

Proposal to the GASPI Specification

Inclusion of `gaspi_read_list_notify`

Christian Simmendinger

November 30, 2016

0.0.1 `gaspi_read_list_notify`

The `gaspi_read_list_notify` variant extends the simple `gaspi_read_list` with a notification on the local side. This applies to communication patterns that require tighter synchronisation on data movement. The local receiver of the data is notified when the read list is finished and can verify this through the procedure `gaspi_waitsome`. It is an *asynchronous non-local time-based blocking* procedure.

```
GASPI_READ_LIST_NOTIFY ( num
                        , segment_id_local[num]
                        , offset_local[num]
                        , rank
                        , segment_id_remote[num]
                        , offset_remote[num]
                        , size[num]
                        , notification_id_local
                        , queue
                        , timeout )
```

Parameter:

- (in) num:* the number of elements to read
- (in) segment_id_local[num]:* the local segment ID's to write to
- (in) offset_local[num]:* the local offsets to write to
- (in) rank:* the remote rank to read from
- (in) segment_id_remote[num]:* the remote segment ID's to read from
- (in) offset_remote[num]:* the remote offsets in bytes to read from
- (in) size[num]:* the size of the data elements to read
- (in) notification_id:* the local notification ID
- (in) queue:* the queue to use
- (in) timeout:* the timeout

```

gaspi_return_t
gaspi_read_list_notify ( gaspi_number_t num
                        , gaspi_segment_id_t *segment_id_local
                        , gaspi_offset_t *offset_local
                        , gaspi_rank_t rank
                        , gaspi_segment_id_t *segment_id_remote
                        , gaspi_offset_t *offset_remote
                        , gaspi_size_t *size
                        , gaspi_notification_id_t notification_id
                        , gaspi_queue_id_t queue
                        , gaspi_timeout_t timeout )

```

```

function gaspi_read_list_notify(num,segment_id_local,&
&      offset_local,rank,&
&      segment_id_remote, offset_remote,&
&      size,notification_id,queue,&
&      timeout_ms) &
&      result( res ) bind(C, name="gaspi_read_notify")

```

```

integer(gaspi_number_t), value :: num
type(c_ptr), value :: segment_id_local
type(c_ptr), value :: offset_local
integer(gaspi_rank_t), value :: rank
type(c_ptr), value :: segment_id_remote
type(c_ptr), value :: offset_remote
type(c_ptr), value :: size
integer(gaspi_notification_id_t), value :: notification_id
integer(gaspi_queue_id_t), value :: queue
integer(gaspi_timeout_t), value :: timeout_ms
integer(gaspi_return_t) :: res
end function gaspi_read_notify

```

Execution phase:

Working

Return values:

GASPI_SUCCESS: operation has returned successfully

GASPI_TIMEOUT: operation has run into a timeout

GASPI_ERROR: operation has finished with an error

User advice: In contrast to the procedure `gaspi_write_list_notify`, the notification in the procedure `gaspi_read_list_notify` carries the (fixed) notification value of 1. Similar to the procedure `gaspi_write_list_notify` a call to `gaspi_read_list_notify` only guarantees ordering with respect to the data bundled in this communication and the given notification. Specifically there are no ordering guarantees to other read operations. For this latter functionality a call to the `gaspi_wait` procedure is required.

Implementor advice: The procedure is not semantically equivalent to a call to `gaspi_read_list` and a subsequent call of `gaspi_notify`, since the latter aims at remote completion rather than local completion. Also this call does not enforce an ordering relative to other read operations. ┘

1 Needed Resources

- none.

2 Additional (necessary) Changes to the Standard

- 8.3.3

For the procedures with notification, `gaspi_notify` and the extended functions `gaspi_write_notify`, `gaspi_write_list_notify`, `gaspi_read_notify` and `gaspi_read_list_notify`, the function `gaspi_notify_waitsome` is the correspondent wait procedure for the notified receiver side (which is remote for the functions `gaspi_notify`, `gaspi_write_notify`, `gaspi_write_list_notify` and local for the functions `gaspi_read_notify` and `gaspi_read_list_notify`).

- additional user advice

User advice: One scenario for the usage of `gaspi_notify_waitsome` inspecting only one notification is the following: The local side posts a `gaspi_read_notify` or `gaspi_read_list_notify` call. GASPI guarantees, that if the notification has arrived on the local process, the posted read request carrying the work load of the respective function has arrived as well. ┘