The HPEC Challenge Benchmark Suite

Ryan Haney, Jeremy Kepner, and Theresa Meuse

Massachusetts Institute of Technology
Lincoln Laboratory

HPEC 2006

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.

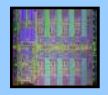
Motivation

Advanced Sensor Platforms

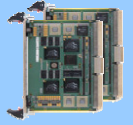


Processor and System Architectures

Single Processor Element



Tiled Processors



Multi-computers

System Analysis and Design

Implement Benchmarks

- Design
- Code
- Tune

Measure Performance

- Throughput
- Power
- Stability

Design System

- Choose components
- Hardware size
- Required software performance

HPEC Challenge Benchmark Suite

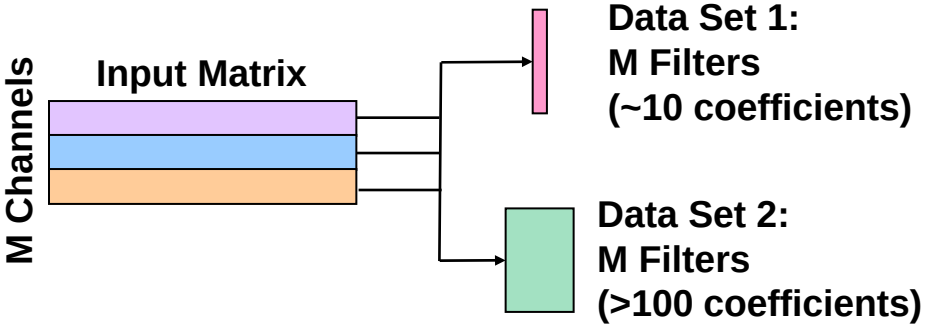
- Kernel Benchmarks
  - Single-processor operations
  - Taken from PCA program
- SAR Application Benchmark
  - Multi-processor compact application
  - Taken from HPCS program

Outline

- Motivation
- Benchmark Suite
  - Kernel Benchmarks
  - SAR Application Benchmark
- Web Site

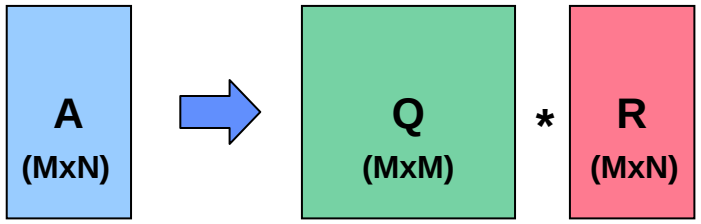
Signal and Image Processing Kernels

FIR



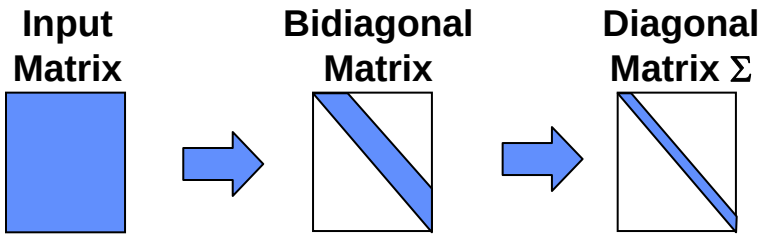
- Bank of filters applied to input data
- FIR filters implemented in time and frequency domain

QR



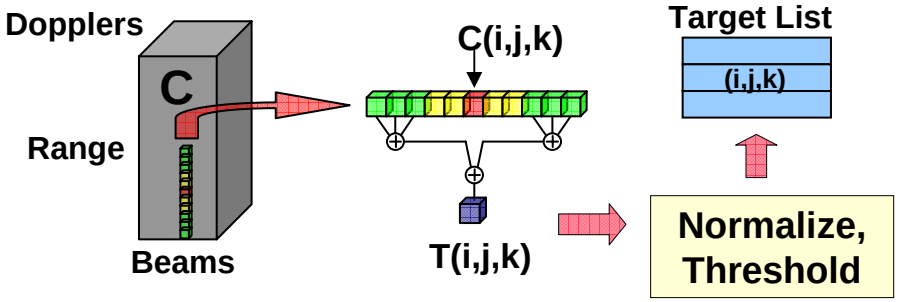
- Computes the factorization of an input matrix, A=QR
- Implementation uses Fast Givens algorithm

SVD



- Produces decomposition of an input matrix, $X=U\Sigma V^H$
- Classic Golub-Kahan SVD implementation

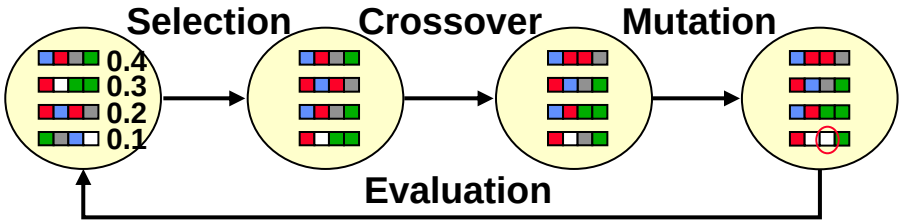
CFAR



- Creates a target list given a data cube
- Calculates normalized power for each cell, thresholds for target detection

Information and Knowledge Processing Kernels

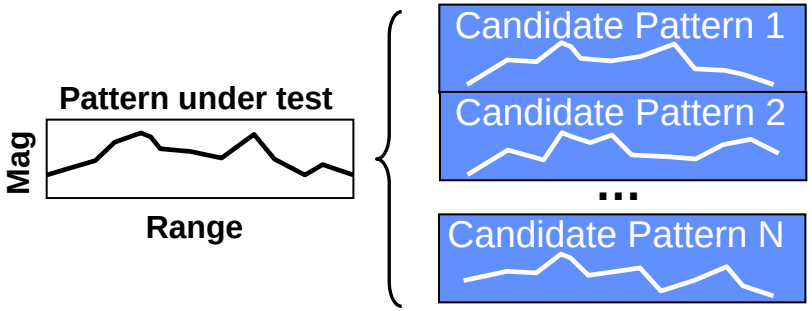
Genetic Algorithm



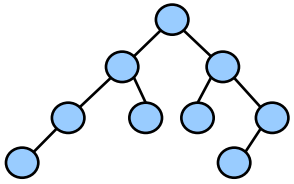
- Evaluate each chromosome
- Select chromosomes for next generation
- Crossover: randomly pair up chromosomes and exchange portions
- Mutation: randomly change each chromosome

Pattern Match

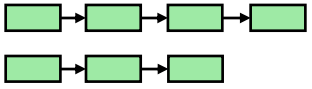
- Compute best match for a pattern out of set of candidate patterns
  - Uses weighted mean-square error



