The INRIA Project LAB HPC-BigData: Addressing the HPC/Big-Data/IA Convergence

Bruno Raffin,
INRIA Grenoble Rhône-Alpes

Lyon, October 2018

The HPC-BigData Project Lab

An INRIA funded project (2018-2022)

- Gather teams from HPC, Big Data and Machine Learning to work on the convergence

INRIA teams:

- **HPC teams:** DataMove, KerData, Tadaam, RealOpt, Hiepacs, Storm, Grid'5000
- **IA teams (and Big Data):** Zenith, Parietal, Tao, SequeL, Sierra

External partners:

- **Academic**: Lab Biologie Théorique (CNRS Paris)Academic: Argonne National Lab (USA)
- **Industry**: ATOS/Bull, ESI-group

https://project.inria.fr/hpcbigdata/

The Convergence

Three Research Directions:

- **Infrastructure and resource management**
- **HPC acceleration for AI and Big Data**
- **AI/Big Data analytics for large scale scientific simulations**

HPC versus BigData/AI

Parallelism for scalability

HPC

→Performance comes first

→ Low level programming

MPI+OpenMP

→ Thin software stack

→ Stable software libs

→ HPC centers

Jobs run a few hours on thousands of cores:

- Sensitivity Analysis : 30 000 cores for 1h30 [Terraz'17]
- Exastamp material simulation: 8000 cores for a few hours

Big Data/AI

→ Ease of programming comes first

→ High level programming

Spark, Flink, TensorFlow, Pytorch

→ Thick software stack

→ Quickly changing software libs

→ Cloud platforms

Jobs run a few days on tens of nodes:

- Pl@ntNet learning: one week on 4 GPUs
- AlphaGo Zero ltraining: 70 hours on 64 GPU workers and 19CPU parameter [Silver'17]
- ResNet-50 on 256 GPUs in 1 hour (mini-batch training) *[Goyal 2017]*

Some of our Software Assets

Machine Learning in Python

Light yet Flexible
Batch Scheduler

StarPU

Task Programming for
Hybrid architectures

FlowVR, Melissa, Damaris

On–line data processing engines
for HPC

Pl@ntNet

Deep Learning based App
for plant identification

Infrastructure and Resource Management

HPC Infrastructure for AI:

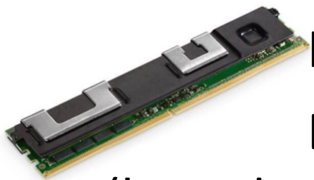
New needs:

- Accelerators (GPUs or other)
- Large resident data sets (learning & benchmarks) (PlantNet: 10 TB of raw data)
- Very long runs (days)
- Fast changing software stacks (TensorFlow, PyTorch)

On-going work on AI/HPC compliant resource sharing approaches

Playground: Grid'5000, Genci experimental GPU cluster, etc.

Get data close to the compute nodes:



HPC versus Cloud platforms: **External file system versus on-node disks**

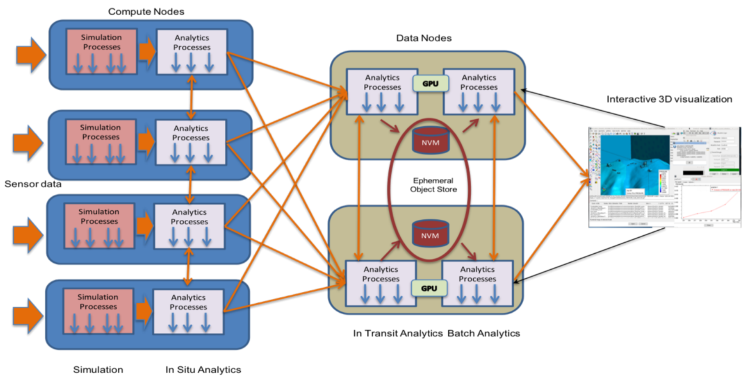
But changing: on-node persistent storage for energy and performance (burst buffers, NVRAM): **Locality aware resource management**

