



Future Development of VSIPL

Mark Richards

*Georgia Institute of Technology
School of Electrical & Computer Engineering*









Current VSIP L Forum Products

- **Standard API for Vector/Signal Processing**
 - Version 1.01 released March 6, 2001
 - minor corrections and updates to VSIP L 1.0
- **TASP VSIP L demonstration library**
 - Developed by Randy Judd of USN SSC-SD
 - ANSI C production mode implementation
 - Core and Core lite profiles
 - “Core Plus” implementation including additional functionality
- **Portable C Test Suite**
 - Developed by Dan Campbell of GTRI
 - Tests compliance with Core Lite Profile of VSIP L 1.01 API
 - Does not test performance (speed or memory)

All may be downloaded from VSIP L web site
<<http://www.vsipl.org>>



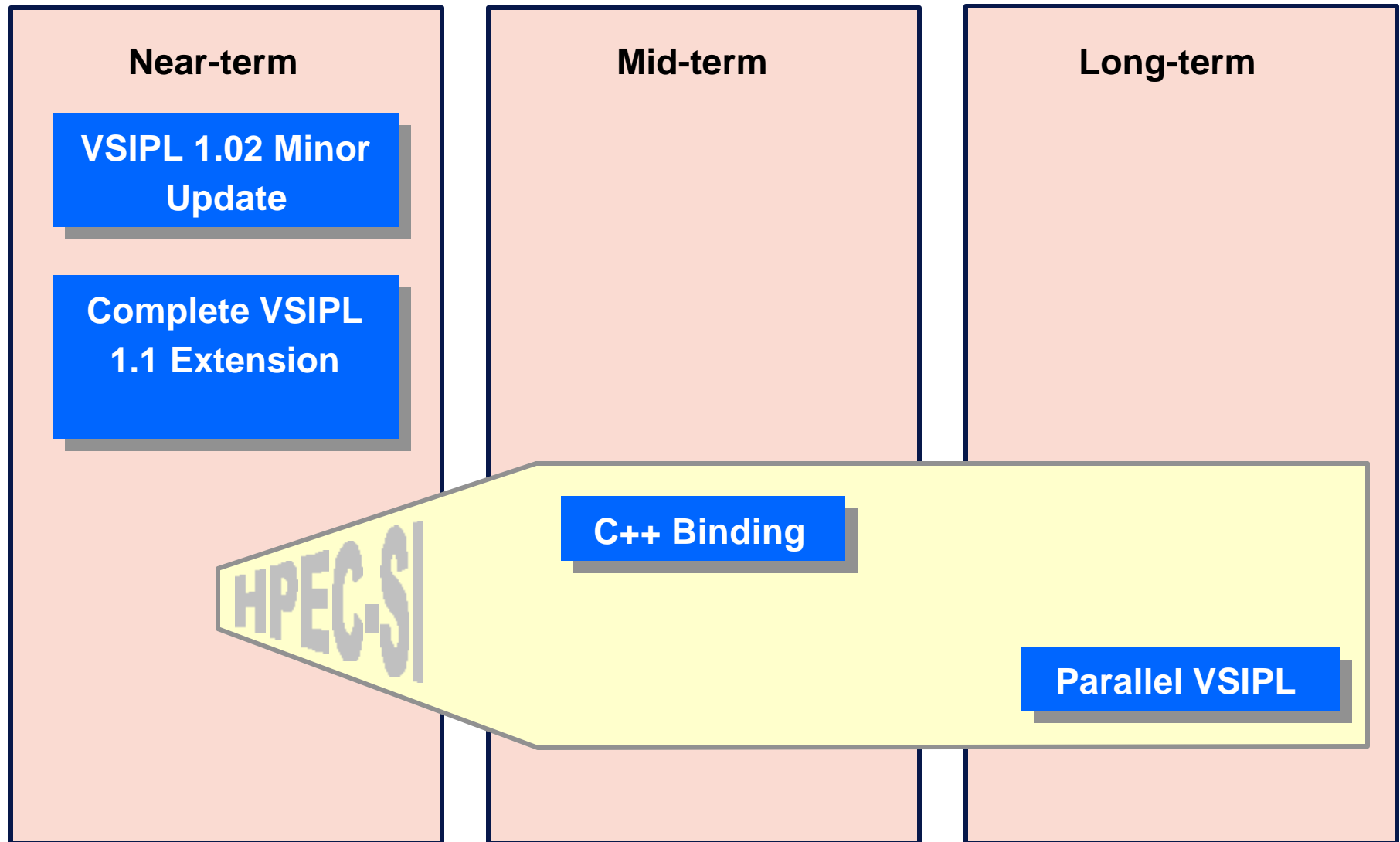
Current Commercial Implementations (Oct. 2001)