Database Operations



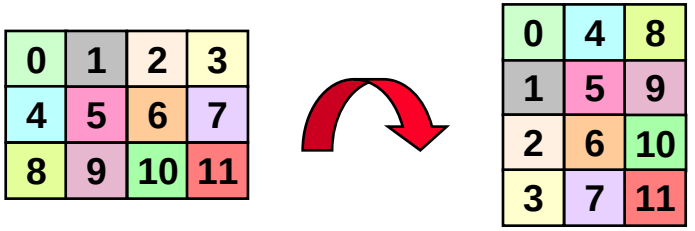
Red-Black Tree Data Structure



Linked List Data Structures

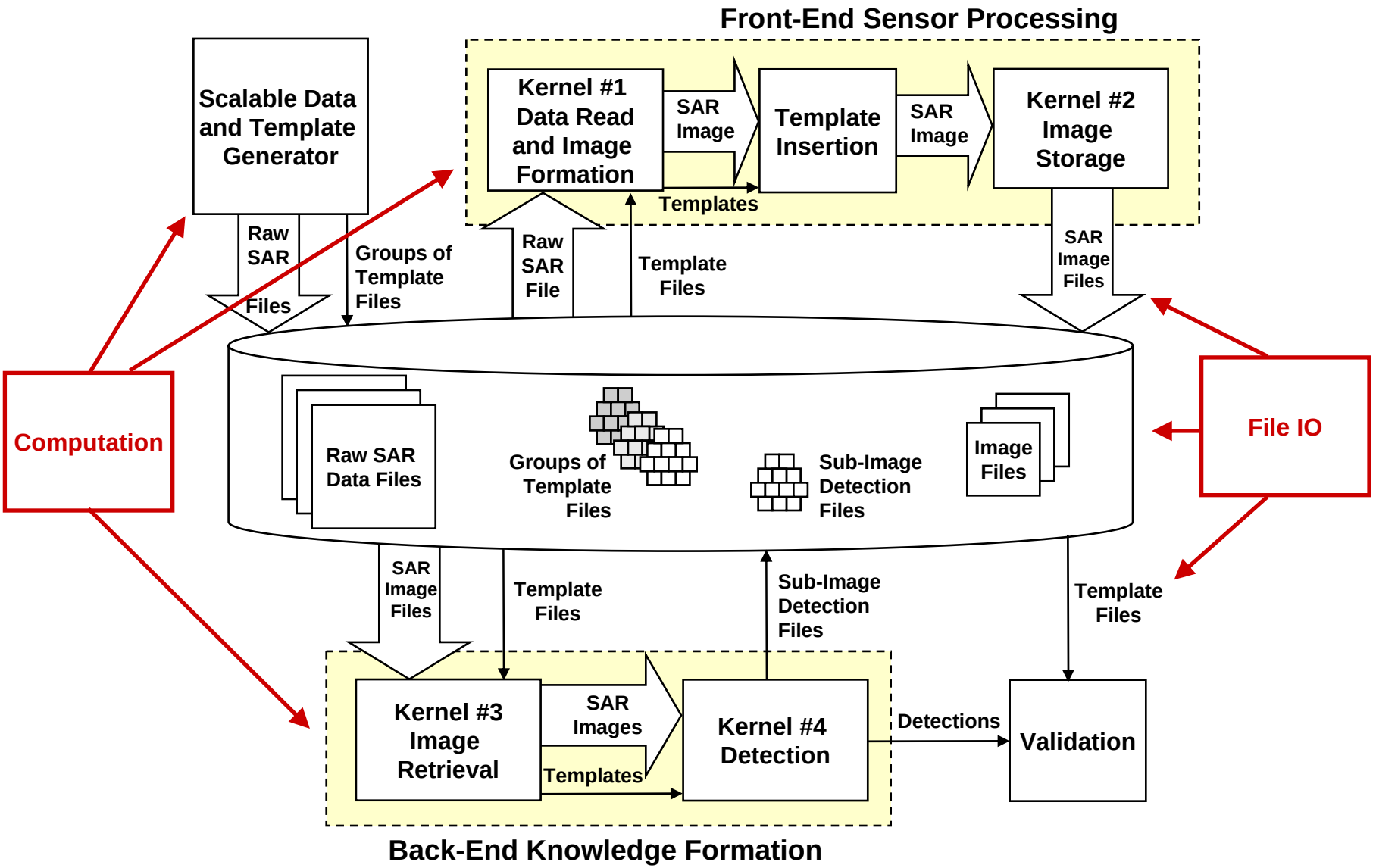
- Three generic database operations:
  - search: find all items in a given range
  - insert: add items to the database
  - delete:remove item from the database

Corner Turn

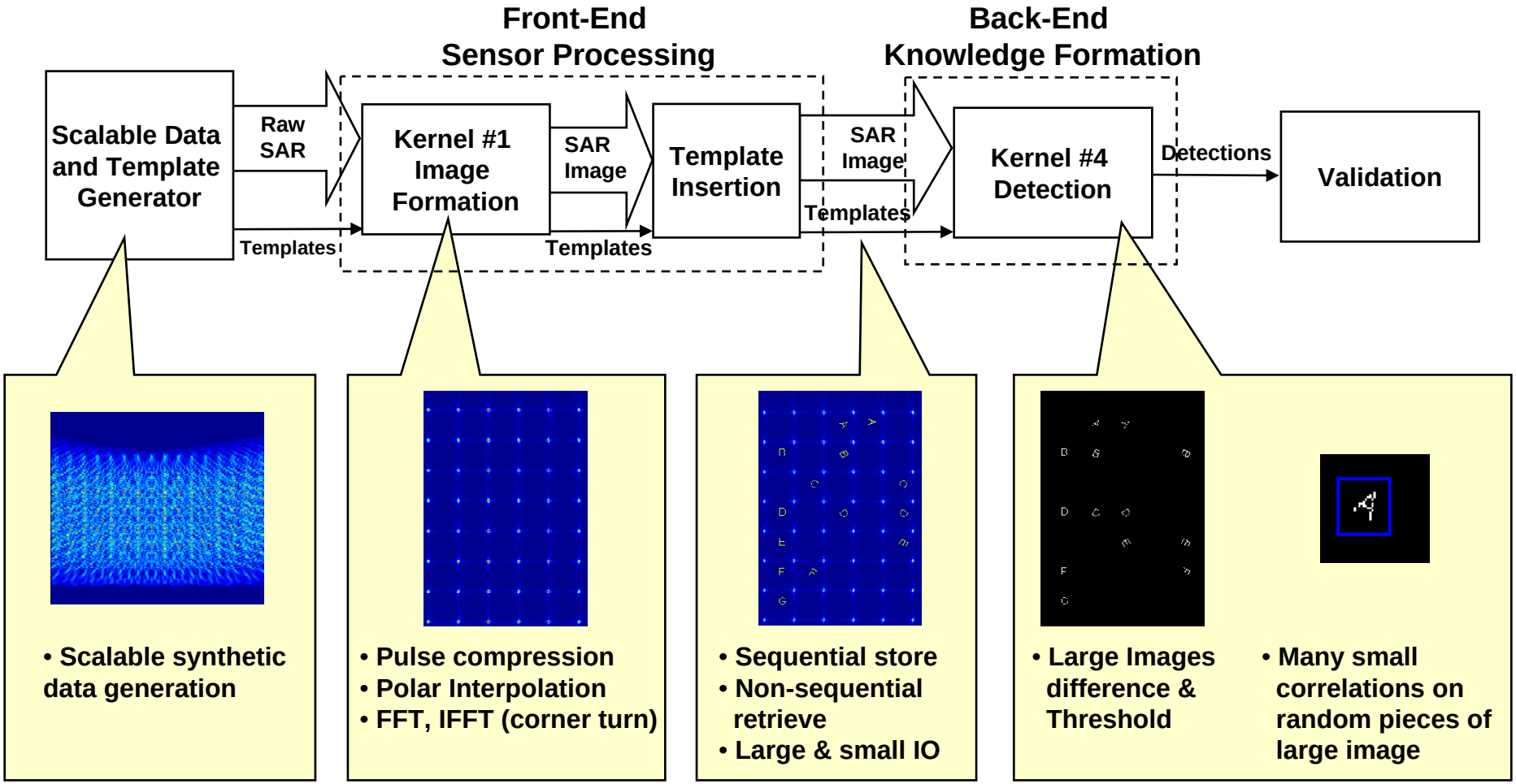


- Memory rearrangement of matrix contents
  - Switch from row to column major layout

