Data Reorganization Interface

Kenneth Cain
Mercury Computer Systems, Inc.
Phone: (978)-967-1645
Email Address: kcain@mc.com

Abstract:
This presentation will update the HPEC community on the latest status of the standard Data Reorganization Interface (DRI). DRI is a software interface for performing data-parallel distribution and reorganization operations (e.g., transpose, reshape) that are frequently required in scalable HPEC applications. DRI provides increased ease of use compared to point-to-point middleware by providing abstractions for multi-dimensional datasets, partitioning and distribution methods (e.g., block, block-cyclic, overlapped elements), and a high-level interface that frees applications from having to orchestrate the multitude of individual transfers required in a single data reorganization. A planned transfer approach in DRI enables high performance data transfers, and its multi-buffering semantics enable (with hardware support) time overlap of an application’s communication and computation operations. DRI is designed to enhance existing standard and proprietary middleware by adding a standard, easy to use interface without compromising high performance.

The DRI-1.0 API was ratified and published in September 2002 by the Data Reorganization Forum, and was announced at the HPEC 2002 workshop. DRI-related activities since that announcement will be discussed in this presentation, including current vendor implementation status, a summary of results from the first use of DRI in a realistic application demonstration (SAR image formation), and candidate features that could be added to an enhanced DRI standard. The DRI-1.0 document can be accessed on the World Wide Web at URL http://www.data-re.org.
<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 AUG 2004</td>
<td>N/A</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Reorganization Interface</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury Computer Systems, Inc.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT: unclassified</td>
</tr>
<tr>
<td>b. ABSTRACT: unclassified</td>
</tr>
<tr>
<td>c. THIS PAGE: unclassified</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17. LIMITATION OF ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>UU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>18. NUMBER OF PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
</tr>
</tbody>
</table>
Data Reorganization Interface (DRI)

Kenneth Cain Jr.
Mercury Computer Systems, Inc.

On behalf of the Data Reorganization Forum
http://www.data-re.org

High Performance Embedded Computing (HPEC) Conference
September 2003

The Ultimate Performance Machine
Status update for the DRI-1.0 standard since Sep. 2002 publication

- **DRI Overview.**

- **Highlights of First DRI Demonstration.**
  - Common Imagery Processor (Brian Sroka, MITRE).

- **Vendor Status.**
  - Mercury Computer Systems, Inc.
  - MPI Software Technology, Inc. (Anthony Skjellum).
  - SKY Computers, Inc. (Stephen Paavola).
What is DRI?

Standard API that complements existing communication middleware

- Partition for data-parallel processing
  - Divide multi-dimensional dataset across processes
  - Whole, block, block-cyclic partitioning
  - Overlapped data elements in partitioning
  - Process group topology specification

- Redistribute data to next processing stage
  - Multi-point data transfer with single function call
  - Multi-buffered to enable communication / computation overlap
  - Planned transfers for higher performance
First DRI-based Demonstration

Common Imagery Processor (CIP)

Conducted by Brian Sroka of The MITRE Corporation
CIP and APG-73 Background

**CIP**
- The primary sensor processing element of the Common Imagery Ground/Surface System (CIGSS)
- Processes imagery data into exploitable image, outputs to other CIGSS elements
- A hardware independent software architecture supporting multi-sensor processing capabilities
- Prime Contractor: Northrop Grumman, Electronic Sensor Systems Sector
- Enhancements directed by CIP Cross-Service IPT, Wright Patterson AFB

**APG-73**
- SAR component of F/A-18 Advanced Tactical Airborne Reconnaissance System (ATARS)
- Imagery from airborne platforms sent to TEG via Common Data Link

© 2003 Mercury Computer Systems, Inc.
APG-73 Data Reorganization (1)

Block data distribution

Source
- No overlap
- Blocks of cross-range
- Full range

Destination
- No overlap
- Full cross-range
- Mod-16 length blocks of range

Source: “CIP APG-73 Demonstration: Lessons Learned”, Brian Sroka, The MITRE Corporation, March 2003 HPEC-SI meeting