<i>Available Now</i>		
	<i>Vendor</i>	<i>Implementation*</i>
	CSPI	Core Lite
	DNA Computing Solutions	Core
	MCCI Autocoding Toolset	Core Lite
	Mercury Computer Systems	Core Lite + some 2D
	MPI Software Technology, Inc. <i>Licensed by Thales, Radstone, Concurrent; Supports G4 under VxWorks, LynxOS, Linux, MacOS, Linux PPC, and Windows</i>	Core
	Sky Computers	"Core Plus", multiple data types
<i>In Development</i>		
Annapolis MicroSystems' WildStar Boards		
MathWorks considering VSIPL target for Simulink		

** Most vendors also accommodate specific customer requirements*



VSIP L Development Plans





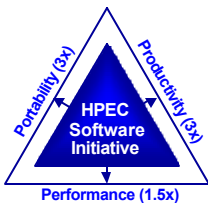
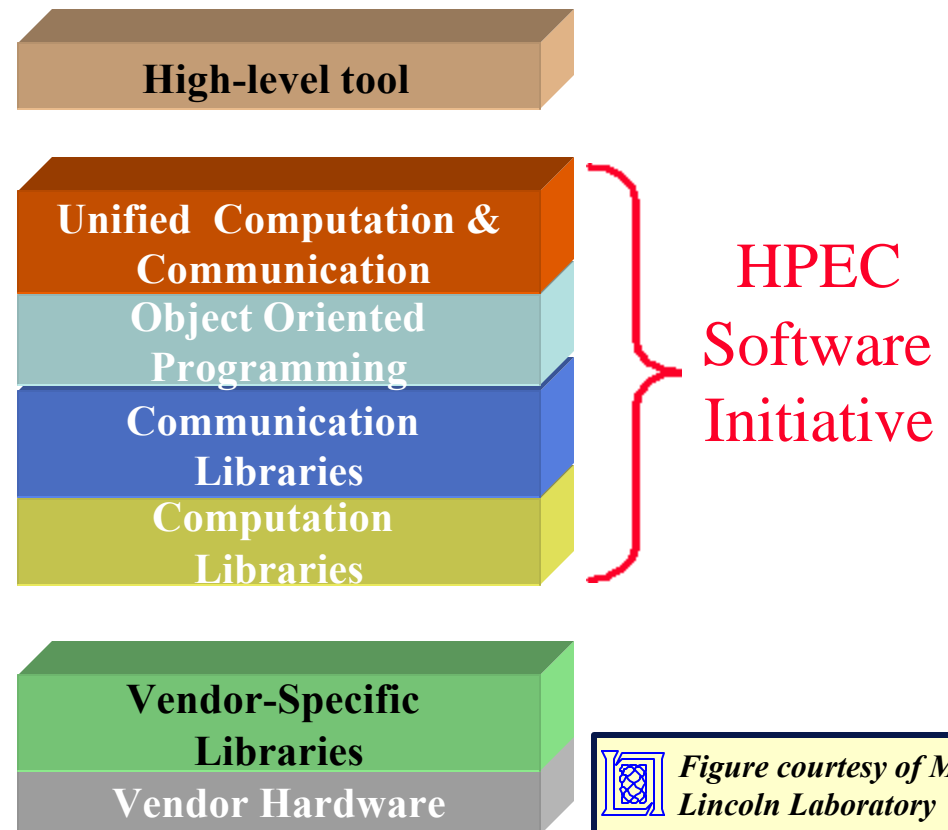
Changes in VSIP L 1.1

- **Modest Update**
- **Correction of various errata**
- **New functions**
 - **Singular value decomposition, $A = USV^H$**
 - includes functionality to extract subspaces corresponding to the highest or lowest singular values
 - supports pre- and post multiplication of a matrix by U or V
 - no solver, use with existing matrix multiplication routines
 - **Windowed FFT**
 - Defines window as part of the FFT object
 - Integrate data taper and FFT calculation
 - **New I/O functions to**
 - Operate on VSIP L vendor-dependent objects (e.g., FFT, QR, LUD)
 - differ from block objects because the data associated with them is implementation-dependent
 - allows objects to be communicated, saved to files, *etc.*