SAR System Architecture



SAR Benchmark Computational Challenges and Data Products

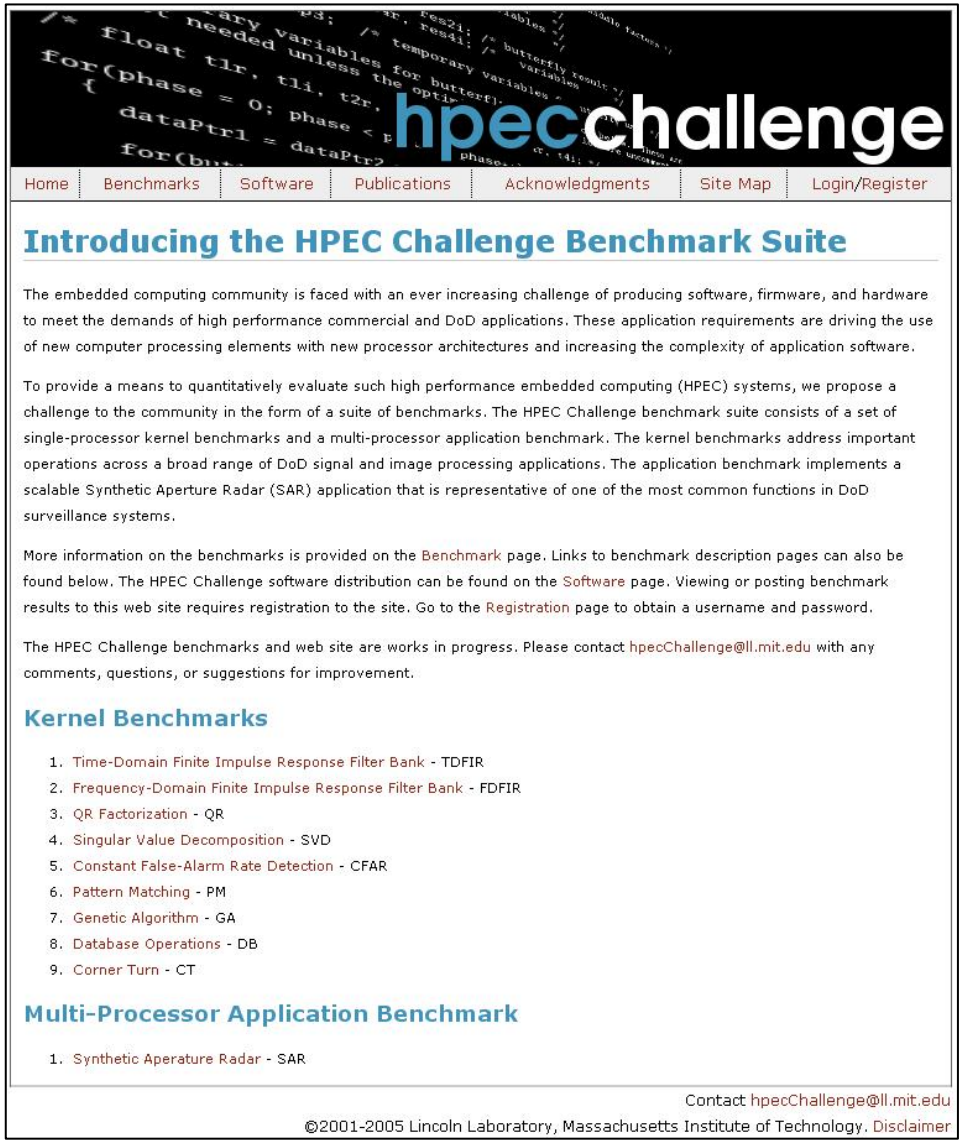


Outline

- Motivation
- Benchmark Suite
  - Kernel Benchmarks
  - SAR Application Benchmark
- Web Site

HPEC Challenge Web Site: http://www.ll.mit.edu/HPECchallenge

- **Purpose:**
  - **Allow exploration of performance results for a wide range of architectures**
- **Provide public and password protected pages**
  - **Secure potentially sensitive architecture results**
- **Public pages:**
  - **Benchmark information**
  - **Benchmark software**
  - **Periodic releases of benchmark results**



http://www.ll.mit.edu/HPECchallenge

Benchmark Information

Benchmarks

The HPEC Challenge benchmark suite consists of a set of eight kernel benchmarks and a s
(SAR) system benchmark for quantitatively comparing HPEC systems. The kernel benchma
across a broad range of DoD signal and image processing applications. The scalable SAR s
one of the most common functions in DoD surveillance systems. In addition, it includes sto
class of applications.

Metrics of interest for the benchmarks are defined on the metrics page.

Kernel Benchmarks

1. Time-Domain Finite Impulse Response Filter Bank - TDFIR
2. Frequency-Domain Finite Impulse Response Filter Bank - FDFIR
3. QR factorization - QR
4. Singular Value Decomposition - SVD
5. Constant False-Alarm Rate Detection - CFAR
6. Pattern Matching - PM
7. Genetic Algorithm - GA
8. Database Operations - DB
9. Corner Turn - CT

Multi-Processor Application Benchmark

1. Synthetic Aperature Radar - SAR

Contact hpecC

Time-Domain Finite Impulse Response Filter Bank

The finite impulse response (FIR) filter is one of the basic operations in signal processing. This kernel implements a set of M FIR filters. Each FIR filter m, m ∈ {0, 1, ..., M-1 }, has a set of impulse response coefficients $w_m[k]$, k ∈ { 0 ... K-1 }. If the length of the input vector is N, the output of filter m, y_m, is the convolution of w_m with the input x_m:

$$y_m[i] = \sum_{k=0}^{K-1} x_m[i-k]w_m[k], \text{for } i = 0, 1, \ldots N-1.$$

We define two datasets in Table 1, one for a short filter and one for a long filter.

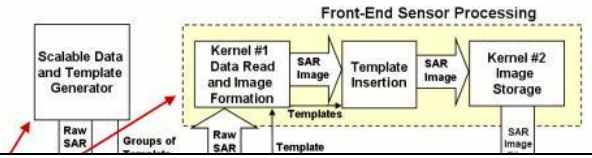
Table 1: FIR filter bank input parameters.