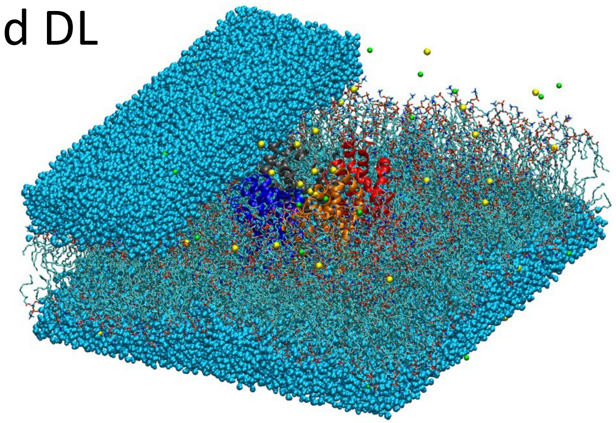
AI/Big Data Analytics for Large Scale Scientific Simulations

Molecular dynamics trajectory analysis with deep learning:

Dimension reduction through DL

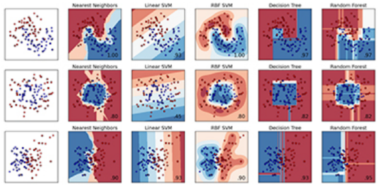
Accelerating MD simulation coupling HPC simulation and DL





Flink/Spark stream processing for in-transit on-line analysis of parallel simulation outputs

[ISAV'18]

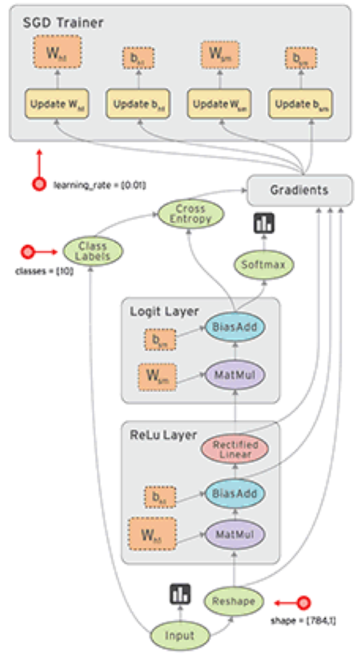


HPC for AI

Shallow Learning

Accelerating Scikit-Learn with task-based progamming (Dask, StarPU)

TensorFlow



Deep Learning:

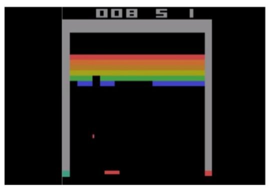
TensorFlow graph scheduling for efficient parallel executions:

Scheduling for automatic differentiation and backpropagation

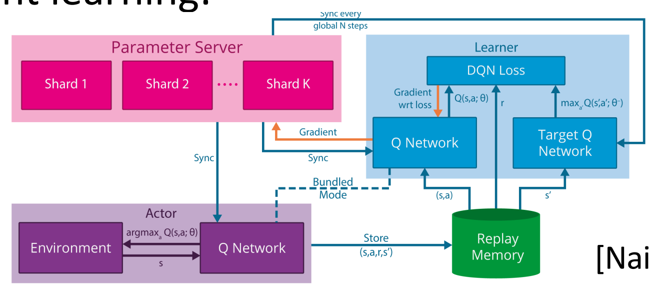
Recompute versus store frontward results

Linear algebra and tensors for large scale machine learning

Large scale parallel deep reinforcement learning:

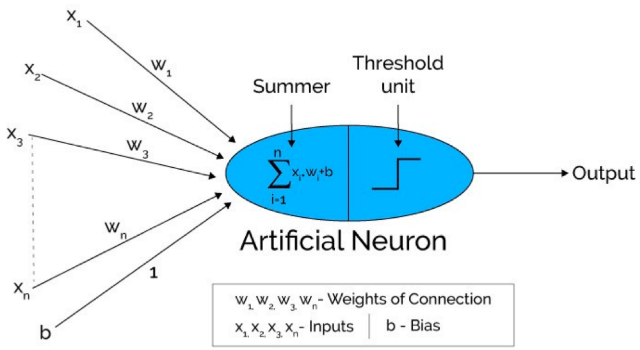


Self-learn to play Atari games



[Nair et al. 2015]

