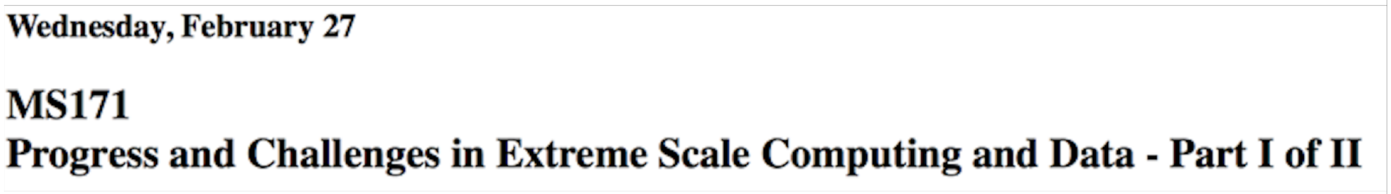


Wednesday, February 27

MS171
Progress and Challenges in Extreme Scale Computing and Data - Part I of II

SIAM Conference on
Computational Science and Engineering

February 25-March 1, 2019
Spokane Convention Center
Spokane, Washington, USA

Combining Extreme Computing and Big Data for Future Machine Learning

Serge G. Petiton, Kesheng (John) Wu, and Osni Marques

BERKELEY LAB
Bringing Science Solutions to the World

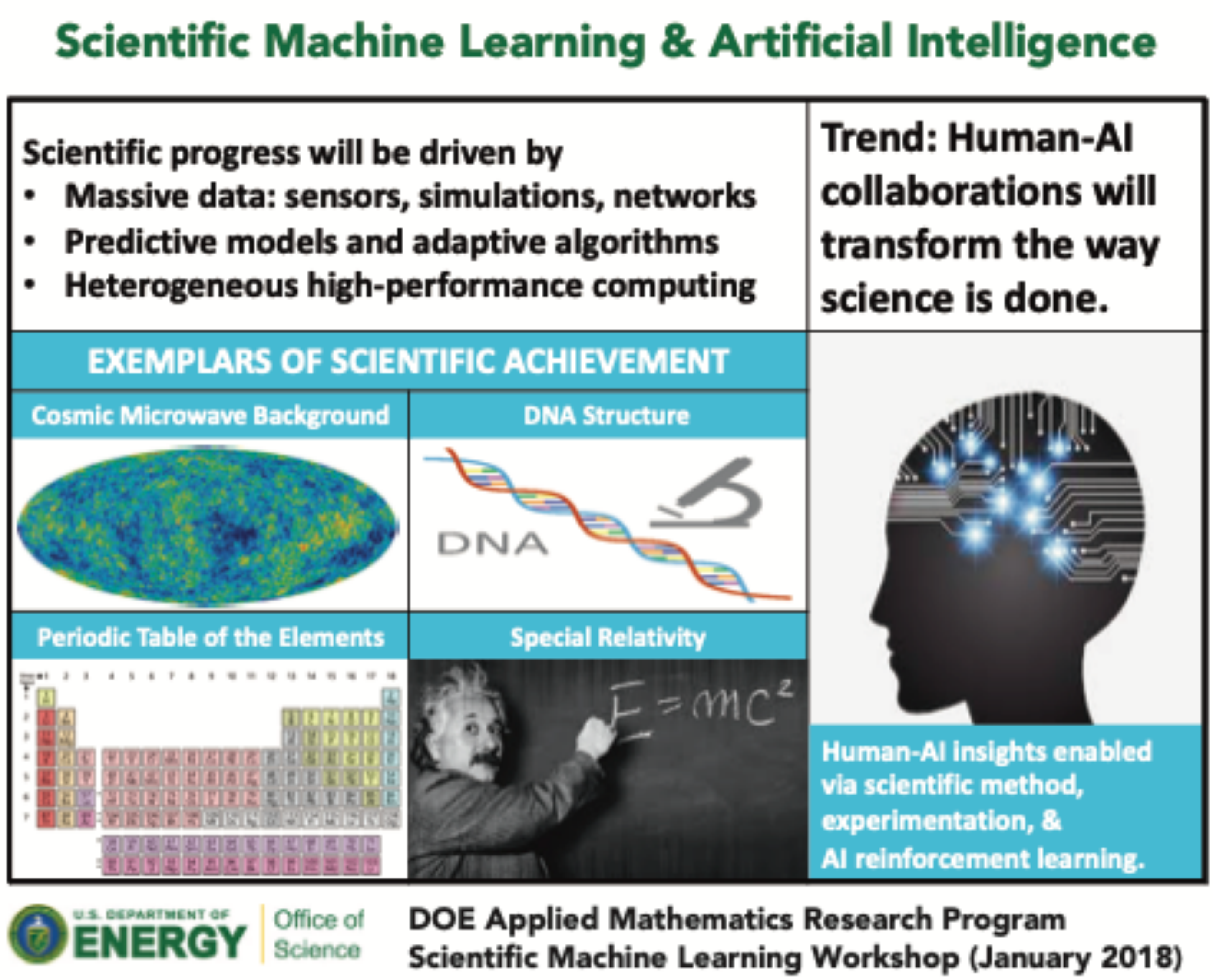
cnrs dépasser les frontières

Université de Lille

Outline

- Introduction
- Future computing, programming, and applications?
- Combining Extreme Computing and ”Big Data”
- Summary

From DOE Report on "Basic Research Needs for Scientific Machine Learning"



Scientific Machine Learning & Artificial Intelligence

Scientific progress will be driven by

- Massive data: sensors, simulations, networks
- Predictive models and adaptive algorithms
- Heterogeneous high-performance computing

Trend: Human-AI collaborations will transform the way science is done.

EXEMPLARS OF SCIENTIFIC ACHIEVEMENT

Cosmic Microwave Background

DNA Structure

Periodic Table of the Elements

Special Relativity

Human-AI insights enabled via scientific method, experimentation, & AI reinforcement learning.