Parameter Name	Description	Values	
		Set 1	Set 2

An operation co

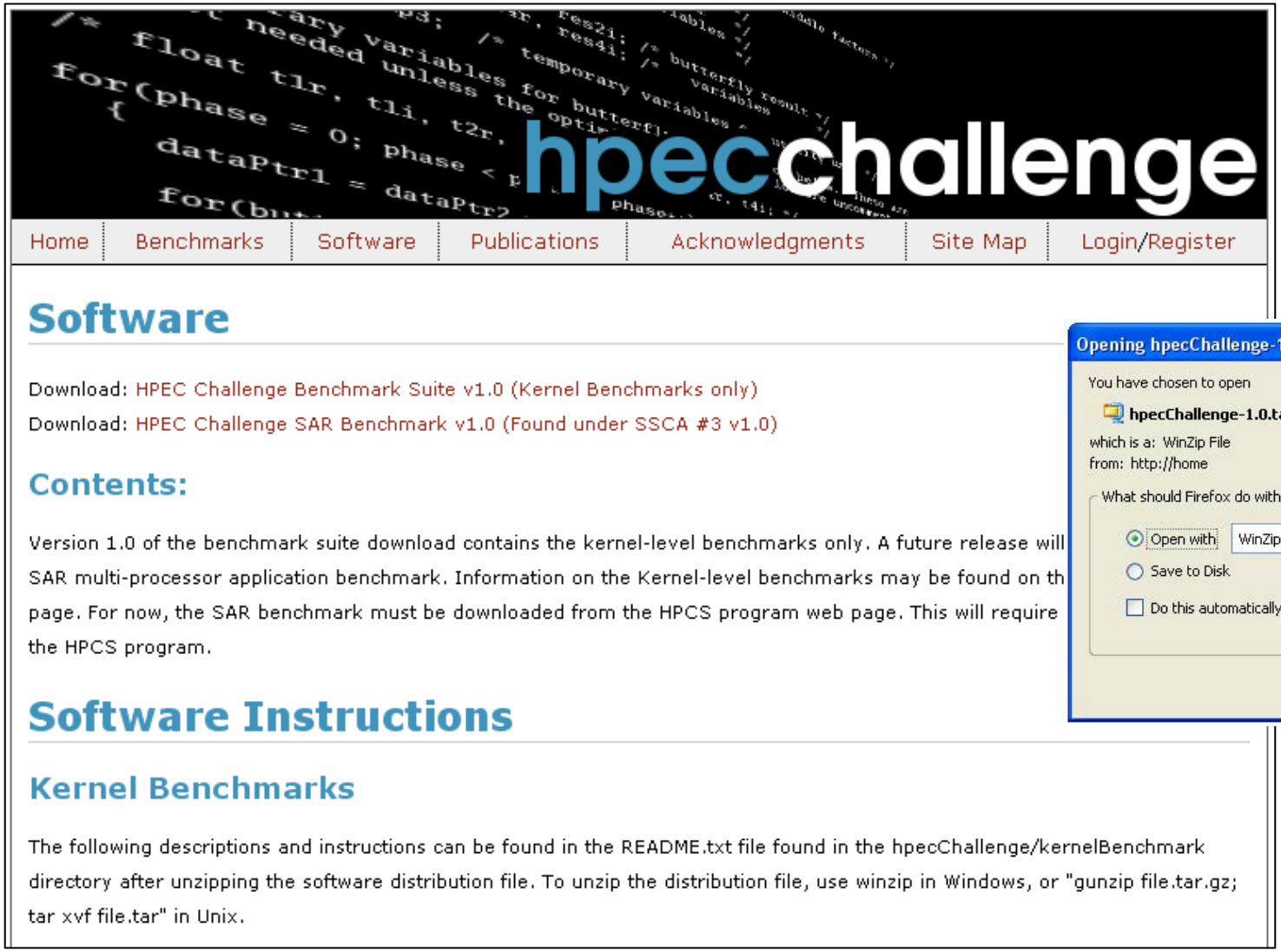
SAR Benchmark

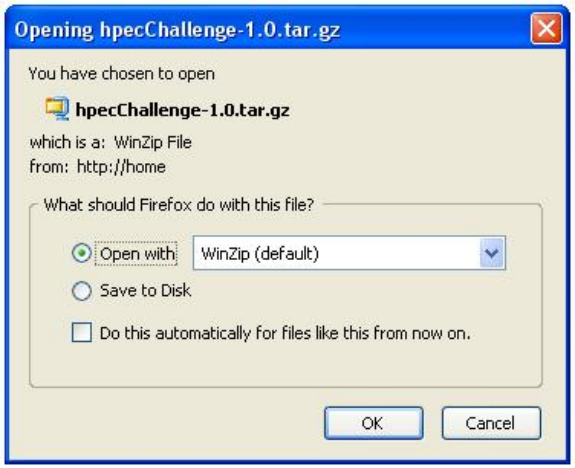
The HPCS Scalable Synthetic Compact Application #3 (SSCA #3) simulates a sensor processing chain (Figure 1). It consists of a front-end sensor processing stage, where Synthetic Aperture Radar (SAR) images are formed, and a back-end knowledge formation stage, where detection is performed on the difference 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 sequential file retrieval.



