

Data Reorganization Interface (DRI)

Kenneth Cain Jr.

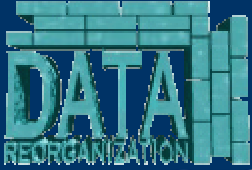
Mercury Computer Systems, Inc.

On behalf of the Data Reorganization Forum

<http://www.data-re.org>

**High Performance Embedded Computing (HPEC) Conference
September 2003**

The Ultimate Performance Machine



Outline, Acknowledgements

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

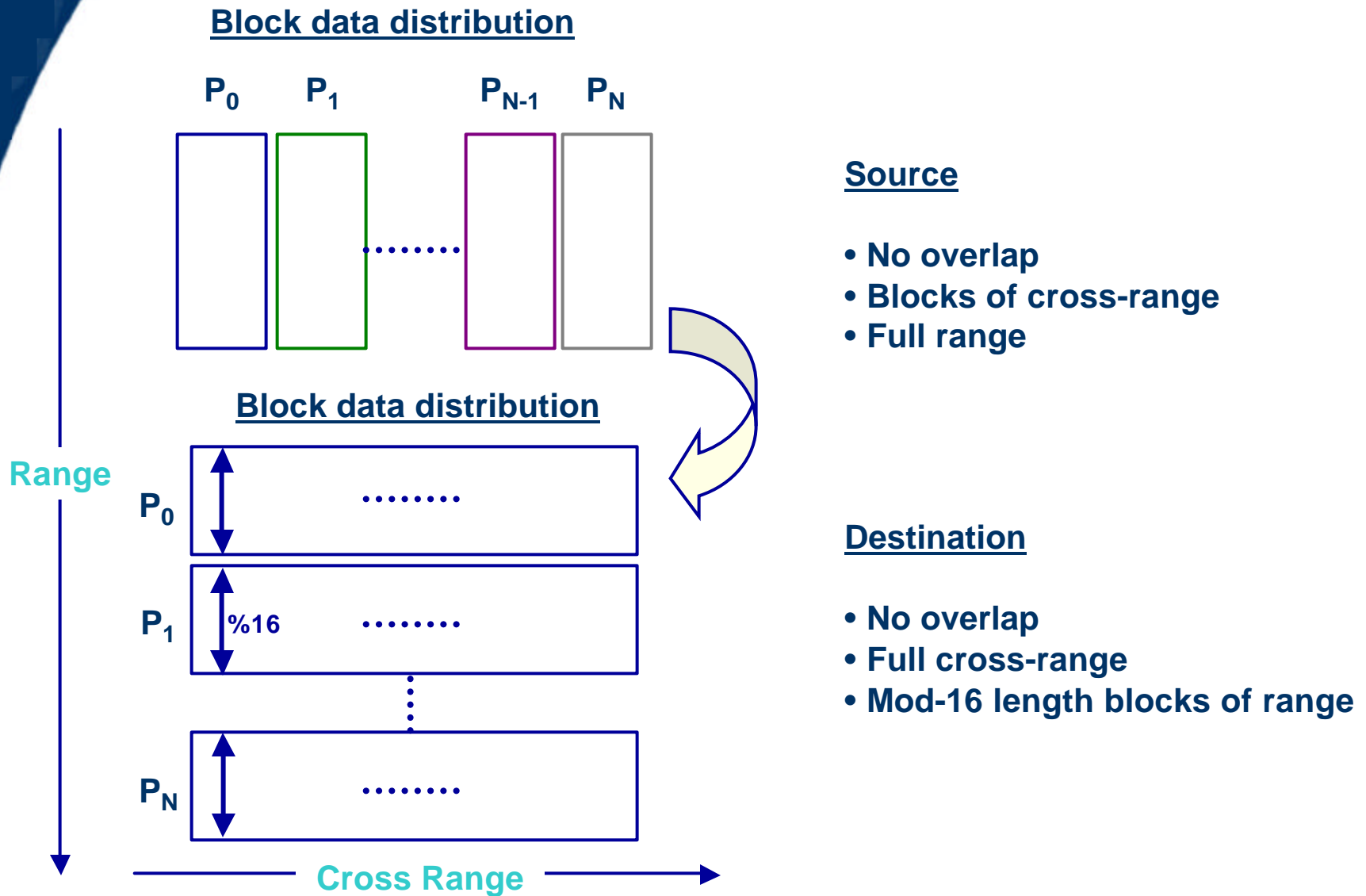
APG-73

SAR component of F/A-18 Advanced Tactical Airborne Reconnaissance System (ATARS)

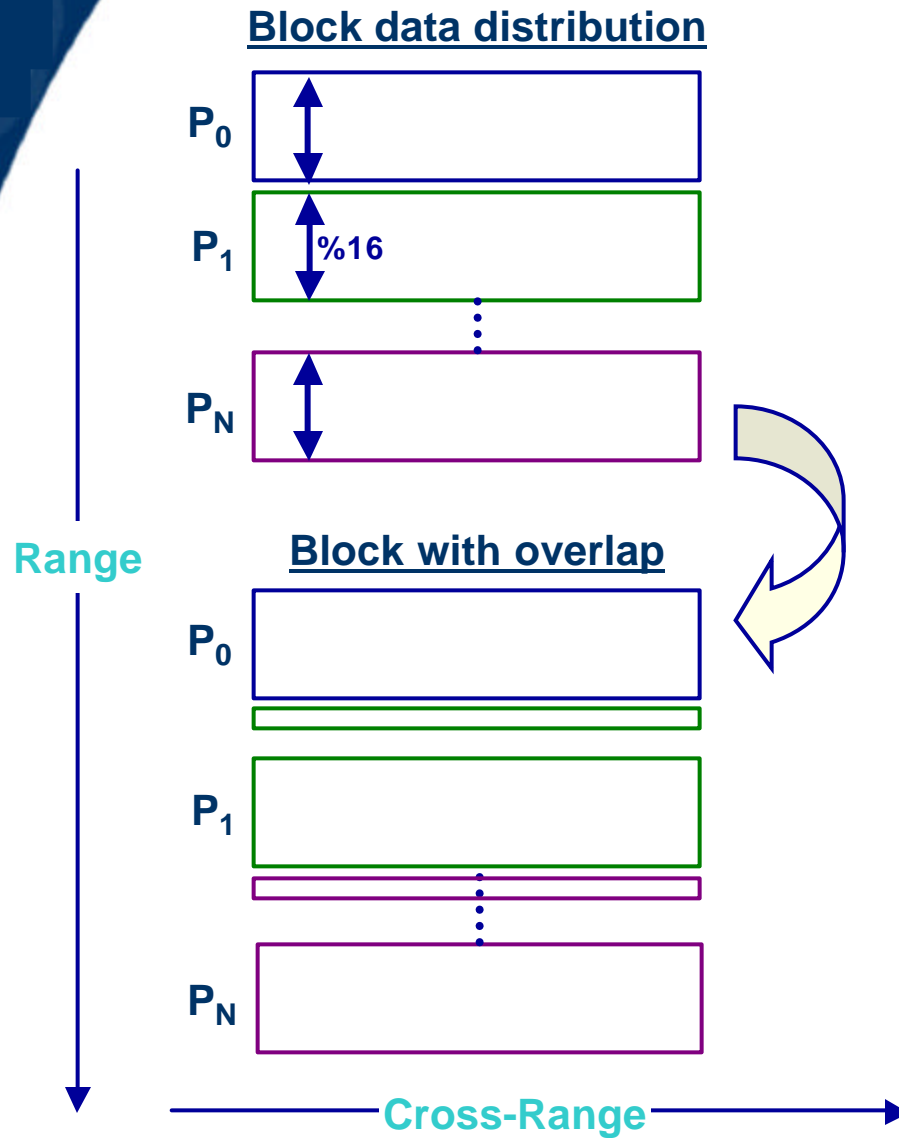
Imagery from airborne platforms sent to TEG via Common Data Link



