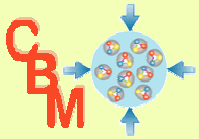


Status InfiniBand software and event buidling

J. Adamczewski, H.G. Essel, S. Linev
EE/GSI

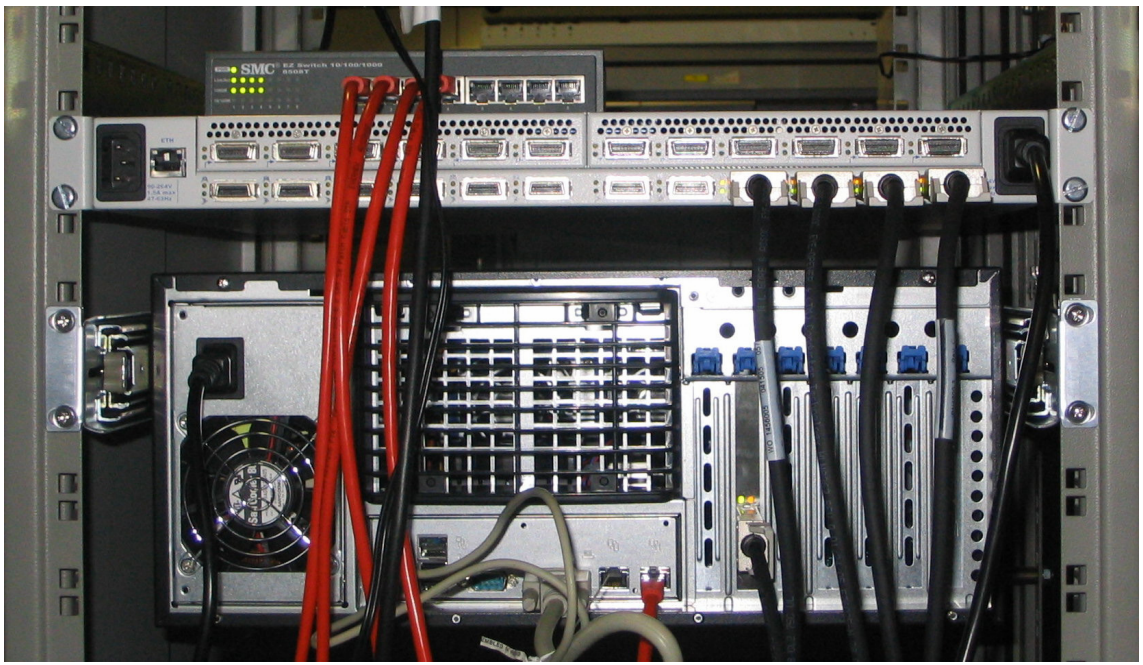
Work supported by EU RP6 project JRA1 FutureDAQ RII3-CT-2004-506078

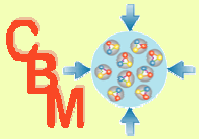
Outline

- New IB-software
- InfiniBand tests in FZK
- First prototype for data transport in DABC

InfiniBand test cluster

- InfiniBand – fast serial interconnect technology, up to 20 GBit/s full duplex
- IB cluster with 4 double Opteron nodes in GSI

