End-User Driven Technology Benchmarks Based on Market-Risk Workloads

Peter Lankford
Securities Technology Analysis Center (STAC)

Lars Ericson
Catskills Research Company

Andrey Nikolaev
Intel Corporation

***Abstract*- Market risk management is a critical, resource-intensive task for financial trading firms. The industry relies heavily on innovation in technical infrastructure to increase the quality and quantity of risk management information and to reduce the cost of its production. However, until recently, the industry has lacked an independent standard for gauging the potential of new technologies to help. This changed when the STAC Benchmark™ Council developed STAC-A2™, a vendor-independent benchmark suite based on real-world market risk analysis workloads. It was specified by trading firms and made actionable by leading HPC vendors. Unlike vendor-developed benchmarks known to the authors, STAC-A2 satisfies all of the requirements important to end-user firms: relevance, neutrality, scalability, and completeness. Intel has demonstrated the utility of STAC-A2 for comparing successive generations of Intel® Xeon® processors.**

Keywords—market risk; Monte Carlo; options pricing; Heston; benchmark; STAC-A2; STAC; Intel

I. Introduction

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. Recent well-publicized events in the financial markets 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, such as shifts in interest rates. 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 has become 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 power and space required for calculations, to analyze more assets, to analyze more scenarios for more users, to increase the accuracy of the calculations, or all of the above.

No matter what the requirement, innovation in technology infrastructure has proven to be a crucial enabler of these business goals. New processors, memory, and interconnects, as well as innovative software libraries, development tools, and grid software can create favorable shifts in the tradeoffs (e.g., accuracy vs speed, capacity vs power consumption, etc.).

In evaluating the potential of new technologies to improve trading functions, end-user firms can benefit from standard benchmarks that enable vendors to publish apples-to-apples comparisons and that enable end-user firms to baseline their existing systems using the same tests.

Such standards exist in areas such as low-latency data distribution and time-series data management.[1] But when it comes to market-risk analysis, the industry has, until now, lacked a vendor-independent technology benchmark suite based on a realistic workload.

The STAC-A2™ Benchmark suite fills this gap. STAC-A2 is a set of test specifications based on a modern market-risk workload. It has been designed by leading trading firms with the input of key high-performance computing vendors. Unlike vendor-provided benchmarks, STAC-A2 satisfies the key customer requirements explained in Section III.

At the time of this writing, STAC-A2 Benchmarks are in a beta state as they undergo refinement. Like all other STAC Benchmark specifications, the exact details of STAC-A2 are confidential, available only to contributing members of the STAC Benchmark Council, a large group of trading organizations and vendors. This confidentiality supports a business model that incentivizes ongoing investment in the benchmarks. Contributing members can access the benchmark specifications, meeting notes, and discussion forums at [2].

This paper explains the STAC-A2 specifications at a high level and illustrates them by sampling some preliminary, unofficial results from Intel's work with the standard.

II. Analytic Operations

Under STAC-A2, the job of the "stack under test" (SUT) is to compute option-price sensitivities ("Greeks") for multiple assets by applying Monte Carlo methods to the Heston model [3], a popular approach in today's capital markets. In addition to benchmarking end-to-end calculation of Greeks, the suite scrutinizes specific layers of the computation.

A key influence on the speed of such analytics is the quality that is required (whether the relevant dimension of quality is precision, closeness of realized to theoretical values, etc.). All other things being equal, higher quality requires more processing time. As discussed in Section III, STAC-A2 does not impose minimum quality standards on an implementation but rather measures and reports its quality. The layers of STAC-A2 and their respective quality measures are:

1. **Double-precision exponential, log, and square root operations,** which are required for the inner loop of the Monte Carlo simulation. Quality is measured by relative error of 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
2. **Generation of independent unit normal random numbers**. Quality is measured by applying the Anderson-Darling test for normality [4] and a Hilbert-Schmidt Independence Criterion [5]. Anderson-Darling transforms the output data to a uniform distribution using the assumption that the data are from a normal distribution, then testing the uniformity of that distribution. Hilbert-Schmidt works by breaking the output data into segments, which it tests against each other for independence.
3. **Generation of correlated unit normal random numbers**. Quality is measured by applying the Anderson-Darling test for a multinormal distribution and computing the root-mean-square error of the output correlations relative to the input correlations.
4. **Single-asset path generation** using the Andersen QE method [6]. QE is one of two methods (the preferred method) proposed by Andersen for time-discretization and Monte Carlo simulation of Heston-type stochastic volatility models. Quality of the generated paths is assessed by computing the root-mean-square error of the mean and variance of the price paths relative to theory, and by the difference between the vanilla call and put prices obtained from the Monte Carlo simulation and the Heston closed-form formula.
5. **Multi-asset path generation**. Quality is assessed by computing the realized correlation matrices for each path, using those to compute a matrix of average correlations, then computing the root-mean-square difference between the average realized correlations and the input correlations.
6. **Early exercise**. Early exercise follows the approach of Longstaff and Schwartz [7]. Quality is assessed for single assets by pricing an American option under the Heston model calibrated to a flat volatility skew and comparing this with the Black-Scholes binomial model approximation given by [8].
7. **Greeks: Theta, rho, delta, gamma, cross-gamma, model vega, correlation vega**. Quality is assessed for single assets by comparing the Greeks obtained from the Monte Carlo with Greeks obtained from a Heston closed form formula for vanilla puts and calls. Quality is assessed for two assets by calibrating the Heston model to a flat volatility skew for two individual assets, applying the Margrabe formula for two-asset spread option pricing [9], and comparing this with spread option Greeks obtained from the Monte Carlo.

III. Scaling

The baseline speed benchmarks in STAC-A2 fix the problem size in order to enable consistent comparisons across multiple technology stacks. They also test the SUT in its entirety. But STAC-A2 also enables two kinds of scale tests:

1. **Workload scaling.** STAC-A2 allows two key dimensions of the workload to vary without limit:

   - The number of correlated assets. This corresponds to the size of the portfolio processed by the SUT.
   - The number of paths in the Monte Carlo simulations. Increasing the number of paths increases the accuracy of the resulting Greeks.

   Workload scaling enables benchmarks of the maximum capacity of the SUT with respect to portfolio size and number of paths. No matter how large the SUT, STAC-A2 measures how much work it can process within a set timeframe, holding everything but the scale dimension constant. Workload scaling also yields scale curves, where each point represents the speed at which the SUT is able to process a different size of workload.

2. **SUT scaling.** An official STAC-A2 report shows the performance of the SUT at various "SUT Scales". A SUT Scale is defined as a subset of the SUT that is capable of independently processing the end-to-end Greeks operation. Each implementation is responsible for defining its SUT Scales, and the tester is responsible for deciding which of those to test. 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 thread, 2 threads, 4 threads, a single server, 2 servers, 4, servers, and 8 servers. STAC-A2 requires both speed and capacity benchmarks to be run at multiple SUT Scales. Plotting the performance of progressively larger scales on the same chart yields a curve that describes an architecture's scalability.

IV. Benchmark Properties

STAC-A2 satisfies several properties that are important for a benchmark standard:

1) **Relevance**. STAC-A2 was designed by trading organizations who deal with market risk on a daily basis. While STAC-A2 does not specify a production-quality algorithm (i.e., not necessarily something on which a trading firm would run its business), it captures the essence of workloads used to analyze options and other derivatives with option-like properties. In addition, STAC-A2 measurements are expressed in terms that are meaningful to a business person at a trading firm, making it easy to relate the performance of an innovation to its economic impact.
2) **Neutrality**. Specified purely in mathematics and English, STAC-A2 is architecture neutral. Implementations have been and are being developed for leading vendor architectures. Implementations are possible across CPUs, GPUs, FPGAs, and other types of processors, as well as virtualized environments. Implementations can be written at a low level to be hardware aware, at a higher level that is partly or completely hardware agnostic, or at the highest level, taking advantage of financial analytic tools. This neutrality enables apples-to-apples comparisons of any layer of the solution stack. For example, STAC-A2 could be used to compare:
    - processors, servers
    - programming languages, compilers
    - analytic libraries
    - distributed computing software (e.g., grid middleware, Hadoop)
    - cloud services, including different service options from a single cloud provider
3) **Scalability**. As described in Section III, STAC-A2 is arbitrarily scalable. This enables it to assess performance from a single core to a large grid.
4) **Completeness**. The benchmark suite reveals the four properties of a SUT that are most important to risk-management technologists:
    a) Speed: the time it takes to obtain analytic results in a standard, usable form. As described in Section II, speed is measured for several discrete steps in computation and for the overall calculation.
    b) Efficiency: power and space consumption, as well as metrics that relate that consumption to the work completed. Power efficiency is of particular importance to most trading firms. A vendor cannot simply “throw hardware at the benchmark” without suffering in the efficiency metrics.
    c) Quality: metrics specific to each calculation that indicate the usefulness of the calculated results. There are well-known tradeoffs between quality and speed. There are also legitimate reasons for firms to pick different points on that tradeoff curve. STAC-A2 quantifies the quality of the implementation being tested, as described in Section II.
    d) Programming difficulty: how easily a given developer could re-create the implementation. The complexity of writing high-performance code is widely assumed to vary considerably by architecture. And the productivity of risk-analytics developers is crucial to technology buyers, since these developers tend to be highly paid. However, programming difficulty is hard to measure and depends on the context within a given trading firm. STAC-A2 lets trading firms draw their own conclusions by requiring any vendor disclosing results to make the source code of its implementation available within the STAC Benchmark Council (with the exception of code beneath productized interfaces).