U.S. DEPARTMENT OF ENERGY | Office of Science

DOE Applied Mathematics Research Program Scientific Machine Learning Workshop (January 2018)

Introduction

Future machine Learning

- Strategic applications, with strong societal impacts : HPDA, Predictive medicine, geoscience, human brain, virtual body, social sciences, genetics, smart cities, …. and **applications we don't imagine yet!**
- Have to use the faster available supercomputers and datacenters

Exascale Machine Learning

- Extreme Computing : Exascale machine (202X and Y Mwatts?) and beyond
  - *What is exascale : 64, 32, 16 bits, mixed arithmetic, TOP500, HPCG, Graph500???*
  - *Supercomputers were parallel machines, they become distributed and parallel machines*
- Big Data : have to be distributed along large number of disks, clouds, platforms
  - *Different (cheaper) hardware than HPC, different methods*
  - *Data Centers were mainly distributed platforms, they have to collaborate with parallel computational based nodes*
- Machine Learning : what new methods?
  - *32, or 16, bits arithmetic often enough*
  - *Linear Algebra (and 64 bit arithmetic) still important (graph decomposition, regression, ranking, …)*

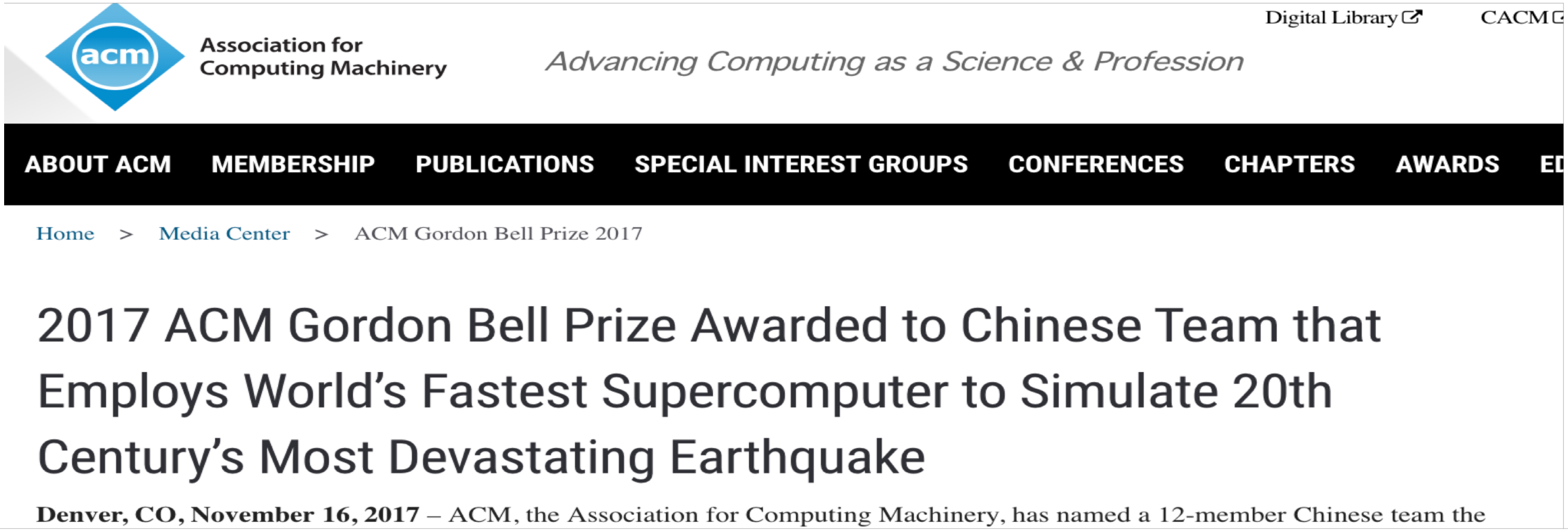
Combining extreme computing and Big Data for future machine learning

