UNIVERSITÄT PADERBORN

PADERBORN CENTER FOR PARALLEL COMPUTING (PC^2)

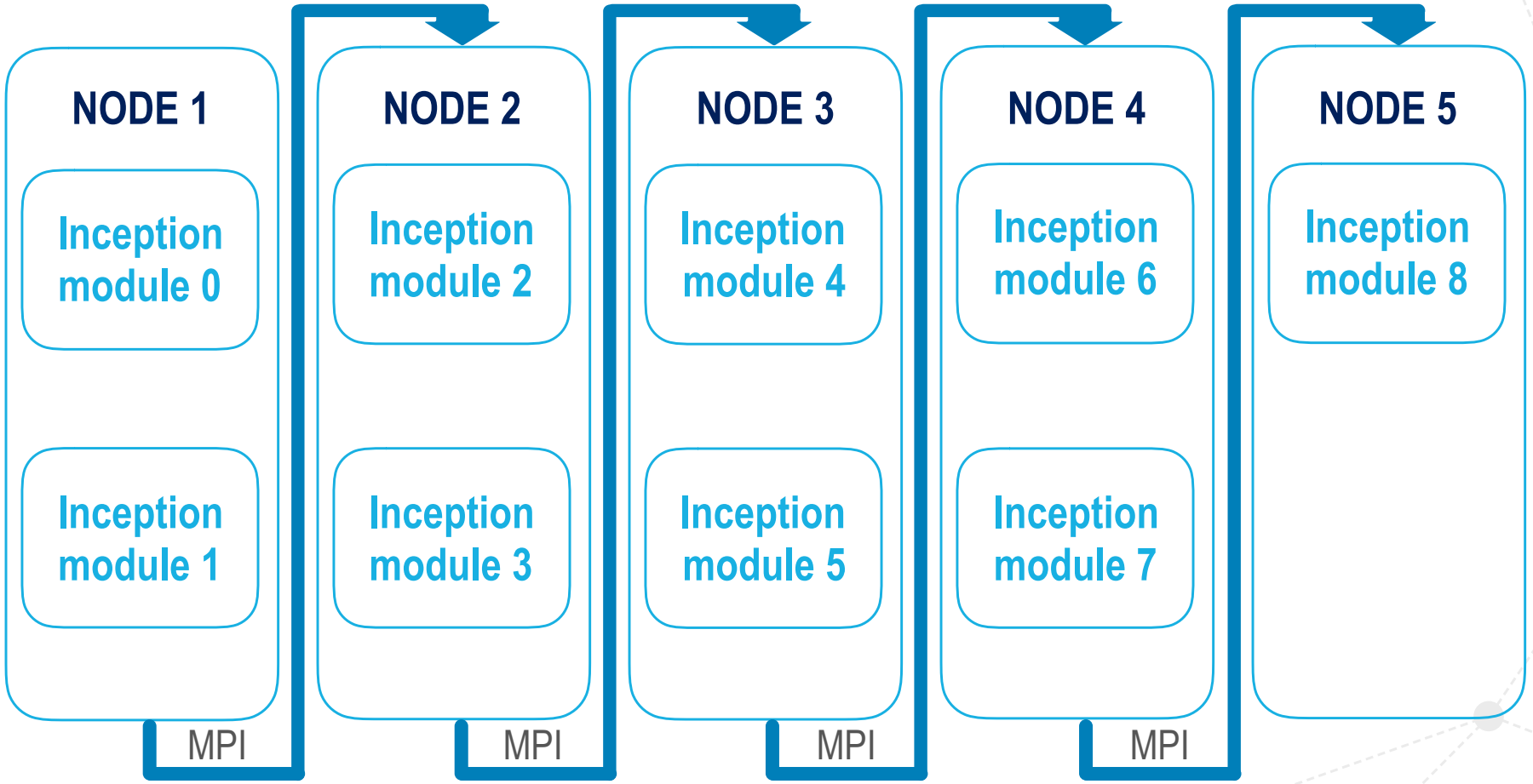
MPI WITHIN PLUG-IN

Alina Egorova

Why do we need the Message Passing Interface (MPI)?

