GPU Enabled Spark MLlib

Lingyun Li
& Lei Yao
CS 848
University of Waterloo

Outline

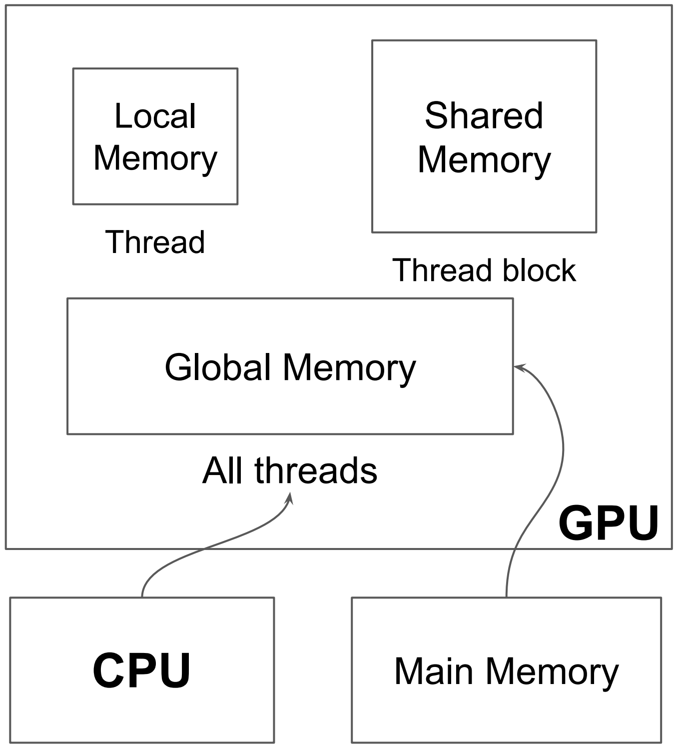
- Motivation
- GPU calculation model
- GPUEnabler
- Spark MLlib Algorithms for GPU computation
- Implementation using GPUEnabler
- Performance evaluation
- Current & future work

Motivation

- Problem
  - Computation heavy spark machine learning applications
  - CPU computation bottleneck
- Goal
  - Accelerate Spark MLlib
  - Leverage high performance GPUs
  - Second dimension of distribution
  - Without change of user programs