- We have to define a realistic roadmap and set milestones
- First combining HPC and Big Data, second proposing programming paradigms and methods to efficiently use ”exascale” machines adapted for such applications, and then introducing new (“exascale”) machine learning methods

Outline

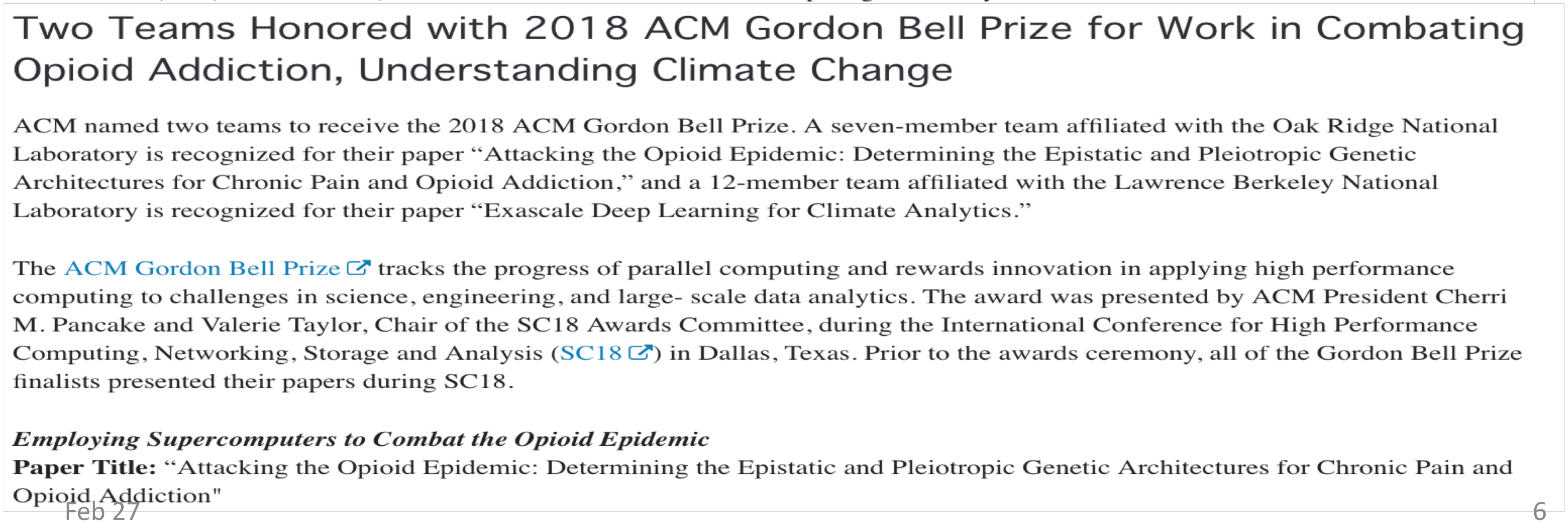
- Introduction
- **Future computing, programming, and applications?**
- Combining Extreme Computing and ”Big Data”
- Summary

Changing HPC Applications: Examples from Gordon Bell Prizes


Association for Computing Machinery
Advancing Computing as a Science & Profession
Digital Library
CACM
ABOUT ACM
MEMBERSHIP
PUBLICATIONS
SPECIAL INTEREST GROUPS
CONFERENCES
CHAPTERS
AWARDS
ED
Home > Media Center > ACM Gordon Bell Prize 2017