V. Example Use of Benchmarks

Intel has created a STAC-A2 implementation for the latest x86 instruction set using components of Intel® Parallel Studio XE[10] such as the Intel® Math Kernel Library and Intel® C++ Composer XE. This implementation makes use of Advanced Vector Extensions (Intel® AVX) on those chipsets able to take advantage of it. Intel® AVX is a 256-bit extension to SSE and helps to improve performance of floating point intensive applications due to wider vectors and rich functionality. Optimization of the Intel x86 implementation is on-going. Results are not official and cannot be fairly compared to other vendors’ results at this stage. However, STAC has authorized release of preliminary results to facilitate discussion of the STAC-A2 specifications as they proceed toward ratification.

While not finalized, the current Intel implementation has proven useful for comparing Intel architectures, as demonstrated in the following sections.

A. Speed Results

Intel compared two systems:

- **System 1**, running Intel® Xeon® X5680, 3.33 GHz, 2 sockets x 6 cores with 24 GB RAM and 12 MB LLC (codenamed Westmere EP).
- **System 2**, running Intel® Xeon® E5-2690, 2.9 GHz, 2 sockets x 8 cores with 64GB RAM and 20 MB LLC (codenamed Sandybridge EP).

Each system had the highest clockspeed available and the most 1333 MHz DRAM that could be accommodated by its processor. Both systems ran Red Hat Enterprise Linux 6.1.

As Table 1 and Figure 1 illustrate, System 2 demonstrated a 33% to 76% speed advantage over System 1. The difference increased with the number of Monte Carlo paths (except at 200,000 paths).

TABLE I. STAC-A2.v0.5.GREEKS.* RESULTS (IN SECONDS) WITH VARYING PATHS (5 ASSETS, 10 TIMESTEPS)

	Number of Paths				
	5K	**10K**	**100K**	**150K**	**200K**
System 1	0.11	0.23	2.75	4.13	5.66
System 2	0.07	0.14	1.62	2.35	3.32

Intel® 64 mode, KMP_AFFINITY=compact, LP64 mode of Intel® MKL. Build options: icl –xSSE4.2 –openmp (System 1), icl –xAVX –openm (System 2).

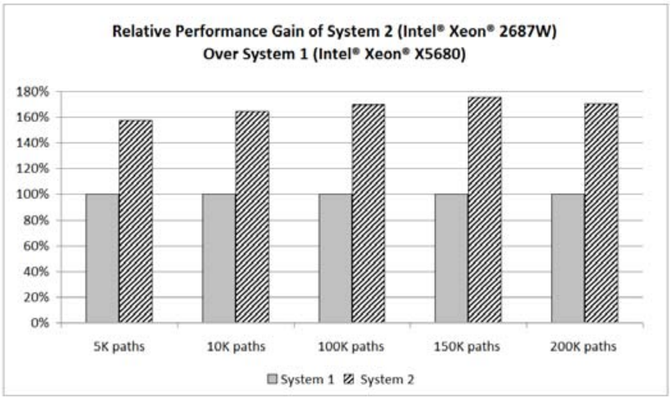

Figure 1 – Relative Speed of Intel Architectures using STAC-A2.v.0.5.GREEKS.*

The improvement in these compute- and memory-intensive benchmarks is due in part to architectural improvements in the CPU, a higher core count, and using Intel® AVX on System 2 (which supports AVX) and Intel® SSE on System 1. The workload fit within 24GB RAM, so the extra memory in System 2 is not believed to have contributed.