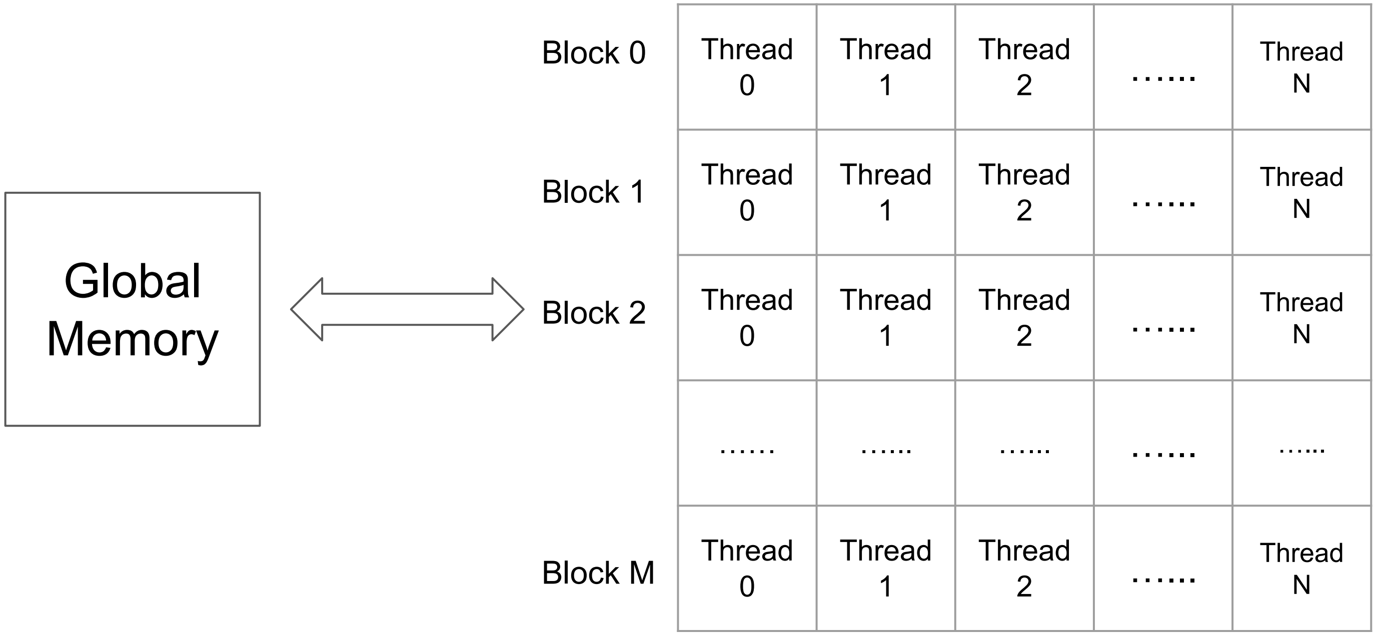
GPU Calculation Model

- Five steps for GPU programming
  - Allocate GPU device memory
  - Copy data on CPU main memory to GPU device memory
  - Launch a GPU kernel to be executed on in parallel
  - Copy back data from GPU memory to main memory
  - Free GPU memory



GPU Calculation Model

BlockDim.x = N



int idx = threadIdx.x + blockIdx.x * blockDim.x

Data Parallelism: Single Instruction, Multiple Data

GPUEnabler

- Offload specific tasks (GPU kernel) to GPU
- Get the data into a format that GPU can consume
- Read data from local memory to GPU memory and vice versa
- Applications can work in a heterogenous environment

Two Transformation APIs

mapExtFunc()

cacheGpu()

One Action API

reduceExtFunc()

Algorithms Suitable for GPU Computation

- Large dataset
- Complex mathematical computation
- Low data inter-dependency
- Low dependency between cluster nodes

Spark MLlib Algorithms for GPU Acceleration

- Naive Bayes
  - Mainly count and aggregation
  - Not enough mathematical computation
- Decision tree learning
  - Mathematical computation (Information gain) hidden deeply under nested map functions
- LBFGS
  - Calculation uses external numerical processing library Breeze
- SVMs and linear regression
  - Not enough mathematical computation
- **Logistic regression**
  - Candidate for GPU acceleration

Implementation using GPUEnabler

- Write CUDA kernel
- Create and broadcast *CUDAFunction* objects
    - Information about CUDA kernel, input/output data type, constant arguments, etc.
- Call *mapExtFunc* and *reduceExtFunc* instead of *map* and *reduce*
    - Execution of CUDA kernel in parallel

CUDA Kernel

```
__device__ void PredictPoint(const double * __restrict__ feature, double
label, double *result, const double * __restrict__ mapWeights, int
dimension) {
    Multiply(result, feature, 1 / (1 +  exp(DotMultiply(mapWeights, feature,
     dimension))), dimension);
}

extern "C"
__global__ void MapGpuKernel(int *number, double *feature, double *label,
double *result, double mapWeights, int dimension) {
    int idx = threadIdx.x + blockIdx.x * blockDim.x;

    if(idx < *number) {
        PredictPoint(&feature[idx * dimension], label[idx], &result[idx *
        dimension], mapWeights, dimension);
    }
}
```

GPUEnabler APIs

```
val ptxURL = LogisticRegression.getClass.getResource("/SparkGpuLogisticRegression.ptx")
val mapFunction = sc.broadcast(
    new CUDAFunction(
    "MapGpuKernel",
    Vector("this.feature, this.label"),
    Vector("this"),
    ptxURL)
    )

val scoreAndLabels = point.mapExtFunc((point: Vector) =>
    Vector(1.0 / (1.0 + math.exp((DotMultiply(mapWeights.value, point.feature)))) : + point.label,
    mapFunction.value,
    outputArraySizes = Array(Dimension),
    inputFreeVariables = Array(mapWeights.value)
    ).cacheGpu
```

Performance Evaluation

- Use logistic regression for classification
- GPU: Nvidia Tesla K80

# of data points	# of features each data point	# of machine in cluster	Use GPU	Runtime (ms)
1000000	10	1	No	1182
1000000	10	1	Yes	2826
1000000	10	2	No	1276
1000000	10	2	Yes	3494
2000000	15	1	No	6511
2000000	15	1	Yes	5938
2000000	15	2	No	5760
2000000	15	2	Yes	5639

Our Work

- Setup cluster with GPU, CUDA, Spark, HDFS and GPUEnabler
- Learn Spark MLlib algorithms
- Study Spark MLlib & GPUEnabler source code
- Integrate GPUEnabler & Spark
- Implement GPU Enabled MLlib algorithms
- Deploy and run GPU code on clusters
- Performance evaluation
- Future work:
    - Implement and evaluate more algorithms
    - Investigate GPU computation bottleneck

Thank you

Questions?