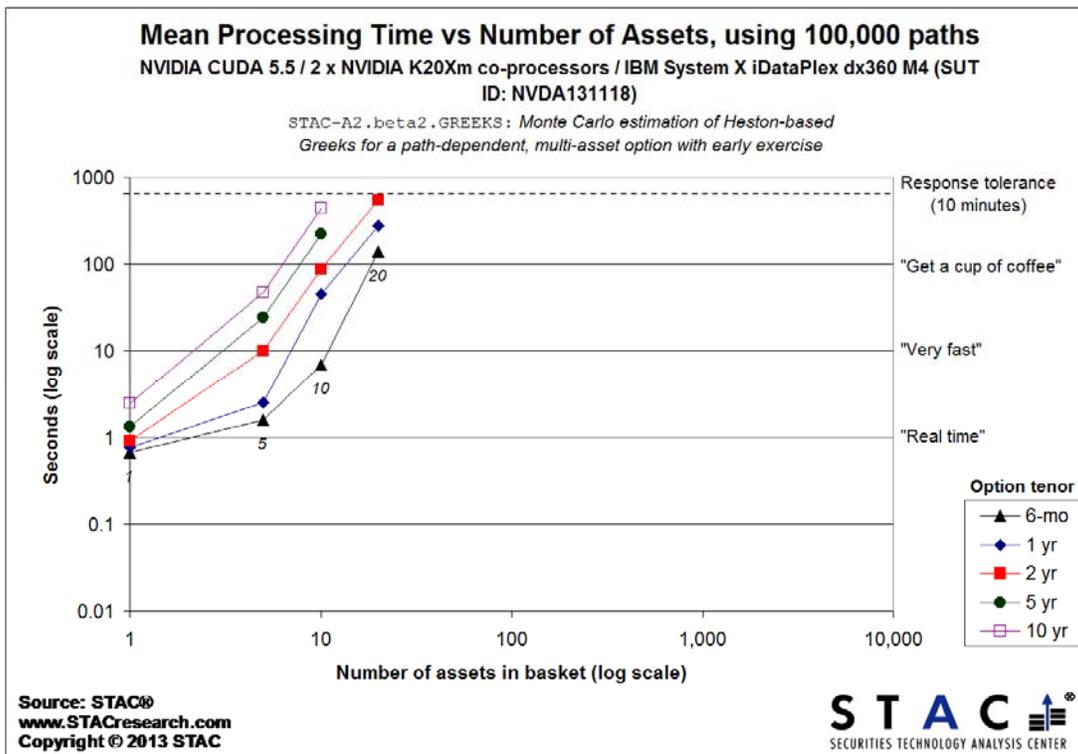


Figure 18

Note that the number of results plotted across these four charts is substantially less than 125. That is because some problem sizes were too big. Either they exceeded available memory or took longer than the 600-second (10-minute) threshold imposed on the tests.

## 13. Vendor Commentary

NVIDIA provided the following commentary:

*NVIDIA is pleased to partner with IBM to release our first STAC-A2 benchmark. The SUT tested here is a currently shipping configuration. Being the first publicly announced system to deliver STAC-A2 baseline results (STAC-A2.β2.GREEKS.TIME) in “real time” highlights the ability of GPUs to deliver breakthrough performance. This translates into either significant time to market advantage or cost savings that are achievable today with GPU accelerated computing.*

*Our accuracy results highlight the quality of the cuRand library, the CUDA 5.5 toolkit, and the quality of the CUDA accelerated Kepler GPU hardware. Our performance results here were all achieved in full IEEE double precision, with full accuracy in all math routines, and full ECC memory protection enabled.*

*Our results clearly demonstrate that GPU accelerated computing can address the most challenging real-world problems and deliver accurate results in transformational timescales. This enables more complex business, delivers better quality analytics and risk management, and does so in shorter timeframes with less hardware.*

*Please contact either NVIDIA or IBM if you have any questions about how to evaluate or begin applying GPU acceleration to your critical business problems.*

## About STAC

STAC® is a technology-research firm that facilitates the STAC Benchmark™ Council ([www.STACresearch.com/council](http://www.STACresearch.com/council)), an organization of leading financial institutions and technology vendors that specifies standard ways to assess technologies used in the financial markets. The Council is active in an expanding range of low-latency, big-compute, and big-data workloads.

STAC helps end-user firms relate the performance of new technologies to that of their existing systems by supplying them with STAC Benchmark reports as well as standards-based STAC Test Harnesses™ for rapid execution of STAC Benchmarks in their own labs. User firms do not disclose their results. Some STAC Benchmark results from vendor-driven projects are made available to the public, while those in the STAC Vault™ are reserved for qualified members of the Council (see [www.STACresearch.com/vault](http://www.STACresearch.com/vault)).

To be notified when new STAC Reports™ become available, please sign up for free at [www.STACresearch.com](http://www.STACresearch.com).