APG-73 Data Reorganization (1)



APG-73 Data Reorganization (2)



Source

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

Destination

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

DRI Use in CIP APG-73 SAR

DRI Implementations Used

Application	Application	Application
MITRE DRI	Mercury	SKY
MPI *	PAS/DRI	MPICH/DRI

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

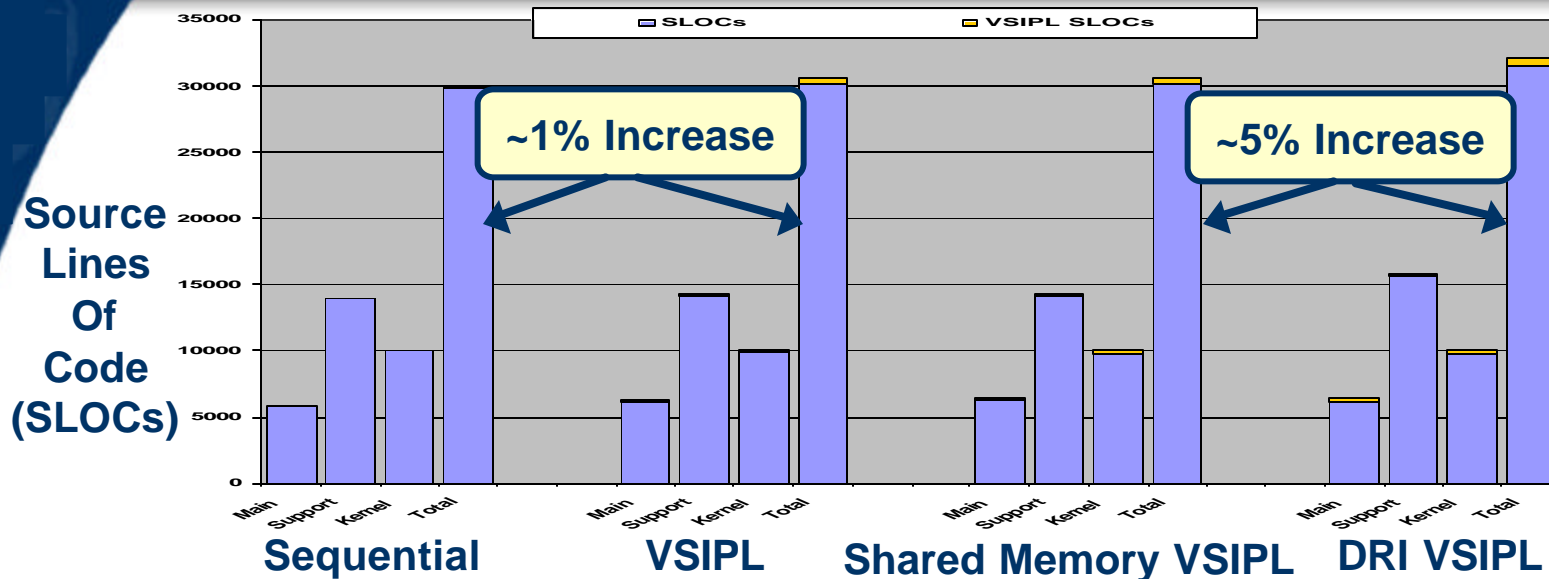
Azimuth compression
Inverse weighting

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

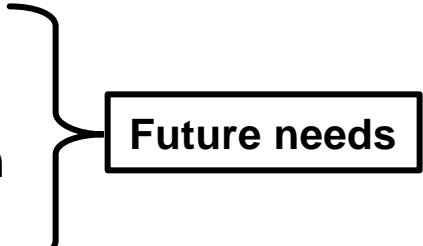
† 3 overlap exchange data reorg cases

† managing interoperation between DRI and VSIP libraries

} 1 for interleaved complex +
2 for split complex data format

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: "HPEC-SI Demonstration: Common Imagery Processor (CIP) Demonstration -- APG73 Image Formation", Brian Sroka, The MITRE Corporation, HPEC 2002
Source: "High Performance Embedded Computing Software Initiative (HPEC-SI)", Dr. Jeremy Kepner, MIT Lincoln Laboratory, June 2003 HPEC-SI meeting



Computer Systems, Inc.