B. *Capacity and Efficiency Results*

Table II compares Systems 1 and 2 using capacity and efficiency tests from STAC-A2. The asset-capacity test determines the maximum number of assets for which Greeks can be calculated within a 10-minute window. System 1 was capable of handling 66 assets, while System 2 handled 79. This 20% improvement in asset capacity masks a greater underlying increase in compute capacity, since the workload increases roughly quadratically with the number of assets.

Table II also shows the difference in power efficiency when these systems run at their maximum asset loads. The fact that System 2 completed significantly more work with slightly less power consumption than System 1 translates into an asset-efficiency improvement of 21%. That is, given a fixed power budget, a manager could expect System 2 to process risk for a portfolio 21% larger than on System 1.

TABLE II. COMPARISON OF INTEL ARCHITECTURES USING STAC-A2.V0.5.GREEKS.CAPACITY AND STAC-A2.V0.5.GREEKS.EFFICIENCY

	System 1	**System 2**	**Percentage difference**
Mean Watts consumed	380	375	-1%
Kilojoules consumed	228	225	-1%
ASSET CAPACITY (Assets completed)	66	79	20%
ASSET EFFICIENCY (Assets per kilojoule)	0.29	0.35	21%

10 time steps, 10K paths. Intel(R) Composer XE 2013. Intel® 64 mode, KMP_AFFINITY=compact, LP64 mode of Intel® MKL. Build options: icl –xSSE4.2 –openmp (System 1), icl –xAVX –openm (System 2).

C. *Scaling Results*

The Intel x86 implementation defines scales in terms of threads, in powers of two. Figure 2 is one example of a scale curve, in this case plotting the speed at which a set workload can be executed at six different scales. As expected, additional threads increase the speed until they exceed the number of cores available (in System 1).

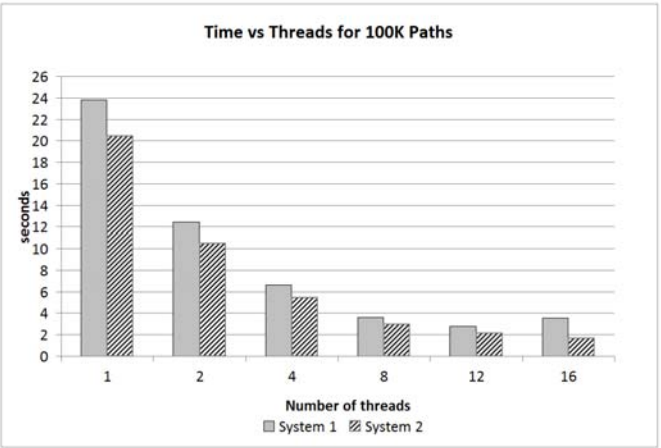

Figure 2 – Speed vs SUT Scale (Threads) for Intel Architectures using STAC-A2.v.0.5.GREEKS.*

D. *Quality Results*

Intel's implementation scores high on the STAC-A2 quality metrics. For example:

- The Anderson-Darling statistic for the output of the unit normal random number generation benchmark converted to a p-value of 0.92, which indicates very high confidence that the generator was, in fact, Gaussian.
- The root-mean-square error between observed and theoretical values for the mean Heston price and variance of Heston price were 0.06% and 0.04%, respectively. Figures 3 and 4 plot the observed and theoretical values.

E. *Programming Difficulty*

In conformance with STAC-A2 requirements, Intel is in the process of submitting the source code of its implementation to the STAC Benchmark Council for inspection by members.