2017 ACM Gordon Bell Prize Awarded to Chinese Team that Employs World's Fastest Supercomputer to Simulate 20th Century's Most Devastating Earthquake

Denver, CO, November 16, 2017 – ACM, the Association for Computing Machinery, has named a 12-member Chinese team the



Two Teams Honored with 2018 ACM Gordon Bell Prize for Work in Combating Opioid Addiction, Understanding Climate Change

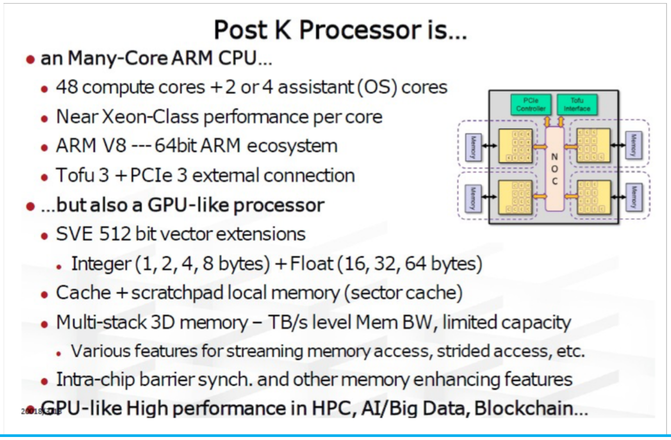
ACM named two teams to receive the 2018 ACM Gordon Bell Prize. A seven-member team affiliated with the Oak Ridge National Laboratory is recognized for their paper "Attacking the Opioid Epidemic: Determining the Epistatic and Pleiotropic Genetic Architectures for Chronic Pain and Opioid Addiction," and a 12-member team affiliated with the Lawrence Berkeley National Laboratory is recognized for their paper "Exascale Deep Learning for Climate Analytics."

The ACM Gordon Bell Prize tracks the progress of parallel computing and rewards innovation in applying high performance computing to challenges in science, engineering, and large- scale data analytics. The award was presented by ACM President Cherri M. Pancake and Valerie Taylor, Chair of the SC18 Awards Committee, during the International Conference for High Performance Computing, Networking, Storage and Analysis (SC18) in Dallas, Texas. Prior to the awards ceremony, all of the Gordon Bell Prize finalists presented their papers during SC18.

Employing Supercomputers to Combat the Opioid Epidemic
Paper Title: "Attacking the Opioid Epidemic: Determining the Epistatic and Pleiotropic Genetic Architectures for Chronic Pain and Opioid Addiction"

Future computing, programming, and applications?

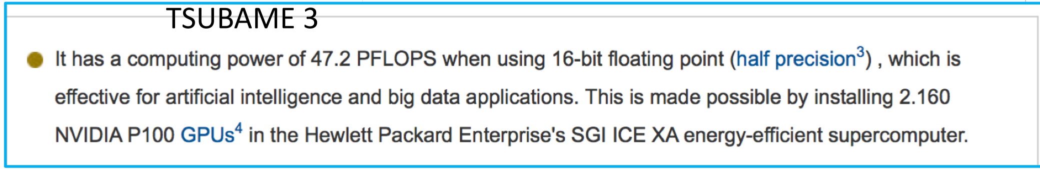
- Simulations would not be the main applications on such platforms
- Hierarchical heterogeneous architectures
- Networks on chip : distributed memory even on chips
- Supercomputers combine distributed and parallel computing
- New architectures would target first 32 or 16 bits arithmetic, to minimize energy
- The data access have to be optimized and computational nodes have to be include on data centers



Post K Processor is…

- an Many-Core ARM CPU…
  - 48 compute cores + 2 or 4 assistant (OS) cores
  - Near Xeon-Class performance per core
  - ARM V8 --- 64bit ARM ecosystem
  - Tofu 3 + PCIe 3 external connection
- …but also a GPU-like processor
  - SVE 512 bit vector extensions
    - Integer (1, 2, 4, 8 bytes) + Float (16, 32, 64 bytes)
  - Cache + scratchpad local memory (sector cache)
  - Multi-stack 3D memory – TB/s level Mem BW, limited capacity
    - Various features for streaming memory access, strided access, etc.
  - Intra-chip barrier synch. and other memory enhancing features
- GPU-like High performance in HPC, AI/Big Data, Blockchain…

