Abstract Presentation Back to Agenda Next Abstract

Successful VSIPL Software Application Migration
A Case Study:
NATO Seasparrow Illumination Radar Signal Processing

Dan Averill
Avatar Engineering, Inc.
Phone: (410) 271-3630
Fax: (410) 267-0642
Email: dan.averill@avatarengineering.com

An embedded weapon system signal processing software application involving 10K SLOC was converted from vendor proprietary middleware to that vendor's VSIPL implementation, deployed aboard US Navy surface combatants, and then ported without modification to a less expensive hardware platform using a different vendor's commercially available VSIPL product. This successful episode of software portability offers an opportunity to assess the utility of VSIPL in the context of a production military application, and yields user and vendor strategies and forecasting information for similar future efforts to cost-effectively leverage the products of this research community.

The NATO Seasparrow ship self defense missile is guided to a maneuvering high velocity inbound target by a ship based 10.125 GHz illumination radar. At 43 Hz, the radar receiver time series is digitized and processed for tracking data products. Inbound targets manifest as Doppler shifted energy spikes in the frequency spectrum. A conically scanned receive antenna scheme imparts amplitude modulation across the time series for off-center targets. A 2 KHz carrier frequency modulation smears the Doppler shift of a target over multiple frequency bins as a function of range. The Naval Research Laboratory (NRL) developed a modest set of signal processing algorithms modeled in MatLab to yield track velocity, traverse and elevation angle errors, estimated range, and audio-video spectral representations given the input time series. The software implementation of these algorithms served as the study case for this VSIPL application migration effort.

Before the VSIPL migration, the legacy implementation of this algorithm set was hosted on a Sky Computers SkyBolt II Excalibur. The source code contained approximately 300 vector library calls to 68 distinct Sky Standard Math Library (SML) functions, including single precision 4K complex FFTs, a 7:1 FIR filter decimation, and several 64 point forward and inverse FFTs. The migration target hardware selected by the design agent is a Motorola MVME5100 single board computer with an Altivec PowerPC processor. The MPI Softtech VSI/Pro product provides a VSIPL compliant vector library compatible with Wind River Systems vxWorks in that hardware architecture. Significant production unit cost savings per radar system are realized by replacing the legacy hardware and software architecture with the migration target architecture, but only if the engineering cost of the migration is minimized. A two stage risk averse strategy was adopted: first convert the algorithm set to VSIPL on the legacy platform and verify

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
20 AUG 2004

2. REPORT TYPE
N/A

3. DATES COVERED
-

4. TITLE AND SUBTITLE
Successful VSIPL Software Application Migration A Case Study: NATO Seasparrow Illumination Radar Signal Processing

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Avatar Engineering, Inc.

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001694, HPEC-6-Vol 1 ESC-TR-2003-081; High Performance Embedded Computing (HPEC) Workshop (7th)., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**26**	

equivalent numeric results, and second recompile the validated VSIPL application for the migration target platform and again verify equivalent numeric results.

The presentation of this case explores the object management challenge presented when converting software from an “operate on bare memory” paradigm to the persistent object paradigm presented by VSIPL. Brief examples instantiating certain subtle implications of the VSIPL standard when applied in a real time embedded software environment are offered, as well as a systematic scheme for identifying and implementing the minimal set of objects necessary to convert an arbitrary legacy signal processing application to VSIPL. The “hints” that would have accelerated the effort under study had they been available before the effort began are offered for the benefit of similar future efforts.

The presentation provides quantitative code growth and performance impact measurements for the studied application in the legacy hardware. Also provided are measured engineering level of effort and schedule as a function of pre-migration source size, for at least the studied case. Since the only tangible product of a successful first stage effort is slower performance, the presentation offers observations on the programmatic benefits (deferred gratification) of conversion to VSIPL. The presentation encourages the managers of similar military application systems to migrate their software to VSIPL.

The second stage of the effort surfaced VSIPL implementation compatibility and relative maturity data. It is gratifying to report that the application ultimately initialized and iterated with acceptable performance and accuracy and without any algorithm modification. The configuration management advantages of this fact are significant to all fielded configurations of the studied weapon system. Result precision, numerical accuracy, and implemented function inventory all emerge as factors when porting a legacy application to a new platform. The utility of the VSIPL development version to the porting effort was limited; such a tool seems likely to be of greater value to an initial implementation. Vendor product support plays an important role at both ends of a port, and user regression testing of an already validated application can yield valuable benefits back to the VSIPL library vendor. The presentation offers vendors an insight into the VSIPL end user experience, permitting a more competitive vendor offering to such users.

