Parallelization in Machine Learning with Multiple Processes

Omar Zapata and Gerardo Gutíerrez

http://oproject.org
Outline

- Parallel architectures
- ROOTMpi
  - MPI Current status and future
  - Implementation with ROOT
- TMVA
  - New architecture for parallelization
  - Parallelization prototype with MultiProc/MPI
- Conclusions
Parallel architectures

Why parallel computing?
- Huge amount of data
- Expensive algorithms

Which paradigm?
- A lot

Parallel Computing Paradigms

- Threads
  - Xeon Phi
  - Gpu
    - Cuda
    - OpenCL
  - OpenMP
  - MultiProc
  - Etc..

- Shared Memory
  - Many Cores
  - Message Passing Interface (MPI)

- Distributed Memory
  - Hadoop/Spark
  - Etc..
Message Passing Interface

- Standard of communication for HPC/Grid Computing
- Widely used in the scientific community.

Implementations
- OpenMPI, MPICH, IBM, Intel etc..

Support for:
- RMA (Remote Memory Access)
- Shared memory
- Checkpointing
- P2P and collective communication

Fault tolerance under development [http://mpi-forum.org/mpi-40](http://mpi-forum.org/mpi-40)
Integration of MPI and ROOT technologies in a framework for parallel computing.

Motivation:

- Communicate ROOT objects through processes.
- Implement MPI with a better design for ROOT.
- Create an interface that uses the new C++ features to write parallel code.

- **Implement TMVA algorithms in parallel for HPC/Grid systems.**
Prototype is going under development with the next features already implemented

- Peer to Peer communication
- Some collective operation implemented
  - Gather/AllGtaher
  - Scatter/AllScatter
  - Redude/AllReduce
  - Bcast
- Communicators
  - IntraCommunicator
  - InterCommunicator
- Blocking and non-blocking communication.
- Tested with OpenMPI and MPICH
ROOTMpi Design

ROOTMpi

ROOT

OpenMPI

MPICH

Etc..

MPI libraries

Serialization

ROOTMpi
int array[size];

MPI_Send(array, size, MPI_INT, dest, tag, MPI_COMM_WORLD);
ROOTMpi Design
Comparison with C

Send in ROOTMpi

```cpp
std::vector<int> array(size);
...
COMM_WORLD.Send(array, dest, tag);
```
ozapatam@gfif:~$ rootmpi -np 2 -machinefile nodes hello.C
hello from process = 1 node = gfif-\wn7
hello from process = 0 node = gfif-\wn6
Example
Peer to Peer

```cpp
using namespace ROOT::Mpi;
void p2p(){
  TEnvironment env;
  TMatrixD mat;

  if(COMM_WORLD.GetRank()==0){ // sending the message
    mat.ResizeTo(2,2);
    mat[0][0]=0.1;
    COMM_WORLD.Send(mat,1,0);
    cout<<"Sending matrix from process 0"<<endl;
    mat.Print();
  } else if(COMM_WORLD.GetRank()==1){ // receiving the message
    COMM_WORLD.Recv(mat,0,0);
    cout<<"Receiving matrix in process 1"<<endl;
    mat.Print();
  }
}
```
Example
Peer to Peer output

[gerardo] [gfig] [~]$ rootmpi -np 2 p2p.C

Sending matrix from process 0

2x2 matrix is as follows

|   0 |   1 |

|--------------|

0 | 0.1 | 0
1 | 0   | 0

Receiving matrix in process 1

2x2 matrix is as follows

|   0 |   1 |

|--------------|

0 | 0.1 | 0
1 | 0   | 0
New parallel architecture for TMVA

Parallel Architecture

Base

Algorithm

Results

#Jobs

Time

Etc..

Basic requirements.

Programming model.

Mpi

Threads

Spark

Etc..
Prototype with examples

Jupyter notebooks

ParallelExecutor (MultiProc)
ParallelExecutorMpi (OpenMPI)
Conclusions

- MPI standard have all need HPC/Grid computing
  - Shared/Distributed memory
  - Checkpointing
  - Fault tolerance under development
- ROOTMpi is
  - A modern interface for MPI that uses powerful C++ design
  - A great communication system through serialization.
  - An easy way to parallelize codes in ROOT for HPC/Grid computing.
- TMVA is developing a modern architecture for multiple parallelization paradigms like HPC/Grid computing
Thanks!

Questions?
Backups

- Machine Learning Software that are using MPI
  - IBM ml-toolbox [link]
  - Microsoft Distributed Machine Learning Toolkit [link]
  - Microsoft Multiverso [link]
  - Theano MPI [link]
  - eXtreme Gradient Boosting [link1] [link2]
  - Distributed TensorFlow with MPI [link]
  - Intel MPI Distributed Machine Learning [link]