PCIe Controller
Tofu Interface
Memory
N O C



TSUBAME 3

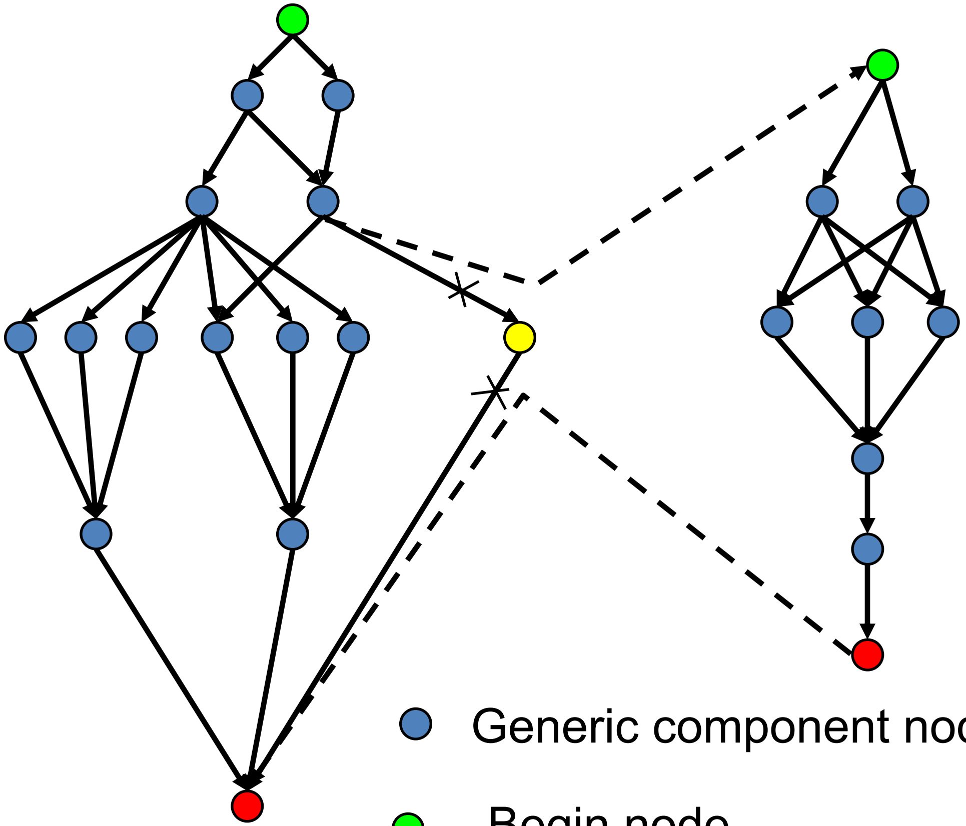
● It has a computing power of 47.2 PFLOPS when using 16-bit floating point (half precision[3]) , which is effective for artificial intelligence and big data applications. This is made possible by installing 2.160 NVIDIA P100 GPUs[4] in the Hewlett Packard Enterprise's SGI ICE XA energy-efficient supercomputer.

- The data ”prefetching” has to be efficiently asynchronously scheduled : a hierarchy of memory have to be design from the cores to the data centers
- What programming paradigms, what architectures and what algorithms?
- We have to :
  - Develop smart schedulers to optimize data communications and computation
  - Multi-level programming : distributed (large granularity), parallel (nodes), manycores, accelerators, MT, on chip,…
- Graph of Parallel Tasks/Conponents/Containers are serious candidates for such programming

Graph of Tasks/Components/Containers (TCC)

- Each task/component/container may be an existing method/software developed for a large part of the cores, but not all of them
- The computation on each core may use multithread optimizations and runtime libraries
- Accelerator programming may be optimized also at this level
- Then we have the following levels of programming and computing:
  - Graph of task/components/containers, already developed or new ones,
  - Each TCC is run on a large part of the computer, on a large number of cores. Using SPMD, PGAS-like, data parallel languages
  - On each processor, we may program accelerators,
  - On each core, we have a multithread optimisation.
- Data migrations between computing nodes and data storage units/Data Centers have to be anticipated, with persistence when possible, based on smart scheduling algorithms
- We have to allow the users to give expertise to the middleware, runtime system and schedulers. Scientific end-users have to be the principal target on co-design process. Frameworks and languages have to consider them first.