High Performance Embedded Computing Software Initiative

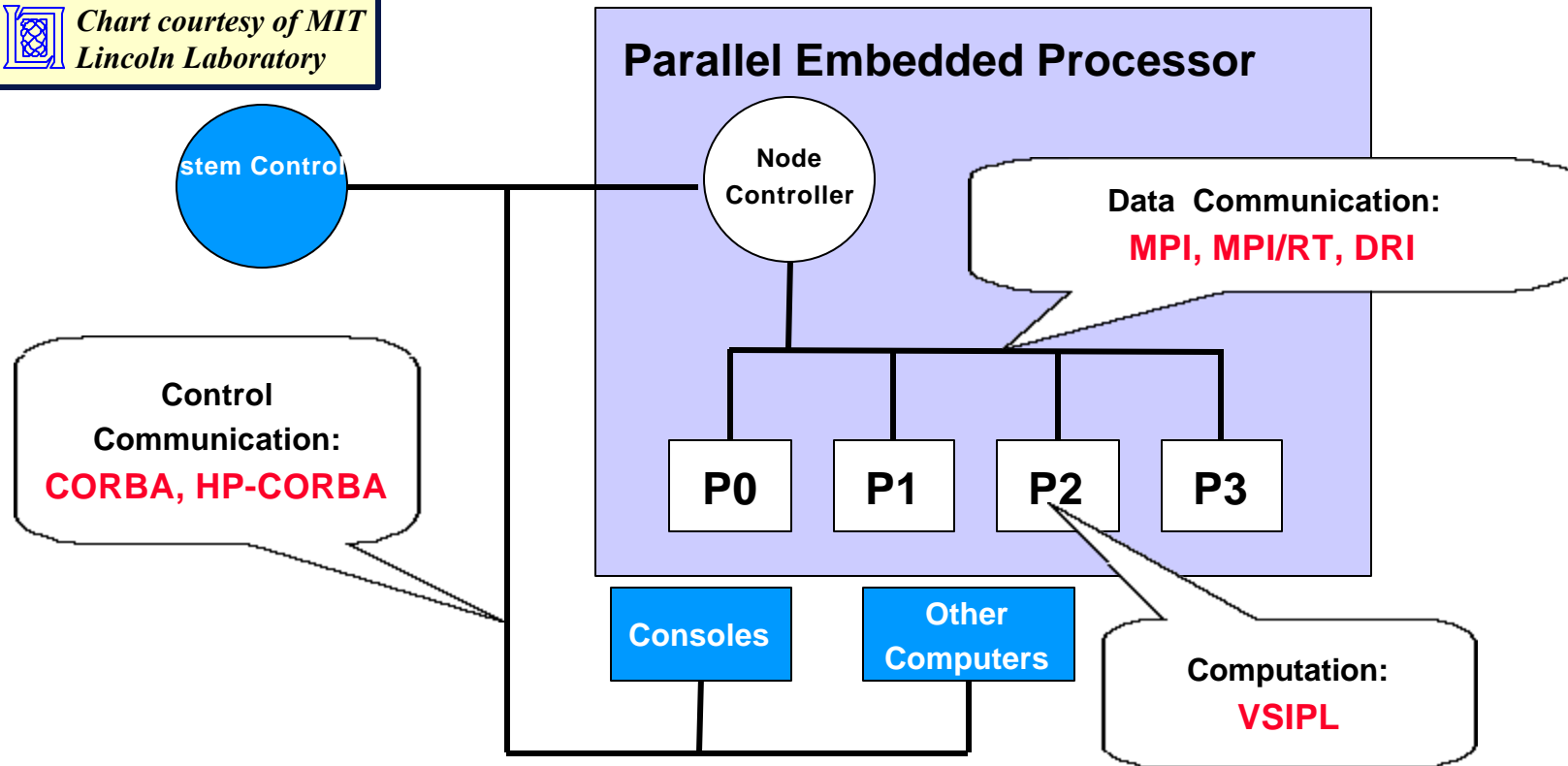
- HPEC-SI seeks to bridge the gap between high level tools and embedded hardware by building and extending on existing open standards such as VSIP, MPI, DRI, *etc.*
- HPEC-SI extensions will extend VSIP into embedded niches not currently addressed
 - C++ binding
 - parallel data distribution and computational algorithms





