GASNet

OFI BoF SC17

Lawrence Berkeley National Laboratory: Paul Hargrove, Dan Bonachea

Intel Corporation: Erik Paulson

gasnet.lbl.gov

What is GASNet?

- Middleware networking API meant to enable PGAS languages
- Developed by Lawrence Berkeley National Labs
- Used by: Three UPC compilers, Chapel, Legion, UPC++, Co-Array Fortran, OpenSHMEM
- Network implementations (conduits) use a layered approach
  - Core Active Messaging layer (directly implemented by all conduits)
  - Extended API with richer operations (conduits selectively specialize)
- Native implementation for most networks used in HPC
  - Cray Gemini/Aries, InfiniBand verbs+mxm, BlueGene/Q, OmniPath
  - Also several portable implementations, including over OFI (OPA, GNI, sockets)

Highlights

- Extended API (point to point puts/gets) match OFI RMA semantics
  - The code path is very simple. For example:

```
void gasnetc_rdma_put(gasnet_node_t dest, void *dest_addr, void *src_addr,
size_t nbytes, gasnetc_ofi_op_ctxt_t *ctxt_ptr)
{
  int ret = FI_SUCCESS;
  ((gasnetc_ofi_op_ctxt_t *)ctxt_ptr)->callback = gasnetc_ofi_handle_rdma;

  fi_write(gasnetc_ofi_rdma_epfd, src_addr, nbytes, NULL, dest, dest_addr,
    0ULL, ctxt_ptr);
  if (FI_SUCCESS != ret)
                          gasneti_fatalerror("fi_write for normal message failed: %d\n", ret);
  gasnetc_paratomic_increment(&pending_rdma,0);
}
```


Highlights cont.

- GASNet's recommended polling model maps nicely to FI_PROGRESS_MANUAL
- Remote-access memory segment registration is simple via fi_mr_reg()
  - FI_MR_SCALABLE mode easily supports GASNET_SEGMENT_EVERYTHING by registering an offset from address 0 to UINT64_MAX
- fi_inject functions optimize the common PGAS case of small messages.
  - Removes the need to poll for local completion on AM injection

Lowlights

- Managing what features various providers support is a pain
  - Especially when “required” features like FI_THREAD_SAFE are missing
  - Requires macros and configuration tricks
- Semantic mismatch for small, non-blocking puts
  - GASNet exposes notification of both local and remote completion
  - OFI operations can support either model, but not both.
  - Solution: A 3-prong approach of using FI_INJECT, bounce buffers, and blocking for remote completion.

Lowlights continued

- Active messaging support
  - Deadlock avoidance requires use of two OFI endpoints for virtualization
    - one for AM requests and one for AM replies
  - Increases time spent polling for completions
  - Not an OFI specific problem, but some providers could be able to support active message channel isolation more efficiently
  - Many AMs carry 4 bytes of empty padding on the wire
    - just to maintain 8-byte alignment in MULTIRECV buffer at target

Questions?

- Please send inquiries to gasnet-users@lbl.gov
- More info: http://gasnet.lbl.gov

BACKUP

OFI provider requirements

- Endpoint type: EP_RDM (Reliable Datagram)
- Capabilities: FI_RMA and FI_MSG
- Secondary Capabilities: FI_MULTI_RECV, FI_RM_ENABLED, FI_AV_TABLE
- Memory Registration Mode: FI_MR_SCALABLE (preferred) or FI_MR_BASIC
  - There is currently no support for providers that require FI_LOCAL_MR
- Threading mode: FI_THREAD_SAFE and/or FI_THREAD_DOMAIN
  - In GASNET_{SEQ,PARSYNC} mode, all providers use FI_THREAD_DOMAIN as only one thread makes calls into the GASNet library.
  - In GASNET_PAR mode, FI_THREAD_DOMAIN is used only for the psm2 provider which currently does not support FI_THREAD_SAFE. All other providers use FI_THREAD_SAFE.

Possible improvements

- Scalable endpoints
  - Would reduce address vector size by ⅔ (currently 3 EPs are used).
  - Could be used to implement implicit access region synchronization using OFI counters (currently implemented in software)
- Vectored/Indexed/Strided operations may use SGL versions of OFI functions to reduce function calls.
  - Requires the provider to support an adequately sized SGL