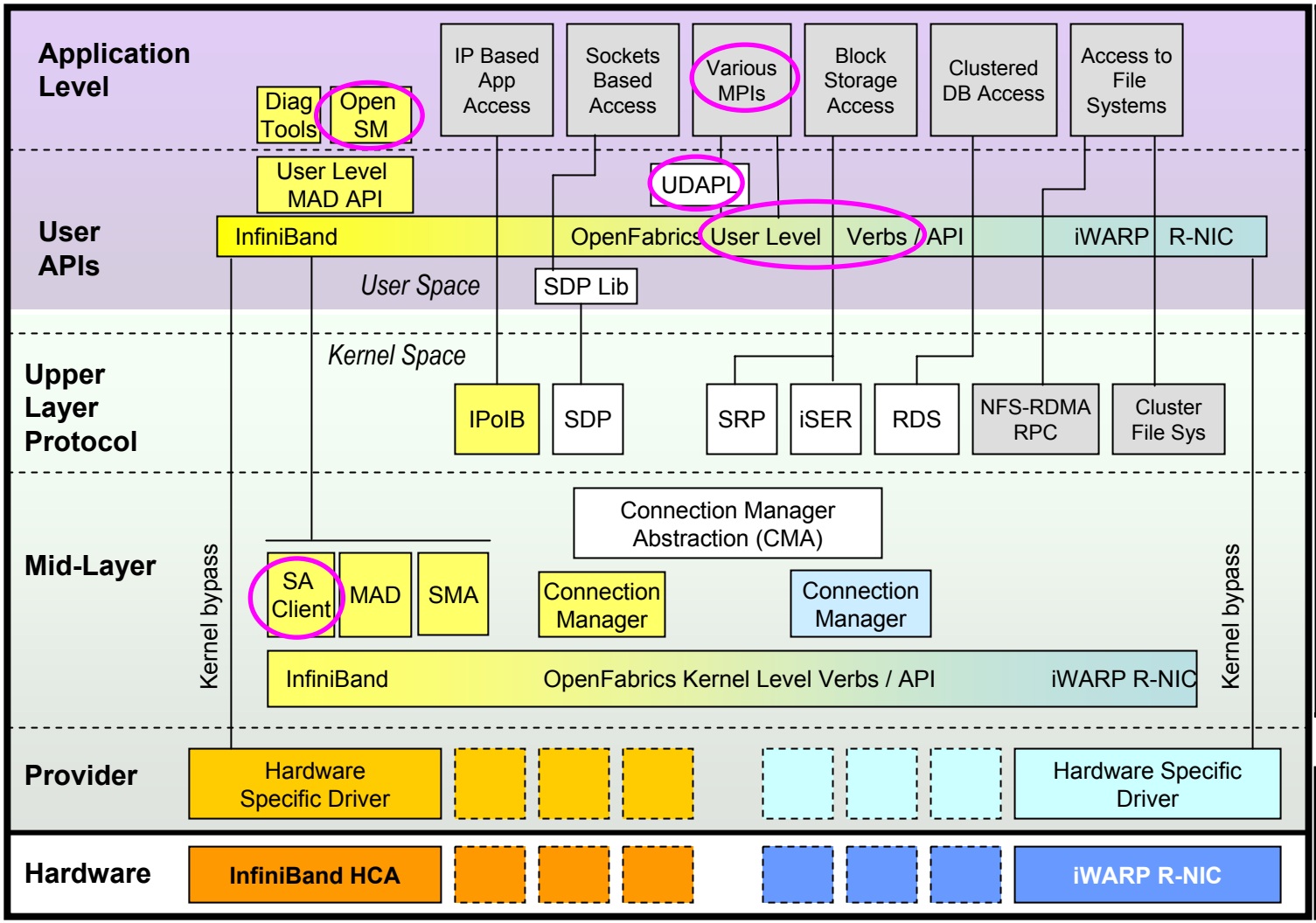




OpenFabrics - new IB software

- One year ago we start with Mellanox *IB Gold 1.8.x* package and using mainly uDAPL.
- *IB Gold* includes most of *OpenIB* components + Mellanox-specific device drivers and administration tools
- *OpenIB* is open source project for IB software development
- Since mid-2006 transformed into *OpenFabrics Alliance (OFED)* www.openfabrics.org, where two technologies are merged together: InfiniBand and iWARP
- Latest *OFED 1.1* release was installed on test cluster – requires newest Linux distribution and Kernel releases.

OpenFabrics Software Stack

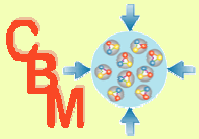


SA	Subnet Administrator
MAD	Management Datagram
SMA	Subnet Manager Agent
PMA	Performance Manager Agent
IPoIB	IP over InfiniBand
SDP	Sockets Direct Protocol
SRP	SCSI RDMA Protocol (Initiator)
iSER	iSCSI RDMA Protocol (Initiator)
RDS	Reliable Datagram Service
UDAPL	User Direct Access Programming Lib
HCA	Host Channel Adapter
R-NIC	RDMA NIC

Key: Common; InfiniBand; iWARP; Apps & Access Methods for using OF Stack

Used components from OFED

- uDAPL – direct access library, was used in our first tests, but has reduced functionality (only peer-to-peer, no multicast) compared to available in InfiniBand network
- VERBS – standard user-level API for arbitrary InfiniBand devices, provides full access to InfiniBand functionality
- SA – subnet administrator client, allows creation/management of multicast groups
- OpenSM – InfiniBand compliant Subnet Manager, involved in configuration and controls of the network



VERBS versus DAPL

- Both have very similar functionality and API:
  - memory, queues, completion events;
  - message and RDMA transport.

