OPENFABRICS ALLIANCE

Extending DAPL for Collectives

Author: Arlin Davis
Date: 4/4/2011

Agenda

- DAPL Framework for extendibility
- MPI collective extensions
  - member addressing
  - group creation
  - collective operations
- DAPL Provider for FCA 2.0

Framework for extendibility

- v2.0 introduced generalized extension interface
  - dat_extension_op(handle, op, va_list)
    - #include dat2/dat_xx_extensions.h
  - dat_ia_query() for supported extended operations
  - events, handles, data ops, service types, all extendable

- Extended transports/features
  - iWARP (iw): socket service point
  - Infiniband (ib): atomics, immed, ud, collectives

MPI collective extensions

- MPI compatible interface, hardware agnostic
  - all members in group must call collective operations
  - collective call interface can be sync or async (evd)
  - member address info opaque, returned by provider
    - MPI needs to collect/exchange on all ranks for group creation
  - data is user-virtual address space, provider handles any registration if necessary

MPI collective extensions

- dat_ib_collective_create_member(), sync
  - provider specific address info returned as ptr/size
  - ia_handle is reference to device interface
  - progress_fn called by provider when blocking too long

```
#define dat_ib_collective_create_member(ia, progress_fn, mbr, mbr_size) \
    dat_extension_op(\
      IN  (DAT_IA_HANDLE) (ia), \
      IN  (DAT_IB_OP) DAT_IB_COLLECTIVE_CREATE_MEMBER_OP, \
      IN  (void *) (progress_fn), \
      OUT (DAT_IB_COLLECTIVE_MEMBER *) (mbr), \
      OUT (DAT_UINT32 *) (mbr_size))
```

MPI collective extensions

- dat_ib_collective_create_group() , async
  - MPI creates new communicator before calling
  - called by all members in the group
  - members address info, self, group id, group_info
    - info = local, external, intranode, internode info for provider

```
#define dat_ib_collective_create_group(mbrs, size, self, id, info, evd, pd, context) \
    dat_extension_op(\
      IN (DAT_EVD_HANDLE) (evd), \
      IN (DAT_IB_OP) DAT_IB_COLLECTIVE_CREATE_GROUP_OP, \
      IN (DAT_IB_COLLECTIVE_MEMBER *) (mbrs), \
      IN (DAT_COUNT) (size), IN (DAT_IB_COLLECTIVE_RANK) (self), \
      IN (DAT_IB_COLLECTIVE_ID) (id), IN (DAT_IB_COLLECTIVE_GROUP *) (info), \
      IN (DAT_PZ_HANDLE) (pd), IN (DAT_CONTEXT) (user_context))
```

MPI collective extensions

- Collective operations supported
  - Bcast, Barrier, Scatter, Scatterv,
  - Gather, Gatherv, Allgather, Allgatherv
  - Reduce, Allreduce, Reduce-Scatter
  - Alltoall, Alltoallv

MPI_Bcast (void *buffer, int count, MPI_Datatype datatype, int root, MPI_Comm comm)

dat_ib_collective_bcast(
DAT_HANDLE coll_handle,
void ***buffer**, int **byte_count**, int **root**,
DAT_CONTEXT context, int comp_flags)

DAPL provider for FCA 2.0

- Porting/testing underway with FCA 2.0
  - Bcast, Barrier, Scatter, Scatterv,
  - Gather, Gatherv, Allgather, Allgatherv
  - Reduce, Allreduce, Reduce-Scatter
  - Alltoall, Alltoallv

Work in progress

- Provider support for FCA 2.0:

  Targeted for OFED 1.6 (config option, --enable-coll-type)

- Beta documentation

  http://www.openfabrics.org/downloads/dapl/documentation/dat_collective_preview.pdf

- DAT consortium review/acceptance
  - Complete IB collective extension addendum for DAT 2.0 specification

Please review beta document and send comments to arlin.r.davis@intel.com