System Based on Component Standards

Chart courtesy of MIT
Lincoln Laboratory



- **HPEC Software Initiative - Builds on completed research and existing standards and libraries**

Definitions

VSIP L = Vector, Signal, and Image Processing Library

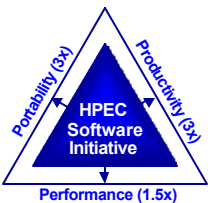
MPI = Message-passing interface

MPI/RT = MPI real-time

DRI = Data Re-org Interface

CORBA = Common Object Request Broker Architecture

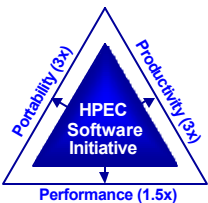
HP-CORBA = High Performance CORBA





VSIP L++

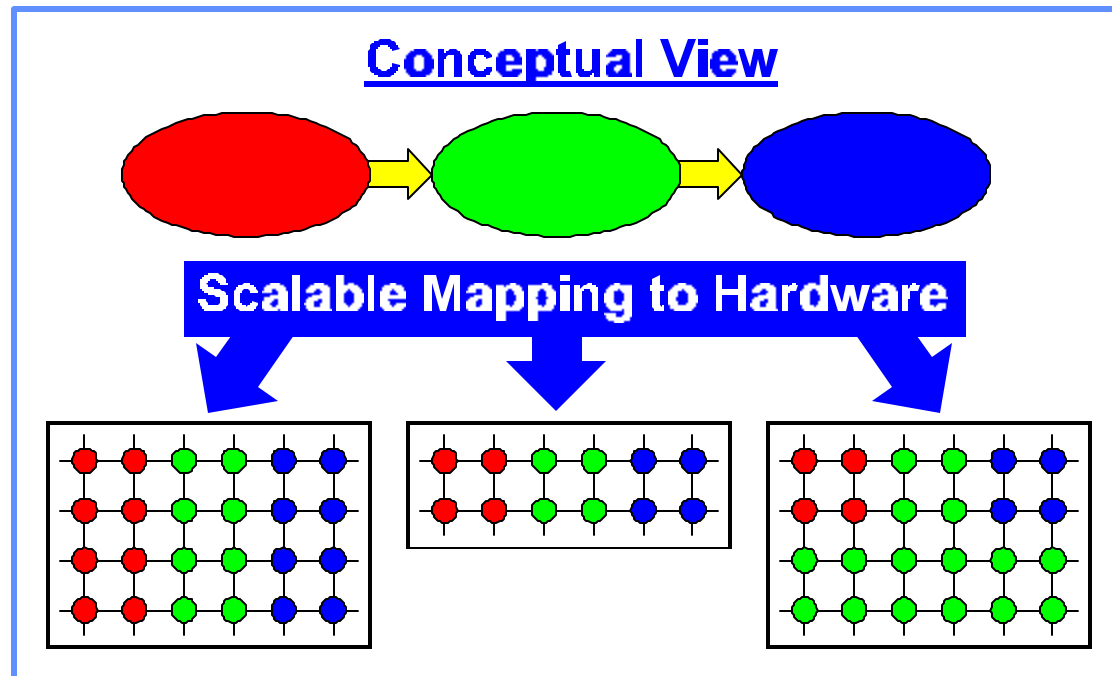
- **C++ binding offers several benefits over C binding**
 - much more compact code
 - drastic reduction in number of function prototypes
 - enables advanced techniques such as generic programming using standard template library
- **HPEC-SI currently elaborating and debating proposals**
- **Goal is to have a demonstration and prototype code in Summer 2002**



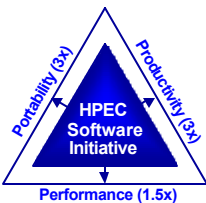


Parallel VSIP L

- **Standard VSIP L machine model is a single threaded uniprocessor**
- **Efficient parallel algorithms require**
 - coordinated data distribution and parallel algorithms strategies
 - user control of data distribution
 - scalability of algorithm to different machine sizes and layouts



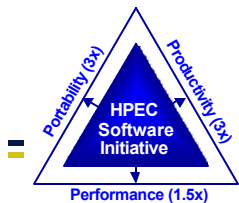
*Figure courtesy of
Dennis Cottel,
USN SSC-SD*





Parallel VSIP L Status

- **HPEC-SI is debating approaches to development of parallel VSIP L**
 - key issue is memory management strategy (blocks and views)
- **Candidate components to build on include:**
 - Data Reorganization Interface (DRI)
 - MIT/LL-Lockheed Martin Parallel Vector Library (PVL)
 - USN SSC-SD Scalable Programming Environment (SPE)
 - Los Alamos POOMA object-oriented parallel library, PETE portable expression template engine
- **Goal is to have a demonstration in Spring – Summer 2003**





Future Embedded Computing Programs that May Impact VSIP L

(or be impacted by VSIP L ...)