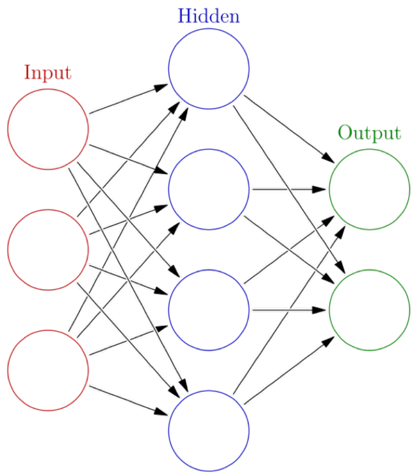
Artificial Neural Networks



Activation function

$$f(x) = K\left(\sum_i w_i g_i(x)\right)$$

Weight optimization by stochastic Gradient descent (backpropagation)



Example x_i

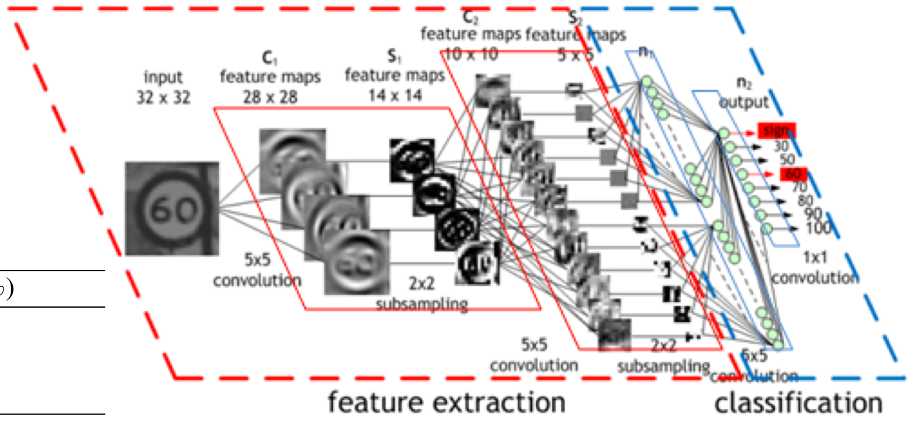
Output: o_i

Compute error on output

Backpropagate error and compute weight updates

Deep Learning

Today's neural networks are deep and complex:



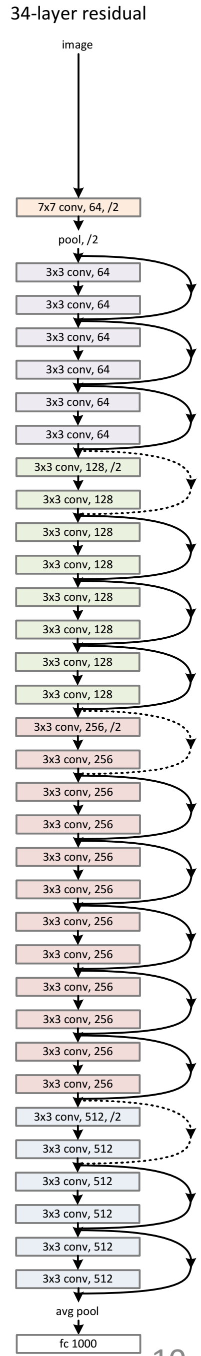
method	# layers	# params	error (%)
Maxout [10]			9.38
NIN [25]			8.81
DSN [24]			8.22
FitNet [35]	19	2.5M	8.39
Highway [42, 43]	19	2.3M	7.54 (7.72±0.16)
Highway [42, 43]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

Table 6. Classification error on the **CIFAR-10** test set. All methods are with data augmentation. For ResNet-110, we run it 5 times and show "best (mean±std)" as in [43].

[HE-CVPR2016]

Hyperparameter setting has become a very complex task -> learning for discovering hyperparameters ?

ResNet-34



Parallelizing Deep Learning

Parameter update function

Generic learning process: $W^t = F(D, W^{t-1})$

Learning Data

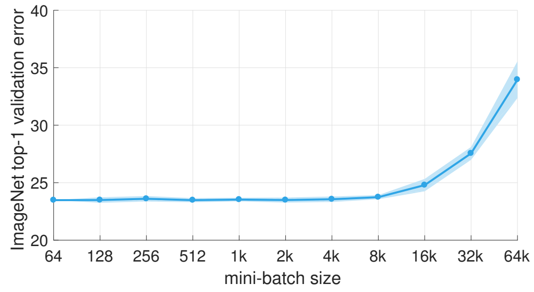
Model Parameters

Often the parameters updates are computed after presenting a batch of examples (batch learning)