This abstract asserts the relevance of the studied case to the topics of “Middleware Libraries and Application Programming Interfaces” and “Case Study Examples of High Performance Embedded Computing”. It has summarized the involved application and the software implementation of that application, highlighted the specific subjects explored by the offered presentation, and identified notional benefits of the presentation to a cross section of the research community. The HPEC technical committee is respectfully requested to consider including the offered presentation in the workshop.

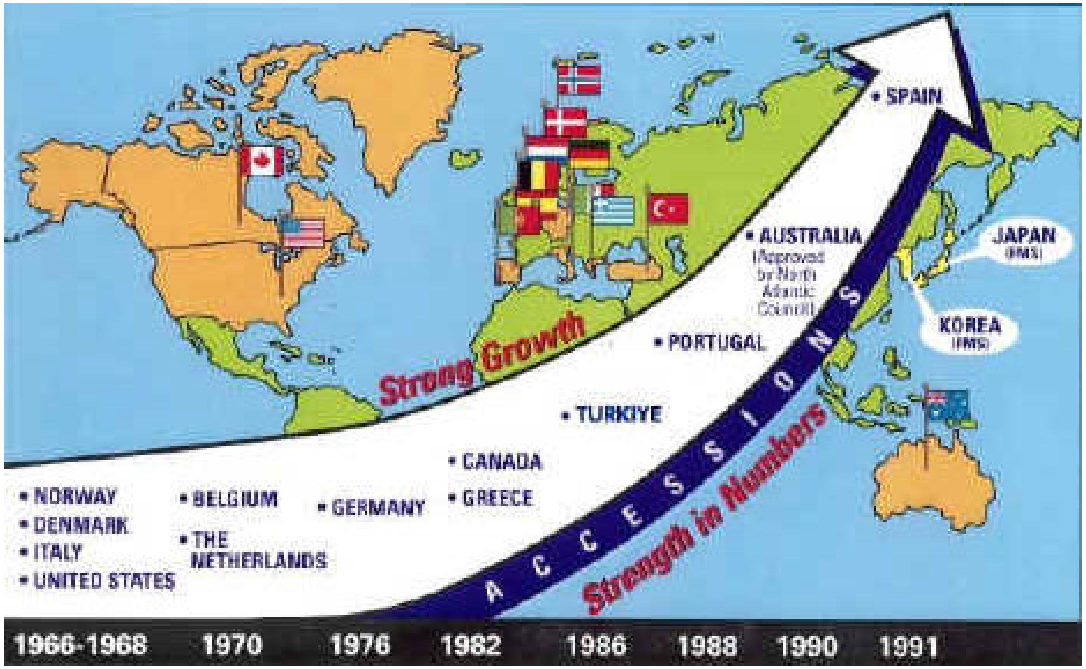
Abstract
Presentation
Back to Agenda
Next Abstract

Successful VSIPL Software Application Migration

A Case Study: NATO SEASPARROW Illumination Radar Signal Processing

Sponsor Information

- NATO SEASPARROW Project Office
  - NATO's Largest And Most Successful Cooperative Weapons Project
    - (18 participating or user governments)



VSIPL Application Migration

- Application Characteristics
- Justification
- Migration Strategy
  - Port In Place To VSIPL
  - Port VSIPL Application To Alternate Platform
- Process Detail
- Results And Experiences
- Costs And Benefits
- Observations On VSIPL
- Observations On Migration

Application Characteristics

- 10 GHz Continuous Wave Illumination Radar Signal Processing Software
- 140 KHz Bandwidth Of Interest
- 43 Hz Iteration Rate
- 7:1 FIR Decimation Of Digitized Time Series
- Two 4K FFTs
- Several 64 Point Inverse And Forward FFTs
- Analyze AM and FM Signal Components To Produce Tracking Data Products

Hardware Platforms

- Existing Platform
  - Sky Computers SkyBolt II Excalibur Hardware
  - Vector Processing Implemented On Sky Standard Math Library
  - Software Is Tested And Deployed Aboard Surface Combatants
- Target Platform
  - Motorola MVME5110 (MPC 74xx) Hardware With Echotek GC214 DDR
  - MPI Software Technology VSI/Pro VSIPL

Software Application Metrics

- 10K Source Lines Of Code
- 15% Comment
- 26 Source And Header Files
- 300 Calls To Vector Library
- 65 Unique Vector Functions Used

- Not Strictly A "Metric", But Application Made Liberal Use Of extern Statement For Data Structure Access