- But, VERBS provides extra functionality:
  - reliable/unreliable data transfer;
  - multicast support.

- We decide to switch to VERBS & OFED while:
  - it supports full InfiniBand functionality;
  - it is new official development line for Mellanox products.

- The only **BIG** verbs problem – lack of good documentation.

VERBS tests

- Test application for various traffic patterns via InfiniBand with VERBS API
- Using OpenSM and SA functionality for multicast group creation

Results:

Traffic kind	Peer-to-peer	Multicast
Unidirectional (3x1)	985 MB/s	-
Bidirectional (4x4)	951 MB/s	-
Multicast (1x4)	-	625 MB/s with ~0.002% losts
B-Net (4x4) emulation	840 MB/s	40 MB/s with 0% losts

MPI evaluation

- MPI – Message Passing Interface, was designed for high performance on both massively parallel machines and on workstation clusters

- MVAPICH – MPI & MPI2 over InfiniBand project. Supports:
  - non-blocking send/receive operation
  - true hardware-based multicast, but only with blocking API

- Tests of data throughput and multicast performance were done. Good results for big (larger than 32K) packets, but difficulty to combine normal and multicast traffic.

- Can be as option in DAQ system, while it is supported on majority of modern massively parallel machines, where different interconnect technologies are used.

Forschungszentrum Karlsruhe tests

- In FZK ~90 nodes Opteron cluster with SilverStorm InfinIO9100 switch http://www.campusgrid.de/en/opteron.html installed:
  - 30 nodes with PCI-X HCAs
  - 32 nodes with PCI-e HCAs (similar to GSI nodes)
  - 32 nodes with PCI-e DDR HCAs (new, in testing now)
- On old nodes only MPI & DAPL are available, on new nodes OFED 1.0 can be optionally installed
- First tests with DDR HCAs gives maximal unidirectional data rate of 1.15 GB/s

- In next week tests with 24-nodes cluster are planned:
  - scheduled versus chaotic transfer
  - multicast scalability & reliability
  - B-Net emulation performance

goal: „Data Acquisition Backbone Core“ DABC

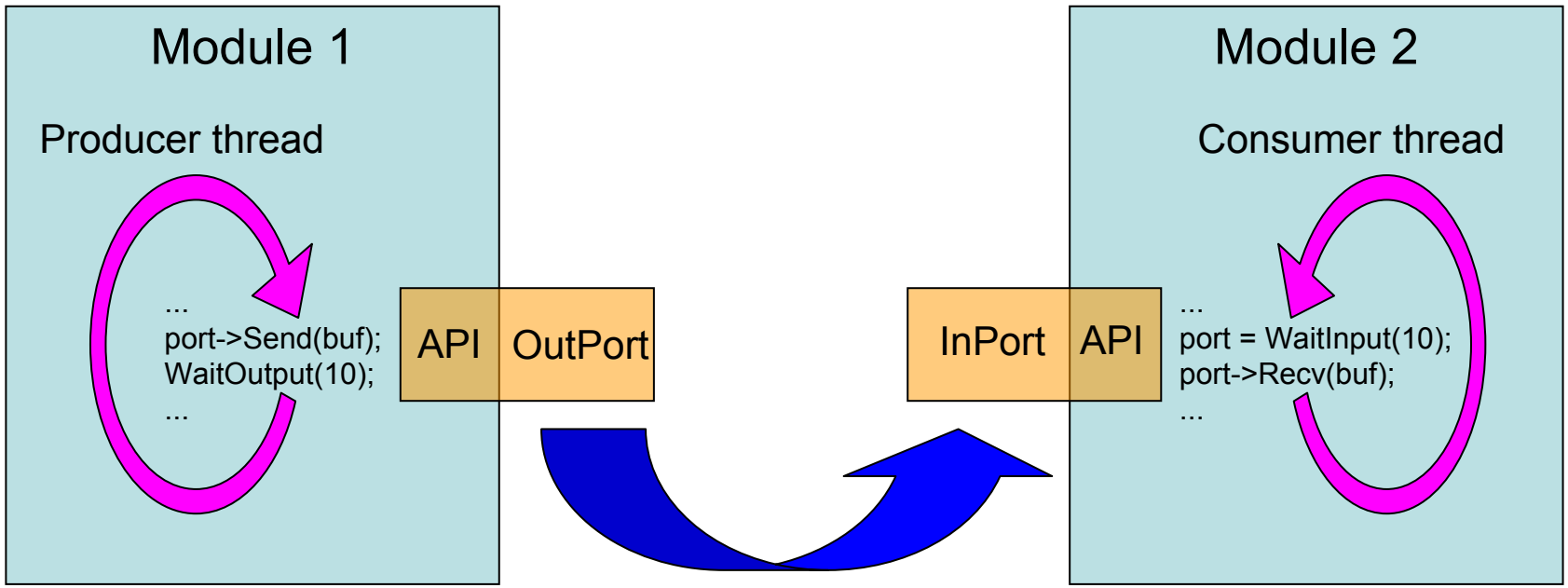
- Modular architecture
- Data transport management
- Configuration of multiple nodes
- Controls, monitoring, message logging
- Error handling, failure recovery
- Hardware driver integration
- ...