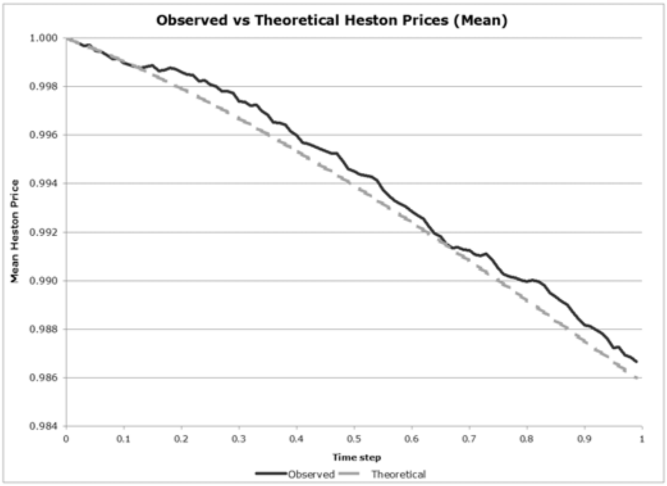

Figure 3 – Observed and Theoretical Mean Heston Prices

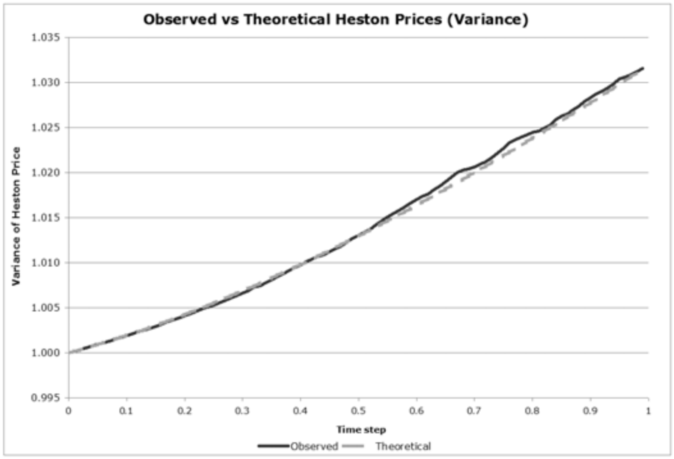

Figure 4 – Observed and Theoretical Variance of Heston Prices

VI. Future Work

The STAC Benchmark Council has three primary next steps it is pursuing with STAC-A2:

1. Finalizing the STAC-A2 Benchmark specifications.
2. Facilitating performance results for as many architectures of interest to trading firms as possible.
3. Enhancing the specifications to allow for different mathematical approaches to computing Greeks. The simple finite difference technique embodied in the component layers of STAC-A2 is just one way to compute Greeks. Other approaches exist, such as Malliavin calculus.[11] An implementation that did not implement some or all of the component operations of STAC-A2 would not be required to report results for them. It would simply report results for the end-to-end Greeks calculations (including the baseline, scale, efficiency, and quality benchmarks). Enabling this flexibility will allow comparisons not only of different technology stacks but also of different analytic approaches. If possible, this flexibility will be added to STAC-A2 before finalizing version 1.0.

Acknowledgments

The authors wish to acknowledge John Ashley of Nvidia and the late Lurng-Kuo Liu of IBM for their significant contributions to the detailed specification of STAC-A2. They would also like to recognize the valuable contributions of several Intel employees to the Intel implementation: Sergey Maidanov, Ilya Burylov, Rick Carlin, and David O’Shea.

References

[1] See www.STACresearch.com/domains.

[2] www.STACresearch.com/a2.

[3] Heston., Steven L., “A closed-form solution for options with stochastic volatility with applications to bond and currency options,” The Review of Financial Studies, Volume 6, Issue 2, 327-343 (1993).

[4] 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).

[5] 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).

[6] Andersen, Leif B.G., “Efficient Simulation of the Heston Stochastic Volatility Model,” SSRN working paper: ssrn.com/abstract=946405 (January 23, 2007).

[7] Longstaff , F.A. and Schwartz, E.S., "Valuing American Options by Simulation: A Simple Least-Squares Approach," Review of Financial Studies, 14 (1), pp.113 – 147 (2001).

[8] Cox , John C., Ross, Stephen A., and Rubinstein, Mark, "Option Pricing: A Simplified Approach," Journal of Financial Economics 7: 229-263 (1979).

[9] Margrabe, William, “The Value of an Option to Exchange One Asset for Another,” Journal of Finance, 33:177–186 (1978).

[10] http://software.intel.com/en-us/intel-parallel-studio-xe

[11] Fournié, Eric, et al, “ Applications of Malliavin calculus to Monte Carlo methods in finance,” Finance and Statistics, 3, 391–412 (1999).