Data Reorganization and Future Embedded HPC Middleware

Ken Cain, The MITRE Corporation (Presenter)
Anthony Skjellum, MPI Software Technology Inc.
James Lebak, MIT Lincoln Laboratory†
20 September 2000

† This author sponsored by the U.S.Navy 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 United States Air Force.
Report Documentation Page

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE 20 SEP 2000
2. REPORT TYPE
3. DATES COVERED 00-09-2000 to 00-09-2000

4. TITLE AND SUBTITLE
Data Reorganization and Future Embedded HPC Middleware

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MITRE Corporation, 202 Burlington Road, Bedford, MA, 01730-1420

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
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:
   a. REPORT unclassified
   b. ABSTRACT unclassified
   c. THIS PAGE unclassified

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES 14

19a. NAME OF RESPONSIBLE PERSON

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
The Data Reorganization Forum

http://www.data-re.org

Join the mailing list discussion!

Goal: Final specification by June 2001

- Broad community participation includes:
  - FFRDCs and Government/Defense Laboratories
  - Defense integrators
  - Commercial embedded multicomputer vendors
  - Commercial HPC tool vendors

- Examining API’s, algorithms, and application requirements
What Problems Does Data Reorg Try To Solve?
Data Partitioning and Redistribution Issues for Signal/Image Processing (SIP) Applications

- Block partitioning is most common
  - Whole problems stored in 1 memory for performance

- Data redistribution communication is “severe”
  - Prototypical example is matrix transpose in 2DFFT/SAR

![Diagram showing data partitioning and redistribution issues]
Interface Scalability

State of the Art (current standard APIs)
- Programmer manually computes data partitioning
- Programmer manually redistributes data (MPI or MPI/RT)
- Compute using VSIPL

Future Practice (with Data Reorg API)
- Programmer uses high-level partitioning services
- Middleware handles data partitioning details
- Data redistribution with a single high-level call
- Compute using VSIPL

Long-term future: higher-level / integrated / OO ???

Easier to scale programming effort

Hard to scale programming effort to large systems

Long-term future: higher-level / integrated / OO ???
Data Reorg Interface Example

- Application programmer uses DRI to move data
- DRI hides complex data movement from programmer

This example shows a one-dimensional data set stored in block fashion mapped on three nodes being re-distributed in cyclic fashion over two nodes.
Model-Year Portability

Portable software leverages inevitable advances in COTS HPC technology

Defense system lifetimes: long
COTS HPC system lifetimes: short

“Point” solutions specific to a single vendor are long-term cost ineffective

Portable software with high performance is a powerful tool and is the ultimate goal

Increasing overall system performance

Processor support limited or dropped in later products

Increasing Network Performance
(latency, throughput, bisection bandwidth, ...)

Increasing Processing Capability
(μP generations)

Vendor A product line
Vendor B product line
Challenges to Achieving Consensus In A Committee Context
Three Areas of Concern

Operational
- Will this API make it easier to write SIP applications?
- Does API support most common data reorgs for SIP?

Scoped / Prioritized to satisfy most SIP application needs

Research
- Allow integration of research approaches in API implementations
- Enable optimized implementations for a broad class of HPC architectures

Overlap with other APIs
- Common user / library buffers
- VSIPL, MPI, MPI/RT
- Which API allocates data?
Data Reorg
Committee Status
Data Reorg
Objects and Implementation Approaches

CORE
• Uniquely part of Data Reorganization API
• Must be provided in all Data Reorg implementations
• Objects:
  - DRI_Global_Data
  - DRI_Partition
  - DRI_Distribution
  - DRI_Layout
  - DRI_View
  - DRI_Overlap
Data Reorg
Objects and Implementation Approaches

**Standalone**

- Functionality overlaps with other middleware
- Full implementation (without Middleware Adapter) gives a “pure” data reorg programming environment

- Objects:
  - Datatypes: DRI_Dataspec
  - Process Sets: DRI_Group
  - User and Library Memory: DRI_Bufferset
  - Data Transmission Constructs: DRI_Buffer_Id, DRI_Channel
Data Reorg
Objects and Implementation Approaches

Middleware Adapter

- Defines a hybrid interface that leverages supporting middleware
  - MPI
  - MPI/RT
  - Mercury PAS
  - Sky SCL

- Objects:
  - Selected from “Standalone”, depending on supporting middleware
Data Re-org Forum Plan

- Two more official meetings
- Several informal “working” meetings
  - Resolve issues with buffers and buffersets
  - Resolve issues with memory layouts and distributions

- Near-Term activities:
  - Establish CORE and Standalone Interfaces
  - Define MPI Middleware Adapter for Data Reorg
  - Final document detailing ideas and lessons learned

In the long term, the forum feels that a larger effort in this area would have substantial benefits for the high-performance embedded computing community