http://www.ll.mit.edu/HPECchallenge

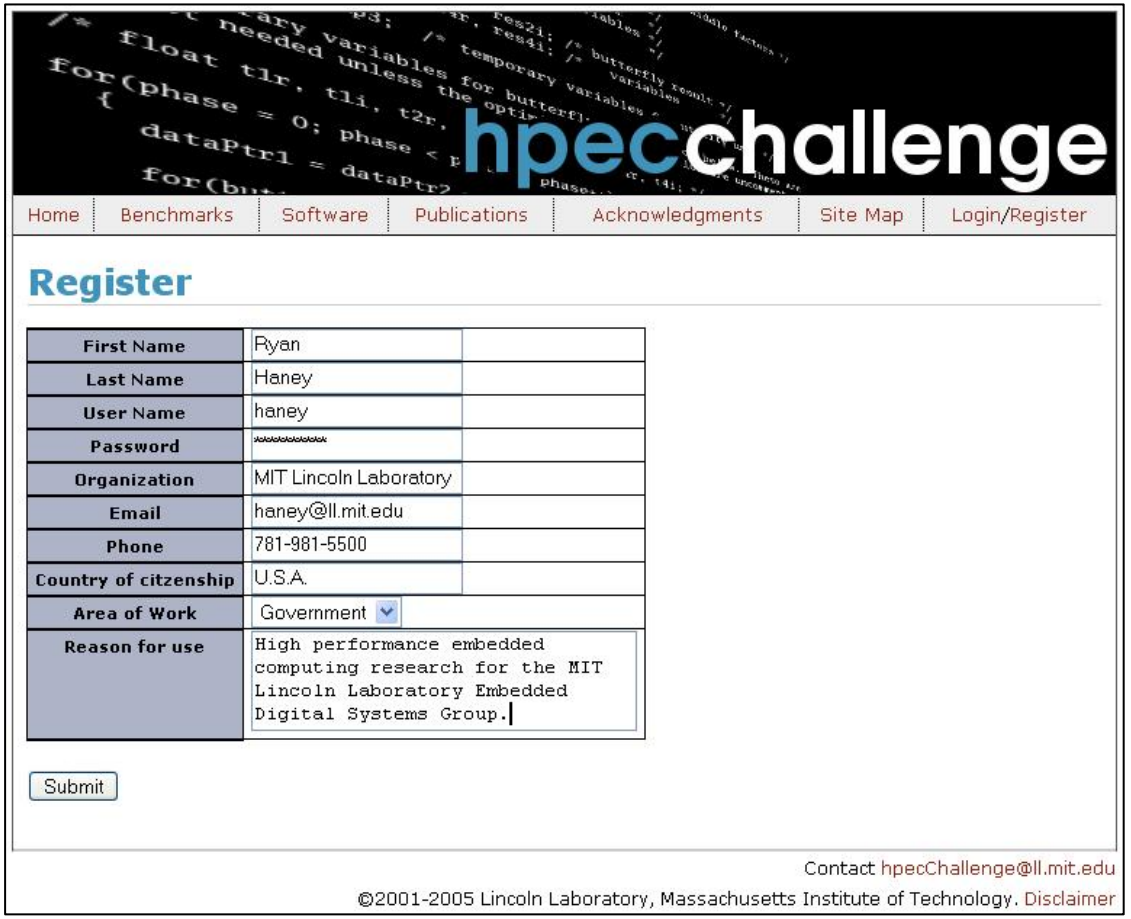
Software Distribution



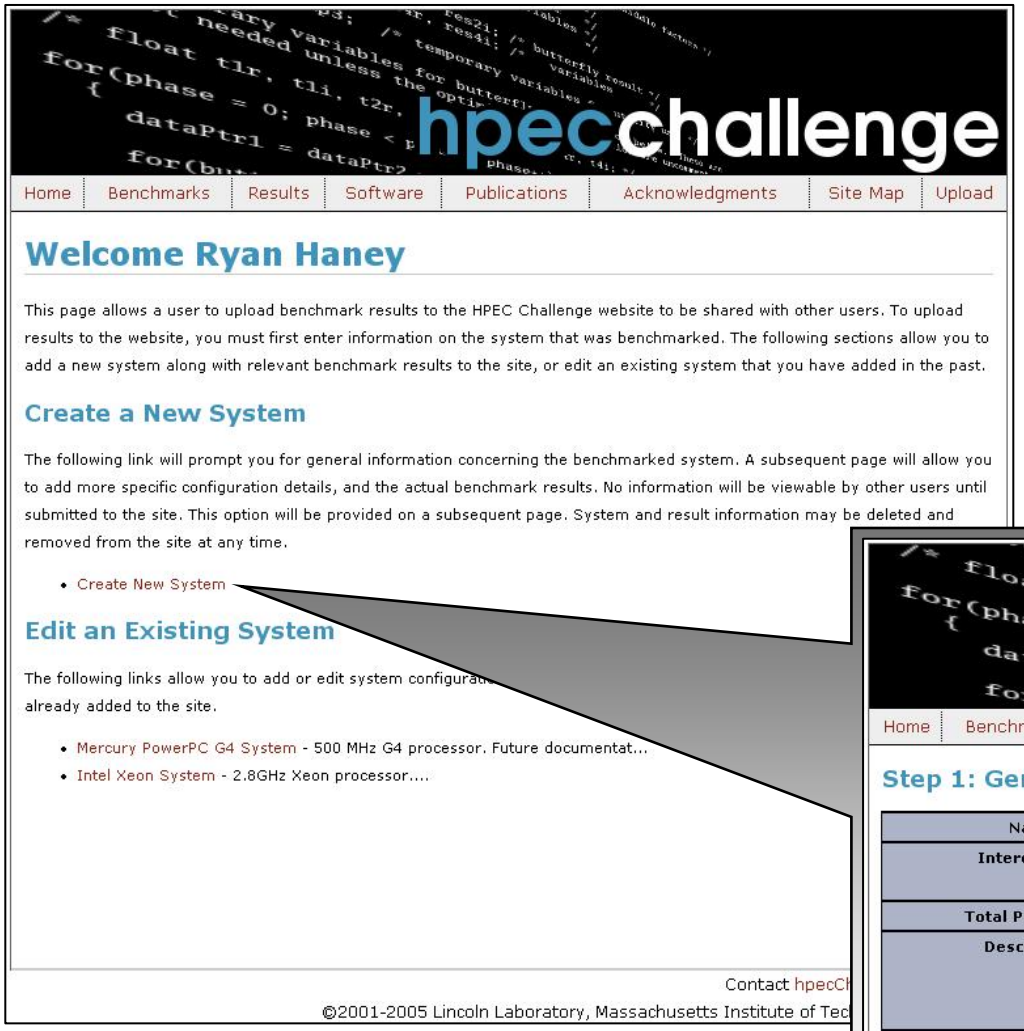


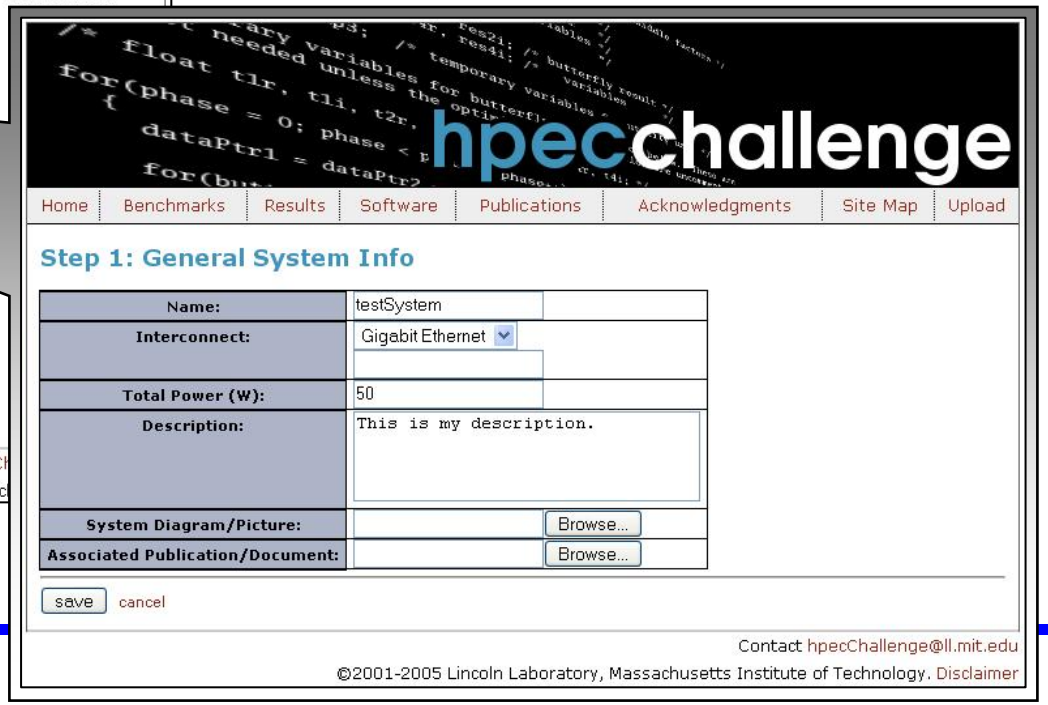
http://www.ll.mit.edu/HPECchallenge
Account Registration

