GASPI-Tutorial MPI Interoperability

1 Introduction

GASPI is designed to be interoperabe with MPI, i. e. both MPI and GASPI can be used within the same program. This allows developers to incrementally port existing MPI applications, or to continue to rely on existing well tested code for non-performance critical parts of the application. However, to allow both programming systems to work together a few restrictions apply. We discuss the restrictions in detail in the next section – the rule of thumb is that the developer must make sure that MPI and GASPI do not communicate at the same time.

2 Restrictions

GASPI 1.0 defines a structure on how MPI and GASPI function calls must be interleaved within the same application, as shown below:

Listing 1: GASPI-MPI structure.

```
1  mpi_startup;
2
3  /* MPI part, no ongoing GASPI communication... */
4
5  /* ...finish all ongoing MPI communication */
6
7  mpi_barrier;
8
9  /* no ongoing MPI communication */
10
11 gaspi_proc_init;
12
13 while (!done) {
14
15   /* GASPI part, no ongoing MPI communication... */
16
17   /* ...finish all ongoing GASPI communication */
18
19   gaspi_barrier;
20
21   /* MPI part, no ongoing GASPI communication... */
22
23   /* ...finish all ongoing MPI communication */
24
25   mpi_barrier;
26 }
27
28 gaspi_proc_term;
29
30 /* MPI part, no ongoing GASPI communication */
31
32 mpi_shutdown;
```

Formally speaking, the application runtime is divided into several epochs that start and end with a barrier sychronization of all processes (and threads). Communication within a epoch can only be done with either GASPI or MPI, but not both. This restriction includes possible nonblocking or asynchronous communication calls, so the developer must make sure that all previously started communication requests are completed before the next epoch is started. In practise ending an MPI epoch may include an `MPI_Waitall()` and ending a GASPI epoch may require a `gaspi_wait()` call and to manually complete all timed out communications.

3 Starting a GASPI-MPI application

The GASPI specification does not specify how an application using both MPI and GASPI should be started, so starting a hybrid MPI/GASPI program is implementation specific. GASPI implementations may provide an appropriated startscript.