2 main sources of parallelism:

- **Data parallelism**: distribute the learning set
- **Model parallelism**: distribute the model parameters

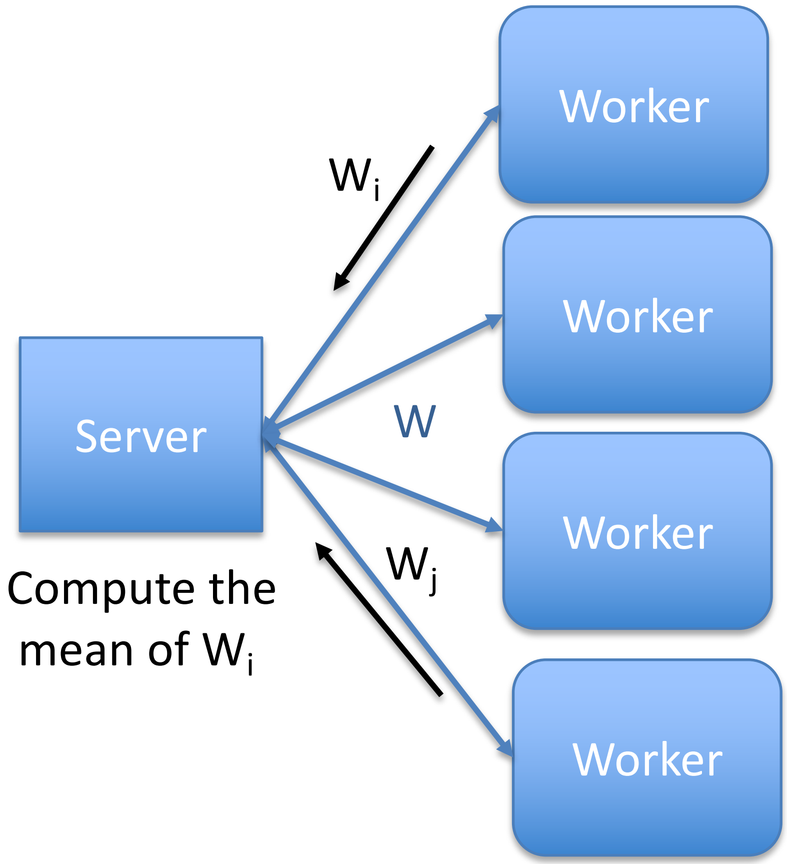
Data Parallelism



[Goyal 2017]

Duplicate the model (one per worker)

Partition the batch into P minibatches, one per worker



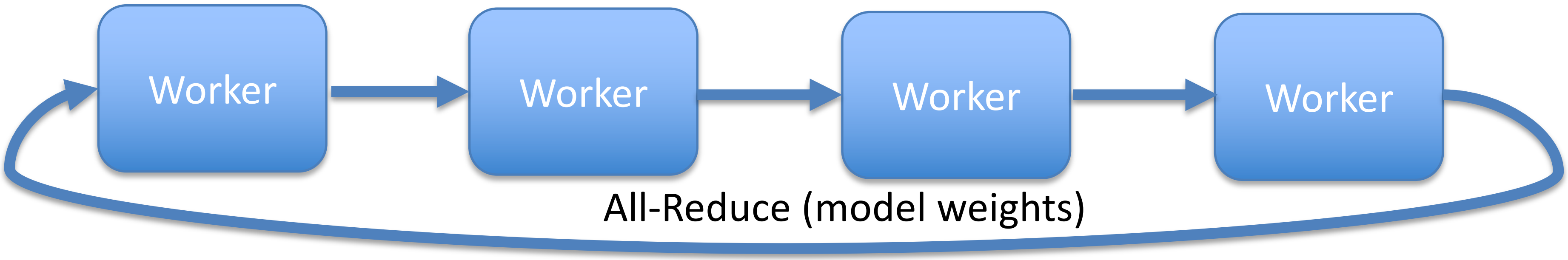
Synchronous update (TensorFlow):

Loop:
  Server sends parameters to all Workers;
  Workers compute parameter updates
    on their mini-batch;
  Server get updates from all Workers;
  Server compute a global model update;
  Server update parameters;
EndLoop