MERCURY

Vendor DRI Status

Mercury Computer Systems, Inc.

MPI Software Technology, Inc.

SKY Computers, Inc.

The Ultimate Performance Machine

Mercury Computer Systems (1/2)

- † **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)**

Mercury Computer Systems (2/2)

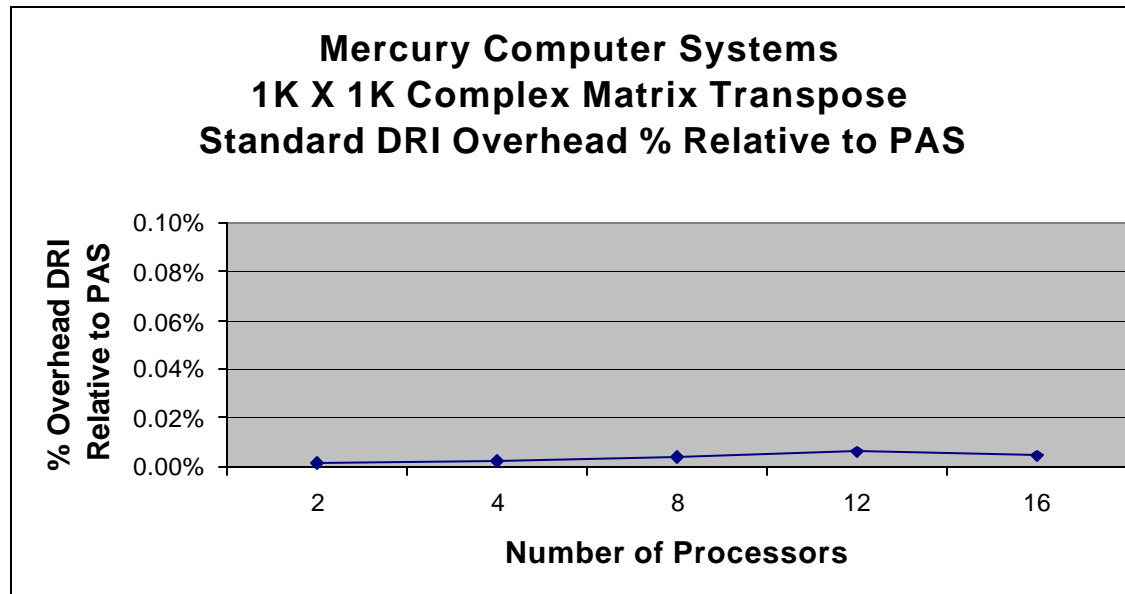
Hybrid use of PAS and DRI APIs

- † **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)

Standard
DRI_Distribution
Object

Standard
DRI_Blockinfo
Object

Built on Existing PAS Performance



**DRI Adds No
Significant
Overhead**

**DRI Achieves PAS
Performance!**

MPI Software Technology, Inc.

- † MPI Software Technology has released its ChaMPlon/Pro (MPI-2.1 product) this spring
- † Work now going on to provide DRI “in MPI clothing” as add-on to ChaMPlon/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)



We take the mess out of message passing.™

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 SMARTpac**
- † **Part of SKY's plans for standards compliance**
- † **Included with MPI library**
- † **Optimized InfiniBand performance**