Integrating OmpSs@FPGA within Eclipse

Presentation for EclipseCon 2019

Ruben Cano-Díaz and Xavier Martorell

Barcelona Supercomputing Center

The LEGaTO project has received funding from the European Union’s Horizon 2020 research and innovation programme under the grant agreement No 780681
Porting algorithms to IP cores in FPGA

```c
#define BS 128

void matrix_multiply(float a[BS][BS], float b[BS][BS], float c[BS][BS])
{
    // matrix multiplication of two A, B matrices, to accumulate the result on C
    for (int ia = 0; ia < BS; ++ia)
        for (int ib = 0; ib < BS; ++ib) {
            float sum = 0;
            for (int ic = 0; ic < BS; ++ic)
                sum += a[ia][ic] * b[ic][ib];
            c[ia][ib] += sum;
        }
}
```

```c
... for (i=0; i<NB; i++)
    for (j=0; j<NB; j++)
        for (k=0; k<NB; k++)
            matrix_multiply(A[i][k], B[k][j], C[i][j]);
...```
#define BS 128

void matrix_multiply(float a[BS][BS], float b[BS][BS], float c[BS][BS])
{
  #pragma HLS inline
  int const FACTOR = BS/2;
  #pragma HLS array_partition variable=a block factor=FACTOR dim=2
  #pragma HLS array_partition variable=b block factor=FACTOR dim=1
  // matrix multiplication of a A*B matrix
  for (int ia = 0; ia < BS; ++ia)
    for (int ib = 0; ib < BS; ++ib)
      {  
        float sum = 0;
        for (int id = 0; id < BS; ++id)
          sum += a[ia][id] * b[id][ib];
        c[ia][ib] += sum;
      }
}
Outline

• Introduction to OmpSs@FPGA
• OmpSs integration within Eclipse / Eclipse Che
  o Autocompletion of directives
  o OmpSs toolchain on Docker Containers
• OmpSs@FPGA toolchain
• Evaluation & tracing
• Conclusions & future work
Autocompletion of directives

OmpSs/ OpenMP support in Eclipse

• Support for OmpSs and OpenMP development in eclipse
  • Plugins developed
  • Support for most of the programming models directives and clauses
  • Including small help descriptions
  • Based on context, with autocompletion

• Integration of the compilation environment
  • Eclipse Che
Autocompletion of directives
Compilation for the FPGA

Incorporated to Eclipse Che

• Eclipse Che is the next version of the IDE environment
  • Easy administration on distributed environments
  • Suitable for local installation and server installation
  • Redesigned and per-stack customizable interface
  • The compilation is done inside a Docker container

• Our Containers
  • OmpSs compilation
  • With links to FPGA vendor tools
    • Xilinx Vivado
  • Automatic installation from git with a Python script
Compilation for the FPGA

Incorporated to Eclipse Che

Contains the per-platform configuration for our Docker image.
All workspaces use our images and our stack configuration.

No configuration needed to compile for either platform thanks to our stack.

There can be any number of workspaces.

Each workspace contains its private projects.

The Vivado installation is shared between all workspaces.
Compilation for the FPGA

Incorporated to Eclipse Che

• Docker container
• Contains preconfigured OmpSs@fpga tools
  • Mercurium, Nanos, Xtasks, Xdma, Extrae, Papi, Cross Compiler
• Increases productivity and helps avoid configuration errors
Compilation for the FPGA

Workspaces

- Creation of a workspace
  - Custom stack selection
Compilation for the FPGA

Workspaces

- Using the workspace
  - Usage is the same as most IDEs
  - Install new software on the workspace using Linux commands on the terminal
  - Can debug and run remotely using serial port or ssh or any existing Linux technology.
Compilation for the FPGA

Compilation environment

• Using the workspace
  • Usage is the same as most IDEs
  • Install new software on the workspace using Linux commands on the terminal
  • Can debug and run remotely using serial port or SSH or any existing Linux technology.
Current developments

- Integration with Theia

Theia is an IDE that runs inside an Eclipse Che workspace
- When version 7 of eclipse Che releases, will be che’s default IDE
- Is based on the opensource Monaco Editor that Powers Visual Studio Code
- It has compatibility with Visual Studio Code Plugins and Extensions
Current developments

• Integration with Theia

- Developed plugins Will be compatible with:
  - Eclipse Che + Theia
  - Visual Studio Code
- Can exploit all the features of the plugin platform
- The programmer can install optional or preferred plugins
Execution environment

- Nanos runtime system

- Resources available
  - OMP_NUM_THREADS / NX_GPUS / NX_NUM_FPGAS
  - Master and workers execute SMP tasks
  - One representative thread per accelerator
    - CUDA/OpenCL/FPGA device

- Getting work from the task pool
  - Different scheduling policies
  - “Implements” policy allows to exploit parallelism on “all resources”
Execution environment

- Nanos runtime system

- Matrix multiplication 2048x2048 single precision
  - On Xilinx Zynq U+ (4x A53 + FPGA)

- Implements
  - SMP OpenBLAS
  - autoVivado on FPGA
  - Great success

- 300 MHz on FPGA
  - Outperforms cores

- Parallel task creation
- New dataflow IP core

(*) 76.0 Gflop/s

26.8 Gflop/s (~Sep. 2017)
39.6
11.7 Gflop/s

3.2

36.5

GFLOPs

SMP Only 1 128 Acc 1 256 Acc 3 128 Acc 3 256 DF
Execution environment

- Tracing facilities

- 3x256 dataflow IP cores with “implements” and 4 ARM workers
  - ARM cores do a lot of work (on their possibilities)
Summary

Conclusions

- Developed IDE plugin for Eclipse for directives support
- Developed docker for automatic compilation on Eclipse Che
- Easy distribution of the OmpSs toolchain
- Use in Cloud environments
Summary

Future plans

• Keep improving the Eclipse/Eclipse Che support
• Keep improving the code generation with OmpSs toolchain
• Evaluating the possibility to use LLVM