MPI and Fault Tolerance: concept and limitations of the current specification

Edgar Gabriel

High Performance Computing Center Stuttgart (HLRS)
gabriel@hlrs.de
Outline

• Motivation
• MPI-1 and error handling
• MPI-2 dynamic communicators
• Fault-tolerant manager-worker frameworks
  – Concept
  – Status with current MPI libraries

• Summary
Motivation

- Process failures happen –
  - and are getting more probable with increasing number of processes

- Checkpoint-Restart mechanisms work
  - but also have their limitations

Is an extension of MPI necessary to handle process failures?
MPI – 1 error handling

- Static group of processes - MPI_COMM_WORLD
- An error handler is attached to each communicator
  - MPI_ERRORS_ARE_FATAL: abort application on error
  - MPI_ERRORS_RETURN: return control to user application
- MPI_Abort is allowed to ignore communicator argument
  - All MPI-1 implementations do ignore the communicator argument.
MPI-2 dynamic communicators

- MPI-2 enables spawning of new processes
- MPI-2 enables connecting two already running applications
- Failure in one application might affect all connected applications

"As in MPI-1, it [MPI_Abort] may abort all processes in MPI_COMM_WORLD (ignoring its comm argument). Additionally, it may abort connected processes as well, although it makes best attempt to abort only the processes in comm."

weak statement
Disconnected processes

- Connected processes can disconnect using MPI_Comm_disconnect
- Parent and child processes might disconnect

"MPI_Abort does not abort independent processes"

- strong statement

- It is not possible to disconnect processes sharing the same MPI_COMM_WORLD
Manager – worker framework 1 (I)

Manager

Manager

Worker 1

Worker 2

Worker 3

MPI_Comm_spawn()
Manager – worker framework 1 (II)

Manager

Worker 1

Worker 2

Worker 3

MPI_Comm_spawn()
Relevant questions

1. Does manager survive the failure of worker processes?
2. What happens if manager tries to send a message to a failed worker process?
4. Can manager re-spawn worker processes after an error occurred?
5. Can manager communicate internally after the failing of worker process(es)?
### Status of current implementations

<table>
<thead>
<tr>
<th>Feature Description</th>
<th>LAM/MPI</th>
<th>MPICH2-0.97b</th>
<th>MPI/SX</th>
<th>Hitachi MPI</th>
<th>SUN-MPI</th>
<th>Open MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Manager survives failing worker process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Manager can handle sending a msg. to failed processes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Manager can spawn new worker processes</td>
<td>(◯)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Manager can communicate internally after worker failed</td>
<td>(◯)</td>
<td></td>
<td></td>
<td></td>
<td>(◯)</td>
<td>(◯)</td>
</tr>
</tbody>
</table>

Edgar Gabriel
High Performance Computing Center Stuttgart
Manager – worker framework 2 (II)

Manager

Worker 1

MPI_Comm_spawn()

MPI_Comm_disconnect()

Worker 2

MPI_Comm_spawn()

MPI_Comm_disconnect()

Worker 3

MPI_Comm_spawn()

MPI_Comm_disconnect()
Manager – worker framework 2 (I)

Manager

Worker 1

Worker 2

Worker 3

MPI_Comm_connect() MPI_Comm_disconnect() MPI_Send() MPI_Recv()
Problems with second framework

- Manager might still be teared down by failing worker processes while being connected
- MPI_Comm_connect/accept has to be able to discover failed worker process
- Slow – you have to reconnect to worker for every single message
Can we write an FT-application based on MPI-2?

- Under optimal circumstances: yes
  - If your MPI implementation supports the *weak statement*

- Problems
  - Still not portable – since MPI implementations don’t have to support the *weak statement*
  - No concept on how to discover process failures (e.g. a unique error code)
Summary

• MPI-2 offers new possibilities with dynamic communicators for ft-applications
• Error handling of dynamically connected processes has a weak statement on process failures and a strong statement
  – Strong statement does unfortunately not help in most ft-scenarios