Session 5
Government Funded Standards

Dr. Edward Baranoski
MIT Lincoln Laboratory

This work is sponsored by the High Performance Computing Modernization Office under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the Department of Defense.
Outline

- Introduction
- DoD Need
- Standards Goals
- Key Efforts
- Summary
Why Is DoD Concerned with Embedded Software?

- COTS acquisition practices have shifted the burden from “point design” hardware to “point design” software (i.e. COTS HW requires COTS SW)
- Software costs for embedded systems could be reduced by one-third with improved programming models, methodologies, and standards

Source: “HPEC Market Study” March 2001

Estimated DoD expenditures for embedded signal and image processing hardware and software ($B)
Evolution of Software Support Towards “Write Once, Run Anywhere/Anysize”

- Application software has traditionally been tied to the hardware
- Many acquisition programs are developing stove-piped middleware “standards”
- Open software standards can provide portability, performance, and productivity benefits
- Support “Write Once, Run Anywhere/Anysize”
**DoD Standards Goal**

**Goal:** Transition advanced software technology and practices into major defense acquisition programs

- **DARPA**
  - Applied Research
  - Development
  - Demonstration
  - Programs

**Common Imagery Processor (CIP)**
- Shared memory server
- Embedded multi-processor
- APG-73

**Enhanced Tactical Radar Correlator (ETRAC)**
Measuring Success

**Program Goals**
- Develop and integrate software technologies for embedded parallel systems to address portability, productivity, and performance
- Engage acquisition community to promote technology insertion
- Deliver quantifiable benefits

**Portability**: reduction in lines-of-code to change port/scale to new system
**Productivity**: reduction in overall lines-of-code
**Performance**: computation and communication benchmarks
Outline

- Introduction
- Key Efforts
  - VSIPL
  - VSIPL++
  - DRI
  - SCA
- Summary
Government funded standards must integrate with the entire system.

Definitions
- VSIPL = Vector, Signal, and Image Processing Library
- MPI = Message-passing interface
- MPI/RT = MPI real-time
- DRI = Data Re-org Interface
- CORBA = Common Object Request Broker Architecture
- HP-CORBA = High Performance CORBA
Development Status of the
Vector, Signal, and Image Processing Library (VSIPL)

Mark Richards / Georgia Institute of Technology
Dan Campbell / Georgia Tech Research Institute
Randall Judd / U.S. Navy SPAWAR Systems Center
James Lebak / MIT Lincoln Laboratory
Rick Pancoast / Lockheed Martin

Will describe API status, vendor adoption and Forum plans

Some other VSIPL work at HPEC:
• VSIPL, from API to Product, Sacco/SKY
• National Weather Radar Testbed, Walsh/SKY
• SIP-7 Experience, Linderman & Bergmann / AFRL
• HPEC-SI Demonstration, Sroka / MITRE
VSIPL++

VSIPL++: Intuitive Programming Using C++ Templates

Mark Mitchell Jeffrey D. Oldham
CodeSourcery LLC

Implementors of prototype VSIPL++

Will describe API and its benefits:
  - Direct support for parallel computation
  - Simpler syntax and improved type-checking
  - Reduced validation verification (V&V) costs
  - Support for specialized data storage formats
  - Potential for higher performance
HPEC-SI: VSIPL++ and Parallel VSIPL

- Demonstrate insertions into fielded systems (e.g., CIP)
  - Demonstrate 3x portability

- High-level code abstraction
  - Reduce code size 3x

- Demonstrated scalability

Phase 1
- Applied Research: Unified Comp/Comm Lib
- Development: Object-Oriented Standards
- Demonstration: Existing Standards

Phase 2
- Applied Research: Fault tolerance
- Development: Unified Comp/Comm Lib
- Demonstration: Object-Oriented Standards

Phase 3
- Applied Research: Self-optimization
- Development: Fault tolerance
- Demonstration: Unified Comp/Comm Lib

Unified embedded computation/communication standard

Functionality

Time

MIT Lincoln Laboratory
Technical Scope

Development

- MAPPING (data parallelism)
- Early binding (computations)
- Compatibility (backward/forward)
- Local Knowledge (accessing local data)
- Extensibility (adding new functions)
- Remote Procedure Calls (CORBA)
- C++ Compiler Support
- Test Suite
- Adoption Incentives (vendor, integrator)

Applied Research

Parallel VSIPL++

- MAPPING (task/pipeline parallel)
- Reconfiguration (for fault tolerance)
- Threads
- Reliability/Availability
- Data Permutation (DRI functionality)
- Tools (profiles, timers, ...)
- Quality of Service
Data Reorganization Interface (DRI)

Kenneth Cain, Jr. / Mercury Computer Systems
Anthony Skjellum / MPI Software Technology

Technology Focus

• Higher level abstraction for collective communication (i.e. “corner turn”)

Will describe API status, vendor adoption and Forum plans
Software Communications Architecture (SCA)
Compliant Software Defined Radios
S. Murat Bicer / Mercury Computer Systems
Jeffrey Smith / Mercury Computer Systems

Technical goal:
• Open architecture radios across multiple domains

Will describe:
• Advantages and difficulties of implementing a SCA-compliant software defined radio
• An implementation to define a Minimum SCA OMG Specification
Summary

• Government funded standards play a key role in transitioning DoD developed technology into DoD systems

• Four efforts are critical for the future success of DoD embedded computing systems
  – VSIPL
  – VSIPL++ and Parallel VSIPL
  – DRI
  – SCA