

Session 4: Emerging High Performance Software

David Cousins

Division Scientist

dcousins@bbn.com

Some observations about the state of software for HPEC....

“In the HPEC community, software development generally lagged far behind hardware development.”

“However, that gap is closing because of standardization and open source software.”

“The power of C++ and STL have finally made it into HPEC.”

Altivec Extensions to the Portable Expression Template Engine (PETE)

Edward Rutledge, MIT Lincoln Laboratory

- HPEC Vendors are providing C++ with STL and usable IDEs
 - These are trickling down from the “Workstation” world.
- Designers are finding C++ and STL significantly reduces code complexity in most non-time-critical sections of the system.
 - Also known as Digital Signal Bookkeeping.
- However, the same benefits have not been easily realized in numerically intensive and time sensitive sections of the code.
 - It is very hard to write vector and matrix object libraries that are time and resource efficient.
- **PETE** is an approach that addresses this problem.
 - another approach is **VSIPL++** (Session 5 tomorrow)

What is PETE?

- PETE is a very clever template techniques to get the C++ compiler to convert the user's source code:

```
Vector A, B, C, D;  
A = -B + 2 * C - D;
```

into compiled code equivalent to a single loop of the form

```
for (i = ... ; ... ; ...) {  
    A[i] = -B[i] + 2 * C[i] - D[i];  
}
```

- The resulting loop is more efficient because:
 - There is no the “hidden” overhead involved in C++ operator overloading such as creation of temporary objects and extra copying
 - The C++ compiler can maximize the use of templates and inlining
- However, on a G4 processor, the above code does not take advantage of the AltiVec SIMD instructions.
- This paper will describe new extensions to PETE that incorporate these features, providing results that rival hand-coded loops.

“We need tools that optimize how our software runs on our hardware and how our hardware runs our software.”

**“The HPEC community won’t wait for
Mathworks to deliver a Parallel
Matlab.”**

300x Matlab

Jeremy Kepner, MIT Lincoln Laboratory

- An update and some Benchmarks on an elegant (and free) solution which provides an MPI “look and feel” API to Matlab.
- 250 lines of “pure Matlab code”

“Mathworks made Matlab better by adding Java support.”

“The large Java development community keeps making Java better.”

**Rapid prototyping of Matlab/Java Distributed Applications using
the JavaPorts components framework**

Elias S. Manolakos, Northeastern University

- Matlab V6 supports use of Java from within Matlab code.
- JavaPorts project allows a distributed application to contain any number of Java components running on networks of heterogeneous systems with anonymous messaging (presented at HPEC 2001)
- This paper presents the new capability of JavaPorts to mix Java and Matlab in networks of heterogeneous systems.

“As HPEC systems become more oriented towards distributed, heterogeneous hardware and software development, CORBA becomes very attractive.”

Meeting the Demands of Changing Operating Conditions at Runtime Through Adaptive Programming Techniques for Network Embedded Computing

Richard E. Schantz, BBN Technologies

- Quality Objects (QuO) is a middleware extension to Real-time CORBA which allows:
 - Dynamic resource commitment to satisfy quality of service contracts.
 - Proper operation of application and system when available resources are different than expected.
 - Failure detection and recovery strategies
- QuO allows this to be done independently from the application or the underlying resource configurations

“Can UML and related design tools handle all the constraints and complexities of HPEC system designs (including various middleware components)?”

Applying Model-Integrated Computing and DRE Middleware to High-Performance Embedded Computing Applications

Aniruddha Gokhale, Vanderbilt University et al.

- Model Integrated Computing expresses application functional requirements and QoS constraints in a high level abstraction.
 - UML models are one example
- Paraphrasing from the abstract:
 - “Use Model Driven Architecture and Model-Integrated Computing technology to synthesize, assemble, and deploy HPEC applications based upon RT-CORBA and DP-CORBA.”

“So what are DARPA’s current interests in all this HPEC anyway?”

Designing the Future of Embedded Systems at DARPA IXO

Doug Schmidt / DARPA IXO / Invited Speaker

- Some IXO Future Projects *
 - Wide area surveillance
 - Foliage penetration MTI
 - High-dimensionality sensing
 - Detection and tracking of dismounts
 - Automatic target recognition
 - Continuous target tracking
 - Network-centric enabling technology
- Many advances in HPEC hardware and software are required in order to meet these challenges.

* From DARPATech 2002 IXO Brief, Dr. Richard P. Wishner, IXO Director