- **Registration**
  - **Obtain username, access to password protected pages**
- **Password protected pages:**
  - **Upload results to site**
  - **Architecture results, reports**



http://www.ll.mit.edu/HPECchallenge
Uploading Results



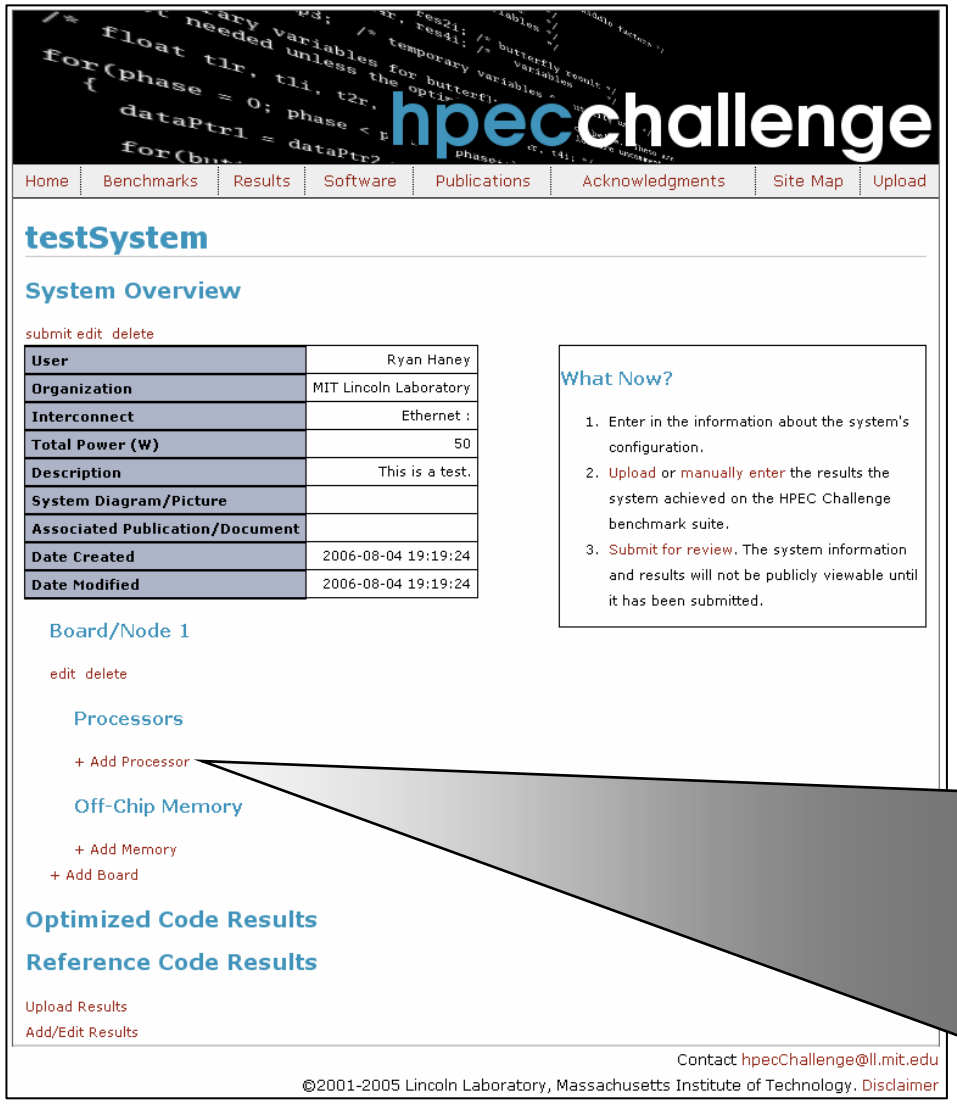


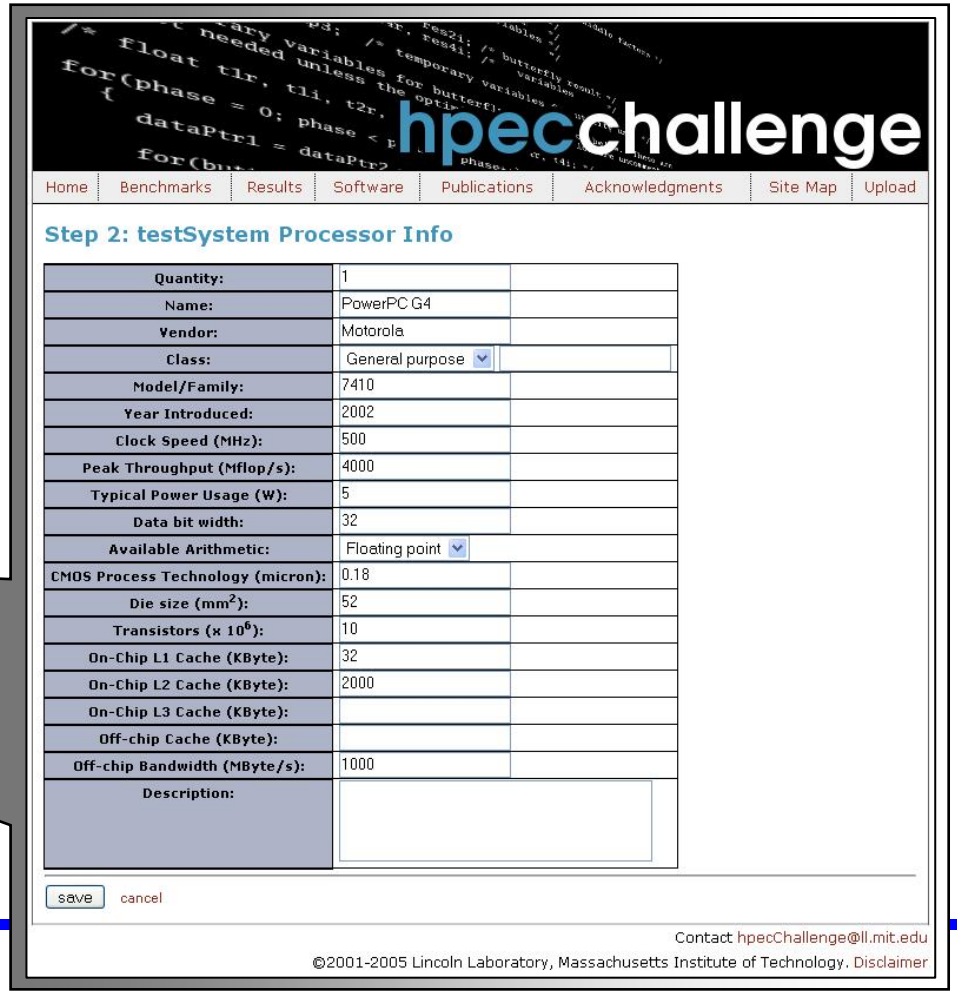
- **Step 1**
  - **Create a new system**
  - **i.e. Describe the system you benchmarked**