Justification For Migration

- NSPO Requires Additional Copies Of Deployed System
  - Existing Platform Includes COTS Products No Longer Available
  - Replacement COTS Processing Capacity Far Exceeds Application Requirement
- Significant Programmatic Investment In Signal Processing Algorithms
  - Development
  - Verification And Testing
  - Ongoing Algorithm Enhancement
- VSIPL Available On Several COTS Computer Platforms
  - Permits Competitive Hardware Cost / Performance Evaluation
  - Future Software Upgrades Portable To All Installations

Risk Reduction Migration Strategy

- Port In Place To VSIPL
  - Sky Computers Provides Correct And Complete VSIPL Implementation As A Layer Over Sky Standard Math Library
  - Verify Port Correctness On Legacy Hardware
- Port VSIPL Application To Alternate Platform
  - Verify Equivalent Results On New Target Hardware

Iterative Real Time Software Constraints

- VSIPL "Create" and "Destroy" Verbs Not Available To Iterative Software
  - Heap Fragmentation Causes Indeterminate Execution Time
  - All "Create" And "Destroy" Calls Confined To Initialization
- Compels The Cataloguing Of Objects Implied By The Existing Code
- Requires Initialization Code To Create Each Needed Object With Proper Scope

The Persistent Object Management Challenge

- Vector Offsets, Lengths, And Strides
  - Were Freely Specified As Function Arguments
  - Now Carried As View Object Attributes
  - Forces Evaluation Of Memory Block Utilization: Need For Different Simultaneous Views Of Any Memory Block
  - Forces Coherent Policy On Attribute Persistence
- FFT Objects
  - Forward, Reverse
  - In-Place, Out-Of-Place
- FIR Filter Objects
- Chebyshev Weight Objects
- Real And Imaginary Derived Views Of Complex Vectors

- Most Conversion Coding Errors Will Be Found In View Attributes

Steps To Port In Place To VSIPL

1) Catalog Existing Vector Library Calls
   - Link Without The Vendor Library
   - Compile Without The Vendor Header File
2) Map To Functionally Equivalent VSIPL Calls
   - Solicit Vendor Mapping
3) Address Surprises And Subtleties
   - FIR Filter
   - Inverse FFT
4) Confront The Object Management Challenge
   - Catalog Vector Memory Buffers Used In Legacy Application
   - Catalog Distinct Offset And Stride Combinations
   - Identify Vector Data Types / Precisions In Use
   - Catalog High Order Process Objects
     - FFTs
     - FIR Filters
     - Chebyshev Weights
   - Identify Scope Of Use

Steps To Port In Place To VSIPL (continued)

5) Add Initialization Functions Throughout Application
   - Create Or Pass Address(es) Of Required Objects
6) Implement VSIPL Function Call Mapping
   - Use Conditional Compilation Switches So VSIPL Calls Are Side-By-Side With Legacy Calls
   - Instrument For Formatted Vector Capture At Manageable Source Interval
   - Pay Special Attention To View Attribute Values (offset, length, stride)
7) Compile Clean With And Without VSIPL Switch
8) Test And Verify Numeric Results
   - Correct Errors
   - Iterate Until Accurate
9) Strip Legacy Source And Compilation Switches Out Of Implementation
10) Deliver

Inverse FFT Before And After

-
  ```
  ccopy(VECSZ,
    psTemp, 1,
    psTemp + 2 * VECSZ, 1);
    /* triplicate */

  ccopy(VECSZ,
    psTemp + VECSZ - shift, 1,
    psTemp, 1);
    /* rotated spectrum */

  cvfftb(psTemp,
    psTemp,
    VECSZ,
    -1); /* time series */
  ```

-
  ```
  fftInvRanging =
    vsip_ccfftip_create_f(
      (vsip_length) VECSZ,
      1.0 / (float) VECSZ,
      VSIP_FFT_INV,
      0,
      VSIP_ALG_TIME);
  ```

-
  ```
  vsip_cvcopy_f_f(pcvSample,
    vsip_cvputoffset_f(pcvSampxDup,
    2 * VECSZ)); /* triplicate */

  vsip_cvcopy_f_f(
    vsip_cvputoffset_f(pcvSampxDup,
      VECSZ - shift),
    vsip_cvputlength_f(pcvCenterdx,
      VECSZ));
    /* rotated spectrum */

  vsip_ccfftip_f(fftInvRanging,
    pcvCenterdx); /* time series */
  ```

Vector Memory Allocation Before And After

-
  ```
  if ((psMainSpec
        = (complex *)valloc(
           sizeof(complex) * FFTSIZE))
     == NULL)
     printf("Error\n"), fflush(stdout);
  else
     vclr((float *)&psMainSpec->cr),
        1,
        sizeof(complex) * FFTSIZE);
  ```

-
  ```
  if ((pcvMainSpec
        = vsip_cvcreate_f(FFTSIZE,
           VSIP_MEM_NONE))
     == NULL)
     logMsg("Error\n",0,0,0,0,0,0);
  else
     vsip_cvfill_f(
        vsip_cmplx_f(0.0, 0.0),
           pcvMainSpec);
  ```

