The HPEC Challenge Benchmark Suite

Ryan Haney, Theresa Meuse, Jeremy Kepner and James Lebak

Massachusetts Institute of Technology
Lincoln Laboratory

HPEC 2005

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.
Outline

• Introduction
• Kernel Level Benchmarks
• SAR Benchmark
• Release Information
• Summary
Acknowledgements

• Lincoln Laboratory PCA Team
  – Matthew Alexander
  – Jeanette Baran-Gale
  – Hector Chan
  – Edmund Wong

• Shomo Tech Systems
  – Marti Bancroft

• Silicon Graphics Incorporated
  – William Harrod

• Sponsor
  – Robert Graybill, DARPA PCA and HPCS Programs

Motivation

Advanced Sensor Platforms

Processor and System Architectures

Single Processor Element
Tiled Processors
Multi-computers
Super-computers

System Analysis and Design

Implement Benchmarks
- Design
- Code
- Tune

Measure Performance
- Throughput
- Power
- Stability

Design System
- Choose components
- Hardware size
- Required software performance

Challenge: Provide benchmarks that test a system at the kernel and multi-processor levels
HPEC Challenge Benchmark Suite

• PCA program kernel benchmarks
  – Single-processor operations
  – Drawn from many different DoD applications
  – Represent both “front-end” signal processing and “back-end” knowledge processing

• HPCS program Synthetic SAR benchmark
  – Multi-processor compact application
  – Representative of a real application workload
  – Designed to be easily scalable and verifiable
Outline

• Introduction
• Kernel Level Benchmarks
  – Kernel Overview
  – Kernel Architecture
  – Baseline vs. Optimized Results
• SAR Benchmark
• Release Information
• Summary
Kernel Benchmark Selection

Broad Processing Categories

“Front-end Processing”
- Data independent, stream-oriented
- Signal processing, image processing, high-speed network communication

“Back-end Processing”
- Data dependent, thread oriented
- Information processing, knowledge processing

Specific Kernels

Signal/Image Processing
- Finite Impulse Response Filter (FIR)
- QR Factorization (QR)
- Singular Value Decomposition (SVD)
- Constant False Alarm Rate Detection (CFAR)

Communication
- Corner Turn (CT)

Information/Knowledge Processing
- Graph Optimization via Genetic Algorithm (GA)
- Pattern Match (PM)
- Real-time Database Operations (DB)
Signal and Image Processing Kernels

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

**Data Set 1:**
- M Filters (~10 coefficients)

**Data Set 2:**
- M Filters (>100 coefficients)

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

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

**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

Selection  Crossover  Mutation

Evaluation

• 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

Pattern under test

Pattern Match Diagram

Candidate Pattern 1
Candidate Pattern 2
...
Candidate Pattern N

Database Operations

• 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

Corner Turn Diagram

Red-Black Tree
Data Structure

Linked List
Data Structures

0 1 2 3
4 5 6 7
8 9 10 11
Kernel Benchmark Architecture

- Kernels written in ANSI C
  - Portable across UNIX based platforms
- Sample data sets provided
- Generation of user defined data sets and sizes possible using Matlab
Baseline vs. Optimized Kernel Benchmark Results

Optimized code outperforms baseline in FIR due to use of VSIPL libraries.

QR and SVD optimized code outperforms baseline because of AltiVec use.

Results similar for CFAR, PM, and GA. Minimal optimizations were made.

Optimized Corner Turn code uses AltiVec intrinsics.

Optimized DB code uses specialized memory management routines.
Outline

- Introduction
- Kernel Level Benchmarks
- SAR Benchmark
  - Overview
  - System Architecture
  - Computational Components
- Release Information
- Summary
Spotlight SAR System

- Principal performance goal: Throughput
  - Maximize rate of results
  - Overlapped IO and computing

- Intent of Compact App:
  - Scalable
  - High Compute Fidelity
  - Low Physical Fidelity
  - Self-Verifying
HPEC community has traditionally focused on Computation ... but File IO performance is increasingly important.
Data Generation and Computational Stages

Sensor Processing

Kernel #1 Image Formation

Kernel #2 Image Storage

Template Insertion

Kernel #3 Image Retrieval

Kernel #4 Detection

Validation

Scalable Data and Template Generator

Raw SAR

Templates

Groups of Template Files

Raw SAR File

Template Files

SAR Image Files

Image Files

Raw SAR Data Files

Groups of Template Files

Sub-Image Detection Files

Detection Files

Template Files

SAR Image

Templates

Detections

Data Generation and Computational Stages

Kernel #1 Image Formation

Kernel #2 Image Storage

Template Insertion

Kernel #3 Image Retrieval

Kernel #4 Detection

Validation
• Radar captures echo returns from a ‘swath’ on the ground

• Notional linear FM chirp pulse train, plus two ideally non-overlapping echoes returned from different positions on the swath

• Summation and scaling of echo returns realizes a challengingly long antenna aperture along the flight path
Scalable Synthetic Data Generator

• Generates synthetic raw SAR complex data

• Data size is scalable to enable rigorous testing of high performance computing systems
  – User defined scale factor determines the size of images generated

• Generates ‘templates’ that consist of rotated and pixelated capitalized letters

Source Code adapted from Soumekh, 1999.
Kernel 1 — SAR Image Formation

Spatial Frequency Domain Interpolation

\[ s(t,u) \xrightarrow{\text{Fourier Transform}} \mathcal{F}(t,u) = \mathcal{F}(\omega, ku) \]

Interpolation
\[ k_x = \sqrt{4k^2 - ku^2} \]
\[ k_y = ku \]

Inverse Fourier Transform
\[ \mathcal{F}^{-1}(k_x, k_y) \xrightarrow{\text{Inverse Fourier Transform}} \mathcal{F}^{-1}(x,y) \]

Source Code adapted from Soumekh, 1999.
Template Insertion (untimed)

- Inserts rotated pixelated capital letter templates into each SAR image
  - Non-overlapping locations and rotations
  - Randomly selects 50%
  - Used as ideal detection targets in Kernel 4
Kernel 4 — Detection

- Detects targets in SAR images
  1. Image difference
  2. Threshold
  3. Sub-regions
  4. Correlate with every template → max is target ID

- Computationally difficult
  - Many small correlations over random pieces of a large image
- 100% recognition no false alarms
Benchmark Summary and Computational Challenges

Scalable Data and Template Generator

Front-End Sensor Processing

Kernel #1 Image Formation

Template Insertion

SAR Image

Back-End Knowledge Formation

Kernel #4 Detection

Detections

Validation

• Scalable synthetic data generation
• Pulse compression
• Polar Interpolation
• FFT, IFFT (corner turn)
• Sequential store
• Non-sequential retrieve
• Large & small IO
• Large Images difference & Threshold
• Many small correlations on random pieces of large image
Outline

• Introduction
• Kernel Level Benchmarks
• SAR Benchmark
• Release Information
• Summary
HPEC Challenge Benchmark Release

- **http://www.ll.mit.edu/HPECChallenge/**
  - Future site of documentation and software

- Initial release is available to PCA, HPCS, and HPEC SI program members through respective program web pages
  - Documentation
  - ANSI C Kernel Benchmarks
  - Single processor MATLAB SAR System Benchmark

- Complete release will be made available to the public in first quarter of CY06
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

• For more information, see http://www.ll.mit.edu/HPECChallenge/