MPI as a programming model for High-Performance Reconfigurable Computers

ArchES Computing

Manuel Saldaña

SciNet SNUG – October 12, 2011
Toronto, Canada
Overview

- FPGAs and High-Performance Reconfigurable Computers

- What is ArchES-MPI?
  - Programming model
  - Message-Passing Engine
  - Functionality
  - Platforms
  - Use cases

- Future Work
High-Performance Reconfigurable Computer Model

- One or more interconnected Hosts
- One or more General Purpose Processors (X86)
- One or more FPGA Clusters
- One or more accelerators per FPGA
FPGAs in High-Performance Computing

- FPGAs as accelerators => Co-processors to CPUs
  - CRAY
  - SGI
  - SRC
  - DRC
  - Xtreme Data
  - Convey
  - ...

- The main obstacle has been the programming model!!
Challenges

- Some FPGA programming models try to automatically:
  - Extract parallelism
    - communication,
    - synchronization
    - load balance
    - algorithm itself
  
- Generate Hardware
  - signal timing,
  - low-level structures: registers, logic gates & LUTs
  - Physical placement of components
At ArchES Computing, we know both worlds and we leverage that knowledge to create High-performance Systems.
Why ArchES-MPI?

- Subset implementation of the MPI standard
  - API syntax and semantics
- Optimization of communications
- Widely used in the HPC world
  - distributed memory machines
- Provides portability
  - by adding layers of abstraction
- Isolates software from hardware changes
- Vast amount of documentation and examples available
Why ArchES-MPI?

- Ease of use: reduce the number of APIs
  - Inter-host communication (sockets, MPI)
  - X86-FPGA communication (vendor-specific)
  - X86-X86 Intra-host communication (pthreads, OpenMP)
  - For embedded processors (custom)

- Use one single API: MPI
  - ArchES-MPI is an implementation to achieve this
What is ArchES-MPI?

- Parallelism is explicitly stated
- Get the parallel algorithm right!
What is ArchES-MPI?

- Test and debug parallel implementation

Typical MPI cluster

- Communicating processes
- Computing Elements
What is ArchES-MPI?

- Gradually introduce accelerators, which are treated as peers to processors.

Communicating processes

Computing Elements

Hardware Accelerator

Hardware Accelerator

ArchES-MPI
Software-Hardware partitioning

- MPI as a common abstraction ("Language") between software and hardware experts
ArchES-MPI is software and hardware

- Adds software and hardware middle-ware layers
- Abstracts low-level communication details
- Makes applications more portable
The MPE provides the equivalent to MPI_Send and MPI_Recv to hardware engines.

The MPI FSM can be easily modeled from the MPI C code:

```c
main() {
    ...
    MPI_Recv()
    Compute()
    MPI_Send()
    ...
}
```

The network is connected to the MPE through a command FIFO and a data FIFO. Status and control signals such as busy, done, enable, etc., are also utilized.
void main (int argc, char **argv) {
    int x, my_rank, size;
    MPI_Init(…);
    MPI_Comm_rank(…,&my_rank);
    MPI_Comm_size(…, &size);
    if ( my_rank == 0 ) {
        x = 1;
        MPI_Send(&x,1,MPI_INT,1,…);
        MPI_Recv(&x,1,MPI_INT,size-1,…);
    }
    else if (my_rank == size-1) {
        MPI_Recv(&x,1,MPI_INT,my_rank-1,…);
        x++;
        MPI_Send(&x,1,MPI_INT,0,…);
    }
    else {
        MPI_Recv(&x,1,MPI_INT,my_rank-1,…);
        x++;
        MPI_Send(&x,1,MPI_INT,my_rank+1,…);
    }
    MPI_Finalize();
}
Supported MPI Functions

- Point-to-Point
  - Blocking
    - MPI_Send
    - MPI_Recv
  - Non-Blocking
    - MPI_Isend
    - MPI_Irecv
    - MPI_Wait/MPI_Test

- One-side-communications
  - MPI_Alloc_mem
  - MPI_Put/MPI_Get

- Collective Operations
  - MPI_Barrier
  - MPI_Bcast
  - MPI_Gather/MPI_Scatter
  - MPI_Reduce
  - MPI_Allreduce

- Miscellaneous
  - MPI_Init
  - MPI_Finalize
  - MPI_Comm_Rank
  - MPI_Comm_Size
  - MPI_Wtime
ArchES-MPI Framework

**Co-Simulation**
(Rapid prototyping and development)

**Profiling**
(Performance analysis, e.g. Jumpshot)

**Partial Reconfiguration**
(dynamic MPI process creation and Generic platforms)

**MPI-to-HDL**
(easier for sw developers)

**NoC infrastructure and Portability**
(e.g. FSB, QPI, PCIe, & Hardware support for Broadcast and Reduce)

**Libraries and applications**
(Third-party development, e.g., Xilinx, UT Austin, UofT)
Profiling with Jumpshot
(Daniel Nunes @ UofT)

- Well-known tool
- Extracts MPI protocol states from the MPE
- Profile just like in Software
- Works only for embedded processors and hardware engines
FSB-based platform

ArchES software and infrastructure brings this machine to life!
- Up to 15 FPGAs can be placed in the server divided across 3 stacks
- Very tight coupling between all FPGAs, system CPU and system memory
Use case (1/3)

University of Texas at Austin: FAST, A Processor Architecture Simulator
“Unlike physical world, computers grow in complexity faster than they get faster”
University of Toronto: A Multi-FPGA Architecture for Stochastic Restricted Boltzmann Machines (Neural Networks)

- BEE2 and BEE3 platforms
- 145X Speedup compared to single CPU
  - 3.13 billion connection-updates-per-second
- Embedded PowerPC processor
- DataFlow communication requirements
  - Simultaneous message reception from different sources
  - Full-duplex
  - Overlap communication and computation
The Hospital for Sick Children (Structural Biology and Biochemistry at UofT) – Molecular Dynamics

\[
U = \sum_i k_i \left[ 1 + \cos \left( n_i \varphi_i - \gamma_i \right) \right], n_i \neq 0 \\
U = \sum_i k_i (\theta_i - \theta_a)^2 \\
U_b = \sum_i k_i (r_i - r_a)^2 \\
V(r) = 4\varepsilon \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \\
U = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} q_i q_j \frac{1}{r_{ij} + \varepsilon} \\
O(n^2) \\
O(n)
\]

- MPMD – A Mix of different SW processes and different HW accelerators
- Software for processors is plain MPI on C++
- Nallatech FSB FPGA Accelerated Platform
Future directions

- QPI Modules
- PCIe Gen2-X8 or better
- Zynq devices (2 ARM cores + FPGA fabric)
- Embedded, Scientific and Data Centre applications will drive future developments
Questions?

Thank you!

ArchES Computing

msaldana@archescomputing.com
www.archescomputing.com