Surprises And Subtleties

- FIR Filter Inversion And Ramp-Up
  - VSIPL Version More Flexible
  - VSIPL Optionally Preserves Filter State From Previous Batch
  - Impossible To Obtain Identical Behavior Porting From Sky Math Library
- Inverse FFT Scaling
  - Behavior Built-In To Sky Math Library
  - Optional In VSIPL
  - Obvious In Hindsight / Devilish To Spot

More Minor Subtleties

- vsdiv() Maps To vsip_svmul_f() With Reciprocal Of Scalar
- cvexp() Maps To vsip_veuler_f()
- VSIPL Clipping Mechanism More General Than SML
  - Requires Analysis To Obtain Equivalent Behavior

Implementation Costs

- 2.5 Man Months Expended
  - Inception To Delivery (4 Calendar Months)
  - Includes Learning Curve
  - Sky Computers Customer Service And Algorithms Group Very Supportive Of This Effort
- SLOC Growth Nominally 10%
- Execution Time Penalty 15%
  - Misleading To Discuss As Percentage
  - Actually A Fixed Overhead
  - Higher Frequency Applications Will See Larger Penalty As Percentage Of Execution Time

Target Platform Constraints

- FIR Filter Decimation Moved To Hardware (GC214 DDR)
  - Input Complex Time Series Presented At Slightly Later Point In Signal Processing Algorithm
  - Significant Execution Time Relief
  - EchoTek Provided Support For This Transition

- MPI VSI/Pro Product Update Overlapped Application Migration
  - Mutually Beneficial Issue Resolution Cycle With MPI Customer Service
    - Function Inventory
    - Numeric Precision
    - Compiler Switches
    - Address Relocation
    - Library Footprint
  - VSI/Pro 1.10 Is A Solid Product

Steps To Port In Place To Target Platform

1) Compile Source In New Target Development Environment
   - Link With New Vendor Library
   - Identify And Resolve Unimplemented VSIPL Function Issues
   - Identify And Resolve Input / Output Issues Peculiar To New Target Environment (vxWorks)
   - Protect Platform Specific I/O With Conditional Compilation Switches

2) Test And Verify Numeric Results
   - Correct Errors
   - Iterate Until Accurate
   - Likely To Succeed On Initial Iteration

3) Deliver

- <u>Numeric Errors (If Any) Anticipated To Be Precision Issues</u>

Subtleties And Implementation Costs

- Minor Subtlety
  - Little Known taskSpawn() Argument VX_ALTIVEC_TASK Required
  - Opaque Error "Altivec Unavailable" Provided By BSP If Flag Is Omitted

- Implementation Costs
  - 2 Man Weeks Expended
  - 4 Calendar Months, Due To External Impacts Not Part Of This Effort
  - Execution Time Benefit: Application Cycles 4 Times As Fast, But FIR Filter Is No Longer In Software

Benefits

- Migration Was Unmitigated Success
- Application Now Excellent Example Of "Write Once / Run Anywhere" Goal Espoused On VSIPL Website
- SEASPARROW Program Can Legitimately Distribute Algorithm Enhancement Costs Across Larger Installed Platform Population

Observations On VSIPL

- Extremely Useful And Well Documented Standard

- "Development Mode" Deserves More Rigorous Specification
  - Rather Crash And Examine Stack Trace Than Exit Gracefully And Not Know What Happened
- Numerical Accuracy / Precision Is Ambiguous
  - Result Precision Or Calculation Precision
  - Possible To Obtain Numerically Different Results From Conforming Implementations Through Vendor Interpretation Of This Point
- Test Suite Fidelity Still Emerging

Observations On Migration

- Effort Is Bounded
  - No Dead Ends, No Meandering
  - Atmosphere Of "Turning The Crank" During Migration
- Low Risk
  - Application Will Still Function When Effort Concludes
- Future Maintenance Easier To Staff
  - Availability Of VSIPL Literate Software Engineers Independent Of Vendor-Specific Signal Processing Library

- Enthusiastically Recommend Similar Efforts

BAE SYSTEMS

Avatar Engineering

VSIPL Application Migration

- Application Characteristics
- Justification
- Migration Strategy
  - Port In Place To VSIPL
  - Port VSIPL Application To Alternate Platform
- Process Detail
- Results And Experiences
- Costs And Benefits
- Observations On VSIPL
- Observations On Migration