Data transport in DABC

- Major tasks and requirements:
  - connection management between distributed system components;
  - support of zero-copy non-blocking data transfer;
  - best possible performance;
  - buffers management;
  - same API for different transports.

- Data-flow approach:
  - data-processing separated in modules;
  - communications from/to modules via port objects;
  - exchanging data between modules ports via transport layer.

Generic data-flow chart



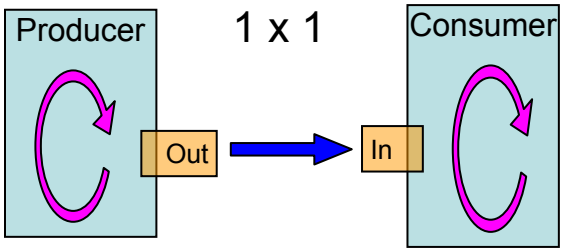
Transport of data from output of one module to input of another module

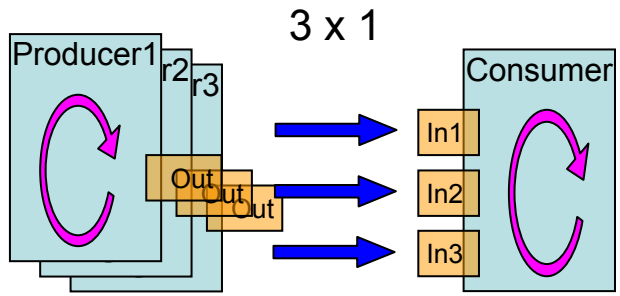
First implementation

- *dabc::TModule* class:
  - uses pthreads for workloops, conditions for synchronization;
  - gateway via XDAQ to controlling/configuration framework.
- *dabc::TPort* class:
  - provides input/output communication API;
  - pear-to-pear and network-kind transport;
  - access to transport functionality via abstract *dabc::TTranportInput* and *dabc::TTransportOutput* interfaces.
- Three concrete implementation for transport:
  - *dabc::TThreadTransport* for communication between modules (threads) in scope of single application;
  - *dabc::TSocketTransport* for communication via Unix sockets;
  - *dabc::TVerbsTransport* for InfiniBand communication.
- XDAQ memory managements classes are used: xdaq::mem::Pool, xdaq::mem:Buffer, xdaq::mem::Reference.

Data-flow benchmarking

- Two simple test modules were implemented:
  - TProducer delivers buffers of specified size;
  - TConsumer with configurable numbers of input ports.
- Two configurations:





- Results of test on InfiniBand cluster:

Transport:	1 x 1	3 x 1
Thread	~7 µs/oper, CPU 100%	~8 µs/oper, CPU 100%
Socket (Gb Ethernet)	117 MB/s, CPU ~20%	117 MB/s, CPU ~40%
Socket (IPoIB)	412 MB/s, CPU ~60%	522 MB/s, CPU 100%
VERBS	950 MB/s, CPU ~7%	950 MB/s, CPU ~10%

Conclusion & plans

- Status:
  - Moving from DAPL to OFED VERBS as main development line
  - DAPL and MPI still can be considered as option
  - First working prototype of data-flow framework in DABC

- Plans:
  - Tests on bigger InfiniBand cluster in FZK
  - Deeper investigation of InfiniBand multicast
  - Further development of data-flow components in DABC
  - Development of B-Net prototype and integration with XDAQ framework (see talk of J.Adamczewski)