

# The HPEC Challenge Benchmark Suite

Ryan Haney, Theresa Meuse, Jeremy Kepner and James Lebak

{haney,tmeuse,kepner,jlebak}@ll.mit.edu

MIT Lincoln Laboratory, 244 Wood Street, Lexington, MA 02420

## Abstract

Quantitative evaluation of different multi-processor High Performance Embedded Computing (HPEC) systems is an ongoing challenge for the HPEC community. The DARPA Polymorphous Computer Architecture (PCA) and High-Productivity Computing Systems (HPCS) programs have created kernel and system level benchmarks and metrics for comparing the different architectures being developed under these programs. In this talk, we will describe a new benchmark suite drawn from the HPCS and PCA programs: the HPEC Challenge Benchmarks. It consists of eight single-processor kernel benchmarks and of a multi-processor scalable synthetic SAR benchmark. We describe an implementation of the kernel benchmarks on the PowerPC G4 and the metrics used to evaluate it. We also demonstrate the parallel SAR benchmark and its scaling to multiple problem and machine sizes. The HPEC Challenge suite will be made widely available to community and will enable more rigorous comparison of HPEC systems.

## Kernel Benchmarks

The single-processor kernel benchmarks are drawn from a survey of several broad DoD signal processing application areas, including radar and sonar processing, infrared sensing, hyper-spectral imaging, signal intelligence, communication, and data fusion. From these applications we distilled a set of eight kernel benchmarks and data sets that are representative of the computing needs of these applications. These kernels are drawn both from “front-end” signal processing systems that operate in a data-independent fashion, as well as “back-end” information and knowledge processing systems that operate in a data-dependent fashion. The signal processing kernels are finite impulse response (FIR) filtering, QR factorization (QR), singular value decomposition (SVD), and constant false-alarm rate detection (CFAR). The information and knowledge processing kernels are pattern matching (PM), graph optimization via genetic algorithm (GA), and real-time database operation (DB). The final kernel is a communication benchmark consisting of a memory rearrangement or corner turn (CT) of a data matrix. We described the kernels and their associated data sets in an MIT/LL project report [2]. As part of the DARPA PCA program, we evaluated these benchmarks on several processors, including the PowerPC G4 (see Figure 1). Important metrics for evaluating these kernels include traditional metrics of throughput, latency, and power efficiency, as well as stability of performance across

different data sizes and kernels (for a definition of stability see Kuck [1]).

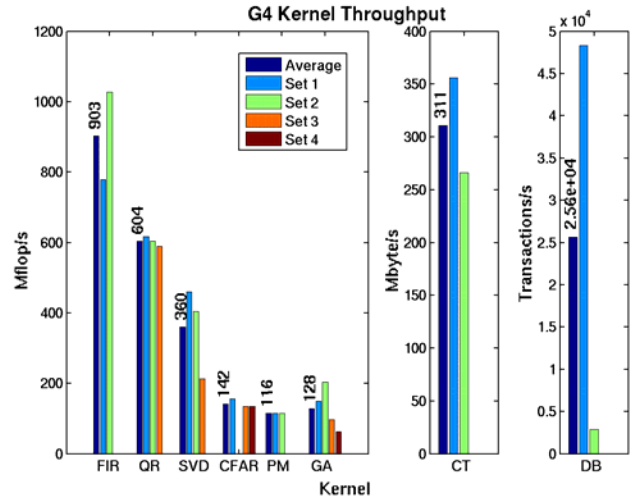


Figure 1. Performance of the 500 MHz PowerPC 7410 on the kernel benchmarks [3].

## SAR System Benchmark

The HPCS Scalable Synthetic Compact Application #3 (SSCA #3) simulates a sensor processing chain (Figure 2). It consists of a front-end sensor processing stage, where Synthetic Aperture Radar (SAR) images are formed, and a back-end knowledge formation stage, where detection is performed on the difference of the SAR images. It generates its own synthetic ‘raw’ data, which is scalable. The goal is to mimic the most taxing computation and I/O requirements found in many embedded systems, such as medical/space imaging, or reconnaissance monitoring. Its principal performance goal is throughput, in other words, to maximize the rate at which answers are generated. The computational kernels must keep up with copious quantities of sensor data. Its I/O kernels must manage both streaming data storage, as well as file sequences retrieval.

The Scalable Data Generator (SDG) creates and stores simulated ‘raw’ SAR complex returns. It also generates and stores templates of rotated and pixelated letters.