http://www.ll.mit.edu/HPECchallenge

Uploading Results

- **Step 2**
  - **Fill in more detailed system specifications**

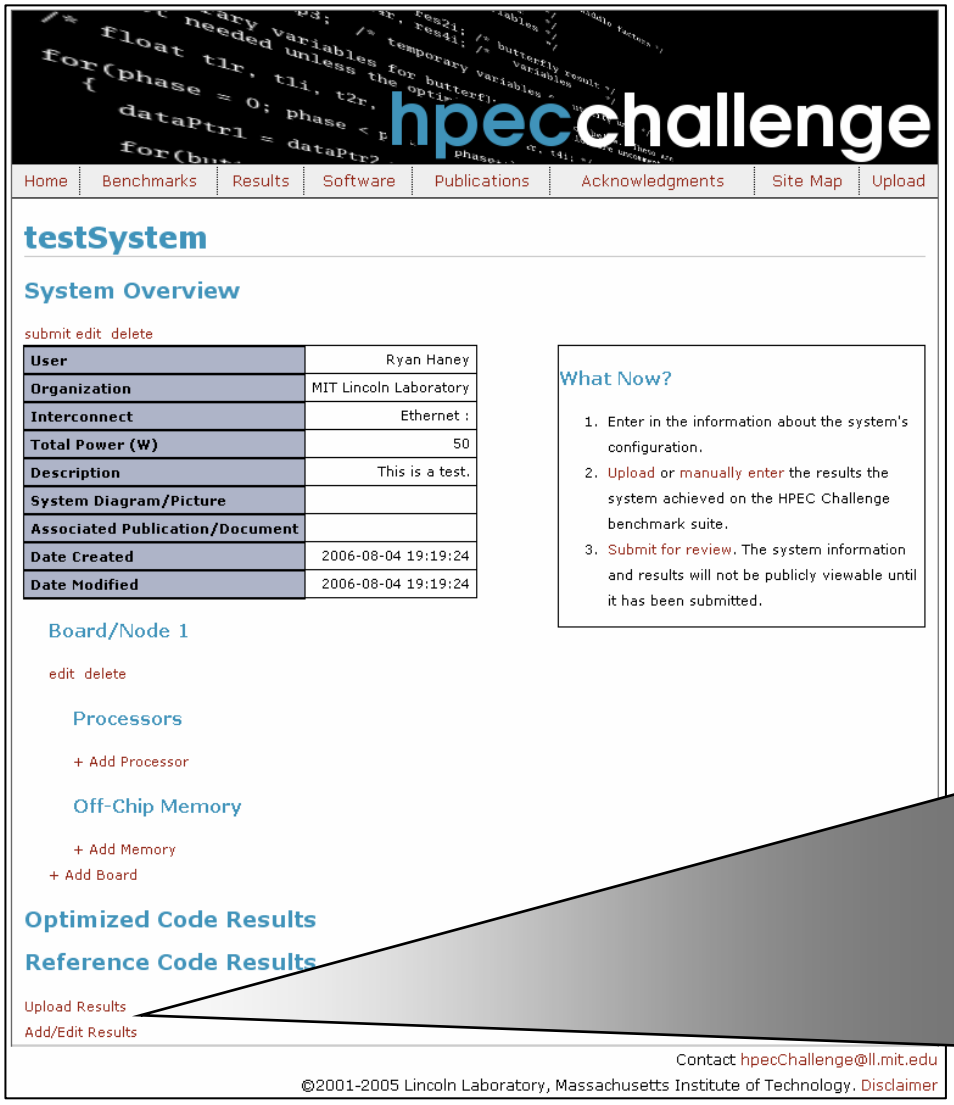


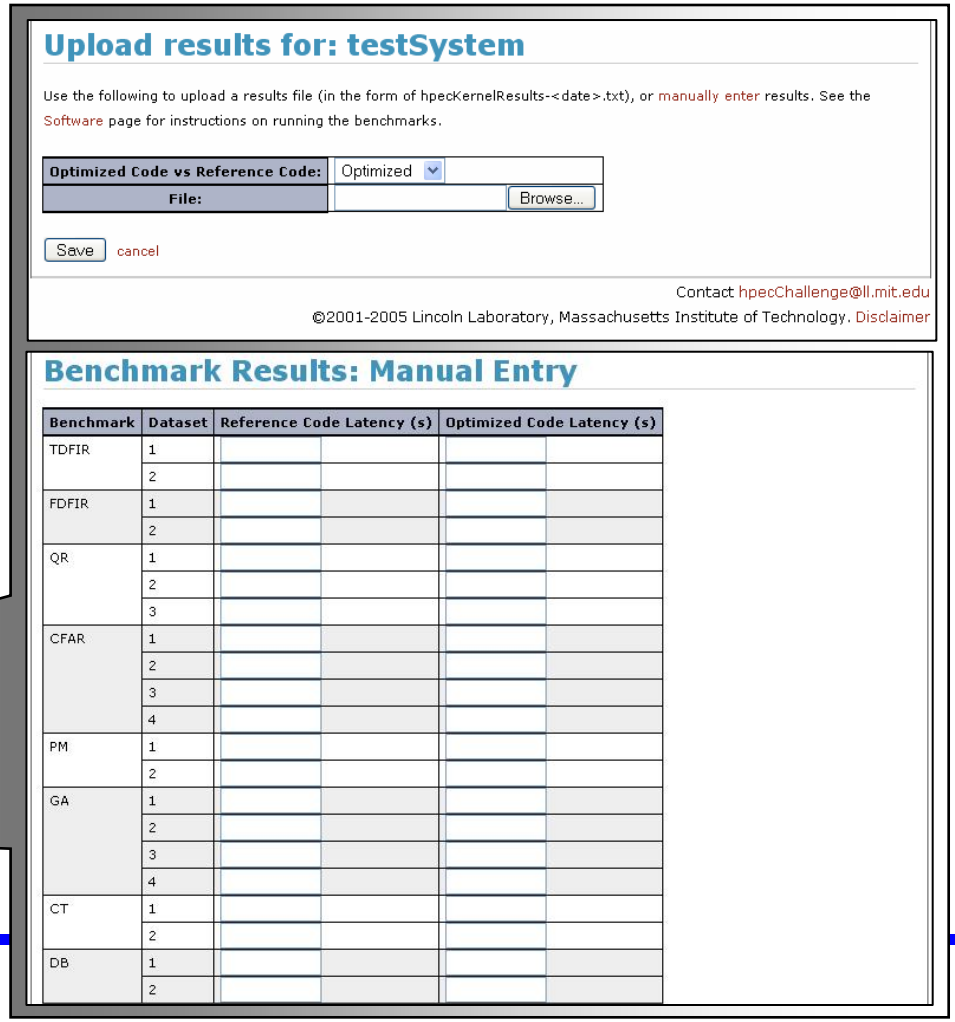


http://www.ll.mit.edu/HPECchallenge

Uploading Results

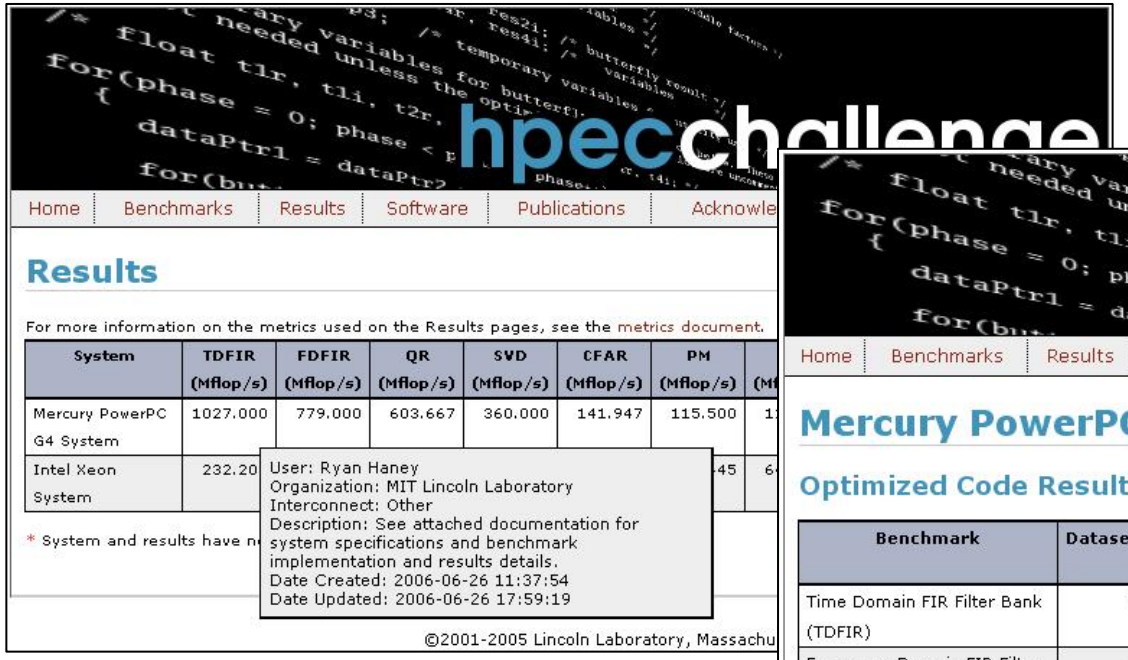
- **Step 3**
  - **Upload results from benchmark results file, or manually fill in**

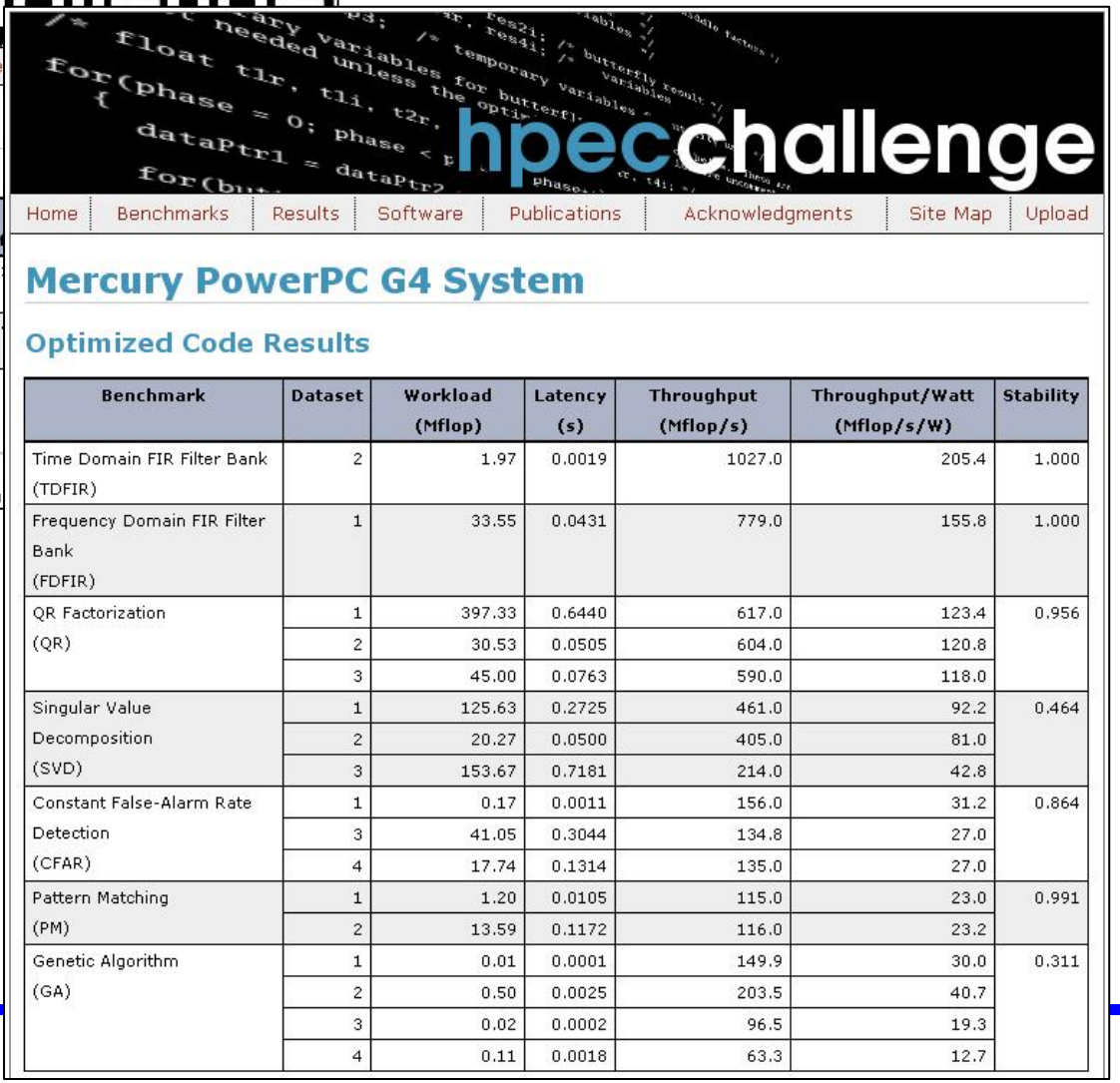




http://www.ll.mit.edu/HPECchallenge

Viewing Results





- **General comparisons between architectures**
- **Detailed results and specifications for architectures**

Summary

- The HPEC Challenge is a publicly available suite of benchmarks for the embedded space
  - Representative of a wide variety of DoD applications
- Benchmarks stress computation, communication and I/O
- Benchmarks are provided at multiple levels
  - Kernel: small enough to easily understand and optimize
  - Compact application: representative of real workloads
  - Single-processor and multi-processor
- See http://www.ll.mit.edu/HPECchallenge/ for benchmark documentation and software
- Please send feedback to hpecChallenge@ll.mit.edu