Limitations:
- Server is a bottleneck:
  gets P sets of model parameters
- minibatch size affects learning convergence

Data Parallelism

Fix the bottleneck: suppress the server and perform a all-reduce



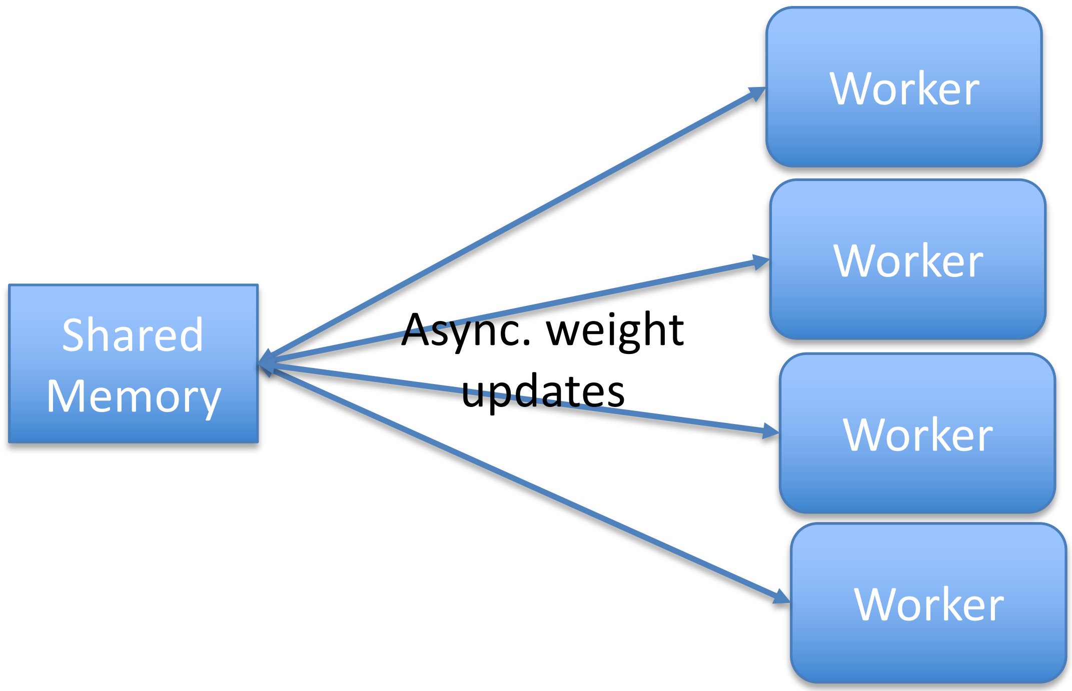
Communication cost per worker is now independent on the number of workers

Baidu initially proposed a modified version of Tensorflow based on MPI, now available in Horvod (Uber, still Tensorflow+MPI)

Data Parallelism: Asynchronous Updates

Asynchronous Stochastic Gradient Descent:

- Each worker update asynchronously the model parameters
- Proven convergence under certain conditions [Hogwild! 2011]
- But practically convergence may be affected in such a way that it outweighs the performance gain from asynchronism.



Software 2.0

Software 1.0

- Deterministic computations with algorithms
- Computation must be correct for debugging

Software 2.0 [introduced by A. Karpathy]

- Probabilistic machine-learned models trained from data
- Computation only has to be statistically correct

Creates many opportunities for improved performance

[K. Olukotun Keynote at ISCA 2018]

Software 2.0

[From K. Olukotun Keynote at ISCA 2018]

Relax, It's Only Machine Learning

- Relax synchronization: data races are better
  - HogWild! [De Sa, Olukotun, Ré: *ICML 2016*, ICML Best Paper]
- Relax cache coherence: incoherence is better
  - [De Sa, Feldman, Ré, Olukotun*: ISCA 2017*]
- Relax communication: sparse communication is better
  - [Lin, Han et. al.: *ICLR 18*]
- Relax precision: small integers are better
  - HALP [De Sa, Aberger, *et. al*.]

Chris De Sa

Song Han

Chris Aberger

Better hardware efficiency
with negligible impact on statistical efficiency

Leverage the stochastic nature of ML for loosening data dependencies constraints and thus support better parallelization.