The Sensor Processing Stage loops until the specified number of desired images has been reached. In this Stage, after reading the ‘raw’ SAR data, Kernel 1 forms a SAR image using a matched filtering and interpolation [4] method. 2D Fourier matched filtering and interpolation involves matched filtering the 2D Fourier transformed returns against the transmitted SAR waveform. Then the results are re-sampled, or interpolated, to go from a polar coordinate representation to a rectangular coordinate representation. A final inverse Fourier transform converts the results into the spatial-domain, where the SAR image

This work is sponsored by the Defense Advanced Research Projects Agency under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

becomes visibly discernible. After Kernel 1, the pixelated templates are inserted at random locations of the SAR image. Kernel 2 stores each ‘populated’ image in a streaming I/O fashion onto a grid of random image locations.

The Knowledge Formation Stage loops until the specified number of desired image sequences has been reached. Kernel 3 randomly picks a given image sequence to read, which is read through its entire grid depth. Kernel 4 compute the differences between each pair of consecutive images, and thresholds the difference image to identify locations to produce a set of changed pixels. A sub-image is formed around each group of changed pixels which is then convolved with all the letter templates. The template that produces the strongest match is then selected as the identity for the particular sub-image.

Verification of the benchmark occurs by comparing the location at the identity of each found letter and comparing with what was inserted. The input data is constructed so that all the pixelated letters should be found with no false alarms.

The SAR system benchmark can be operated in one of three modes: System Mode (which includes both its computational kernels and I/O kernels), Compute Mode (which includes its computational kernels while bypassing its I/O kernels), and File I/O Mode (which includes its I/O kernels while bypassing its computational kernels). Each kernel’s operation is timed.

The performance of Compute Mode corresponds to the traditional focus of the HPEC community. The System Mode can be used to measure both compute and storage I/O throughput, which is becoming increasingly important in HPEC systems.

The benchmark has both a serial and a parallel implementation. All the kernels and the I/O are designed so as to be run on parallel computing and parallel storage systems.

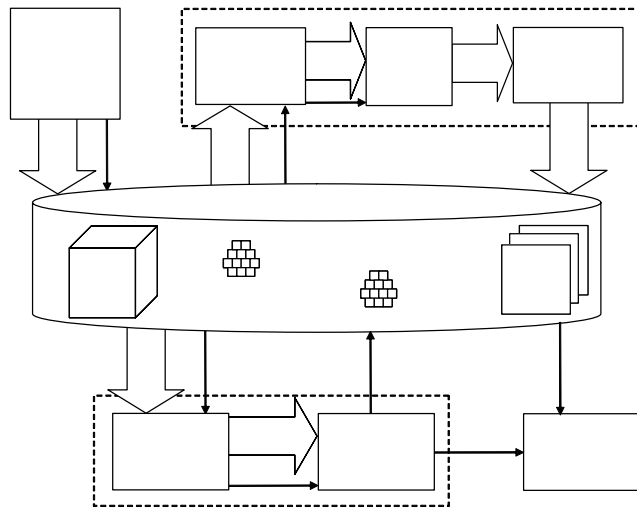


Figure 2. Block diagram of SAR system benchmark.

## Summary

We have developed a set of eight kernels and a scalable SAR system benchmarks for quantitatively comparing HPEC systems. The kernels address important operations across a broad range of DoD signal and image processing applications. The scalable SAR system benchmark is representative of one of the most common functions in DoD surveillance systems. In addition, it includes storage I/O components found in a broad class of applications. The HPEC Challenge Benchmark Suite will provide the community with a valuable tool for objectively evaluating systems and the potential impact of new technologies.

## References

- [1] David J. Kuck. *High Performance Computing: Challenges for Future Systems*. Oxford University Press, New York, NY, 1996.
- [2] James Lebak, Albert Reuther, and Edmund Wong. Polymorphous Computing Architecture (PCA) Kernel-level Benchmarks. Project Report PCA-KERNEL-1, MIT Lincoln Laboratory, Lexington, MA, January 2004.
- [3] James Lebak, Hector Chan, Ryan Haney, and Edmund Wong. Polymorphous Computing Architecture (PCA) Kernel Benchmark Measurements on the PowerPC G4. Project Report PCA-KERNEL-2, MIT Lincoln Laboratory, Lexington, MA, January 2004.
- [4] Soumekh, Mehrdad. *Synthetic Aperture Radar Signal Processing with Matlab Algorithms*. Wiley, New York, NY, 1999.