- **Communication between nodes.**
  - Sending the result of last layer to the next FPGA

Schematic View



MPI. Introduction

/ test_plugin / **main.cpp**

```
#include "mpi.h"

MPI_Init(NULL, NULL);
int rank;
MPI_Comm_rank(MPI_COMM_WORLD,&rank);
fpga_launcher(network,inputModel,imageNames,
                                    cnn_model,rank);
```

MPI

/ noctua_plugin / **fpga_plugin.cpp**

```
#include "mpi.h"

std::string file1 = GoogLeNet_DIR+"inception"+std::to_string(2*rank)+".aocx";
std::string file2 = GoogLeNet_DIR+"inception"+std::to_string(2*rank+1)+".aocx";


std::string file1_xml = GoogLeNet_DIR+"inception"+std::to_string(2*rank)+".xml";
std::string file2_xml = GoogLeNet_DIR+"inception"+std::to_string(2*rank+1)+".xml";

char f1_xml[file1_xml.length()];
strcpy(f1_xml,file1_xml.c_str());

std::vector<std::string> first_kernels = xml_parser1(f1_xml);
```

MPI_Send

/ noctua_plugin / **fpga_plugin.cpp**

```
// MPI write to the next host instance
if(rank<com_sz-1&&program_number == 2)
{
        std::string concat_layer_name = p->layerName;
        MPI_Send(concat_layer_name.c_str(), concat_layer_name.size(), MPI_CHAR,
        int dims1 = p->outH * p->outW * p->outDepth;
        MPI_Send(&dims1, 1, MPI_INT, rank+1, 0, MPI_COMM_WORLD);
        float concat_out[p->outH * p->outW * p->outDepth];
        cmd_queues[p->layerID]->enqueueReadBuffer(*buffers[p->layerOutBufferIndex
        MPI_Send(concat_out, p->outH * p->outW * p->outDepth, MPI_FLOAT, rank+1,
}
```

MPI_Recv

/ noctua_plugin / **fpga_plugin.cpp**

```
if(rank!=0)
{
        char layer_name[50];
        MPI_Recv(layer_name, 50, MPI_CHAR, rank - 1, 0, MPI_COMM_WORLD,MPI_STATUS_IGNORE)

        std::string previous_l_name(layer_name);
        int dims_prev;
        MPI_Recv(&dims_prev, 1, MPI_INT, rank - 1, 0, MPI_COMM_WORLD,MPI_STATUS_IGNORE);

        float prev_data[dims_prev];
        MPI_Recv(prev_data, dims_prev, MPI_FLOAT, rank - 1, 0,MPI_COMM_WORLD,MPI_STATUS_

        find_by_name(root,previous_l_name,buffer_index);
```

MPI. Instruction

```
module load intel/18.0.3
module load mpi/OpenMPI/1.10.3-GCC-5.4.0-2.26
module load devel/CMake/3.6.1-foss-2016b
module load intelFPGA_pro/19.1.0 nalla_pcie/19.1.0 gcc/6.1.0

cd /upb/scratch/departments/pc2/groups/pc2-cc-user/custonn2/AEGOROVA/dldt/inference-engine/build/
/cm/shared/apps/pc2/EB-SW/software/devel/CMake/3.6.1-foss-2016b/bin/cmake -DCMAKE_BUILD_TYPE=Release -DENABLE_CLDNN=OFF -DENABLE_GNA=OFF ..
make -j16
```

MPI. Instruction

In dldt/inference-engine/bin/intel64/Release folder:

salloc -N 5 --partition=fpga -A hpc-lco-kenter —constraint=19.1.0

mpirun -npernode 1 ./test_plugin -m /upb/scratch/departments/pc2/groups/pc2-cc-user/custonn2/hakathur/misc/IR/frozen_quant.xml -i /upb/scratch/departments/pc2/groups/pc2-cc-user/custonn2/intermediate_representation/pepper.png > out3.txt

exit