© 2003 Mercury Computer Systems, Inc.
APG-73 Data Reorganization (2)

Block data distribution

Source
- No overlap
- Full cross-range
- Mod-16 length blocks of range cells

Block with overlap

Destination
- 7 points right overlap
- Full cross-range
- Blocked portion of range cells

Source: “CIP APG-73 Demonstration: Lessons Learned”, Brian Sroka, The MITRE Corporation, March 2003 HPEC-SI meeting
DRI Use in CIP APG-73 SAR

DRI Implementations Used

<table>
<thead>
<tr>
<th>Application</th>
<th>Application</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>MITRE DRI</td>
<td>Mercury PAS/DRI</td>
<td>SKY MPICH/DRI</td>
</tr>
<tr>
<td>MPI *</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Demonstration completed

Demonstrations underway

* MPI/Pro (MSTI) and MPICH demonstrated

Simple transition to DRI

- `#pragma splits loop over global data among threads`
- DRI: loop over local data

for

Range compression
Inverse weighting

DRI-1: Cornerturn

for

Azimuth compression
Inverse weighting

DRI-2: Overlap exchange

for

Side-lobe clutter removal
Amplitude detection
Image data compression

Portability: SLOC Comparison

- 5% SLOC increase for DRI includes code for:
  - 2 scatter / gather reorgs
  - 3 cornerturn data reorg cases
  - 3 overlap exchange data reorg cases
  - Managing interoperation between DRI and VSIPL libraries

Using DRI requires **much less source code than manual distributed-memory implementation**

CIP APG-73 DRI Conclusions

- Applying DRI to operational software does not greatly affect software lines of code
- DRI greatly reduces complexity of developing *portable* distributed-memory software (shared-memory transition easy)
- Communication code in DRI estimated 6x smaller SLOCs than if implemented with MPI manually
- No code changed to retarget application (MITRE DRI on MPI)

Features missing from DRI:
- Split complex
- Dynamic (changing) distributions
- Round-robin distributions
- Piecemeal data production / consumption
- Non-CPU endpoints

Source: “High Performance Embedded Computing Software Initiative (HPEC-SI)”, Dr. Jeremy Kepner, MIT Lincoln Laboratory, June 2003 HPEC-SI meeting
Vendor DRI Status

Mercury Computer Systems, Inc.
MPI Software Technology, Inc.
SKY Computers, Inc.

The Ultimate Performance Machine
Commercially available in PAS-4.0.0 (Jul-03)
- Parallel Acceleration System (PAS) middleware product
- DRI interface to existing PAS features
  - The vast majority of DRI-1.0 is supported
  - Not yet supported: block-cyclic, toroidal, some replication

Additional PAS features compatible with DRI
- Optional: applications can use PAS and DRI APIs together

Applications can use MPI & PAS/DRI
- Example: independent use of PAS/DRI and MPI libraries by the same application is possible (libraries not integrated)
Hybrid use of PAS and DRI APIs

- PAS communication features:
  - User-driven buffering & synchronization
  - Dynamically changing transfer attributes
  - Dynamic process sets
  - I/O or memory device integration
  - Transfer only a Region of Interest (ROI)

Built on Existing PAS Performance

DRI Adds No Significant Overhead
DRI Achieves PAS Performance!
MPI Software Technology, Inc.

- MPI Software Technology has released its ChaMPIon/Pro (MPI-2.1 product) this spring
- Work now going on to provide DRI “in MPI clothing” as add-on to ChaMPIon/Pro
- Confirmed targets are as follows:
  - Linux clusters with TCP/IP, Myrinet, InfiniBand
  - Mercury RACE/RapidIO Multicomputers
- Access to early adopters: 1Q04
- More info available from: tony@mpi-softtech.com (Tony Skjellum)
Initial Implementation

- Experimental version implemented for SKYchannel
- Integrated with MPI
- Achieving excellent performance for system sizes at least through 128 processors
SKY’s Plans

- Fully supported implementation with SMARTpacc
- Part of SKY’s plans for standards compliance
- Included with MPI library
- Optimized InfiniBand performance