- **High Performance Computing Modernization Program**
 - **Computational Technology Area on Signal and Image Processing**
 - **Naval Oceanographic Office Major Shared Research Center (MSRC) project**
- **DARPA**
 - **Polymorphous Computing Architectures**
 - **High Productivity Computing Systems**
 - **program announced February 2002**



HPCMP NAVO MSRC VSIPL Project

- **GOAL: Develop high-performance VSIPL implementations for MSRC supercomputer platforms**
 - IBM SP, Sun E10000, SGI ORIGIN, MIPS
- **Initial version:**
 - Core profile
 - C wrappers around optimized BLAS, FFTW, and TASP VSIPL
 - double precision and long integer
 - “good” performance
- **Delivered August 2001**



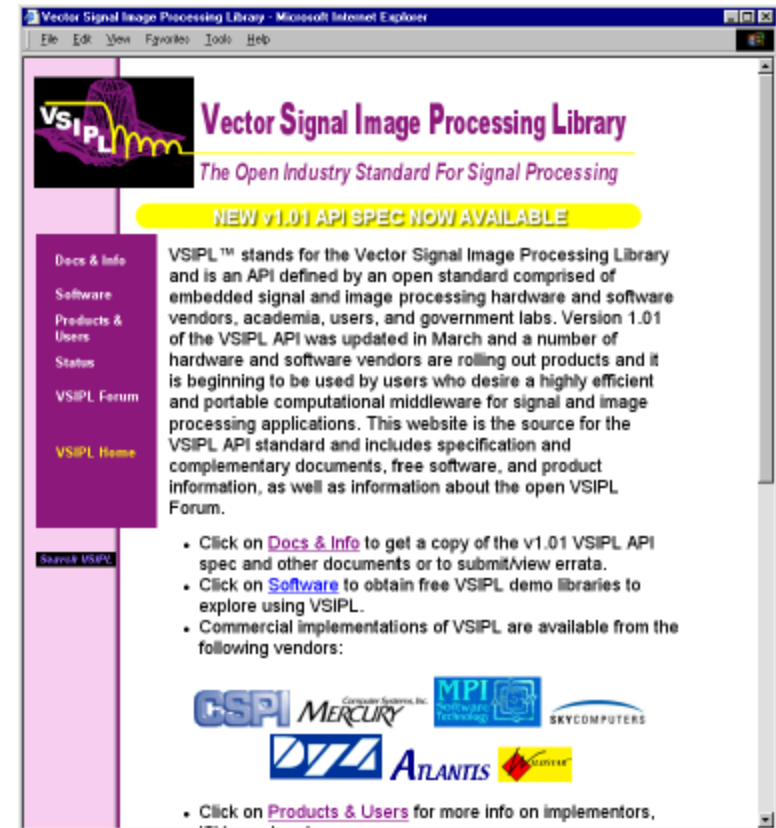
High Productivity Computing Systems

- **New DARPA program**
 - FY02 Phase 1 Industry Study BAA released Feb. 2002
- **HPCS “... creates a new generation of high productivity computing systems characterized by balanced system architecture including high effective bandwidth, robust implementation, and responsive software/hardware components. These new systems will address the inherent difficulties associated with the development and use of current high end systems and applications such as programming productivity, performance portability, scalability, reliability, and tamper resistance. This program is targeting the high end computing medium to long term national security application requirements”**
- **FY2002 goals:**
 - Identify application requirements.
 - Initiate productivity benchmarks and stressmarks.
 - Develop innovative programming models and virtual machine forms.
 - Explore scalable computing programming and profiling techniques.



Summary

- **VSIP L 1.01 is available**
 - VSIP L 1.02 very soon
- **VSIP L 1.1 is under active development**
- **Implementations are here**
- **The Forum is continuing to support and enhance VSIP L**
 - cooperating with HPEC-SI on C++ and Parallel VSIP L
 - leads into future work in HPCS and PCA
- **Actively soliciting examples and “field reports”**



www.vsipl.org