Graph (n dimensions) of TCC
The YML Example



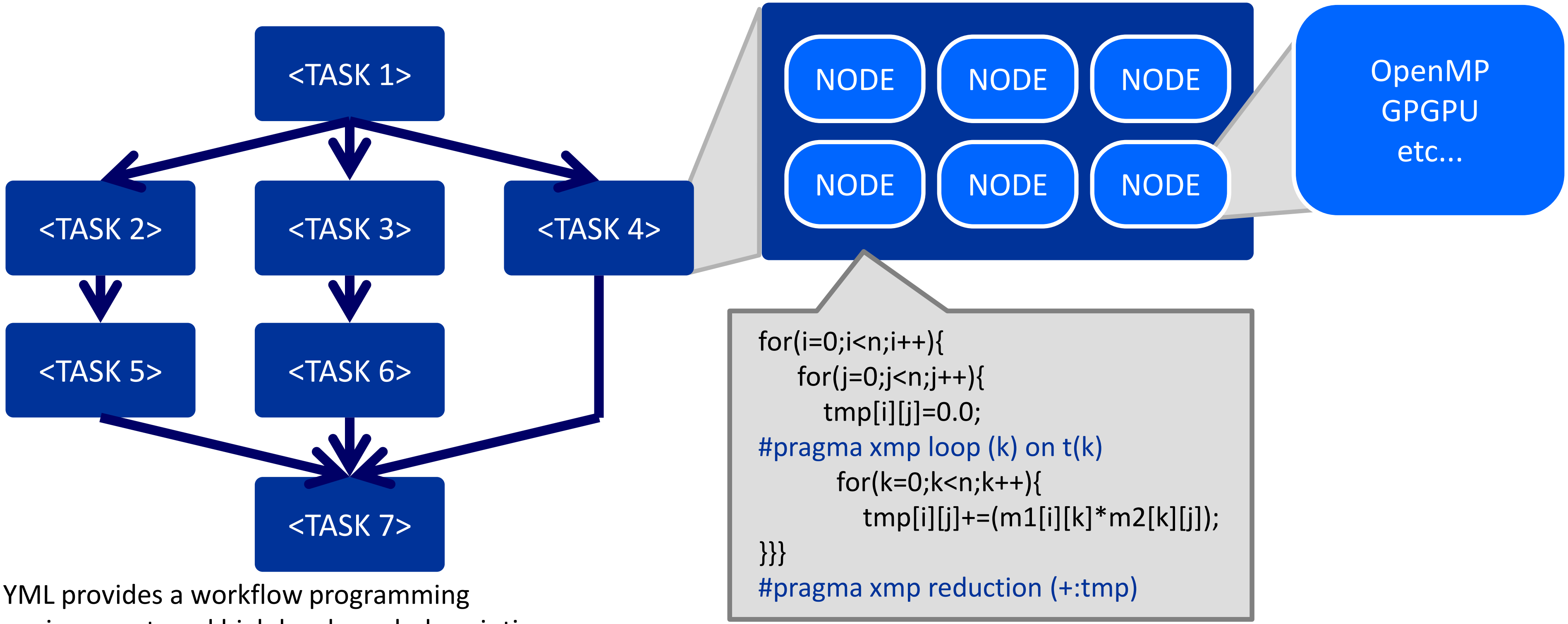
- 🔵 Generic component node
- 🟢 Begin node
- 🔴 End node
- 🟡 Graph node
- → Dependence

```
par
  compute task1(..);
  notify(e1);
//
  compute task2(..); migrate matrix(..);
  notify(e2);
//
  wait(e1 and e2);
  Par (i :=1;n) do
    par
        compute task3(..);
    notify(e3(i));
    //
    if(I < n)then
      wait(e3(i+1));
      compute task(..);
      notify(e4);
     endif;
    //
    compute task5(..); control robot(..);
    notify(e5); visualize mesh(…) ;
    end par
  end do par
//
  wait(e3(2:n) and e4 and e5);
  compute task6(..);
  compute task7(..);
end par
```

Multi-Level Parallelism Integration: YML-XMP

Experimented on GRID, P2P, supercomputers : ”K” (RIKEN), Hooper (LBNL), Romeo (U. Reins),…

N dimension graphs



```
for(i=0;i<n;i++){
     for(j=0;j<n;j++){
         tmp[i][j]=0.0;
#pragma xmp loop (k) on t(k)
          for(k=0;k<n;k++){
              tmp[i][j]+=(m1[i][k]*m2[k][j]);
}}}
#pragma xmp reduction (+:tmp)
```

YML provides a workflow programming environment and high level graph description language called YvetteML

Each task is a parallel program over several nodes.
XMP language can be used to descript parallel program easily!

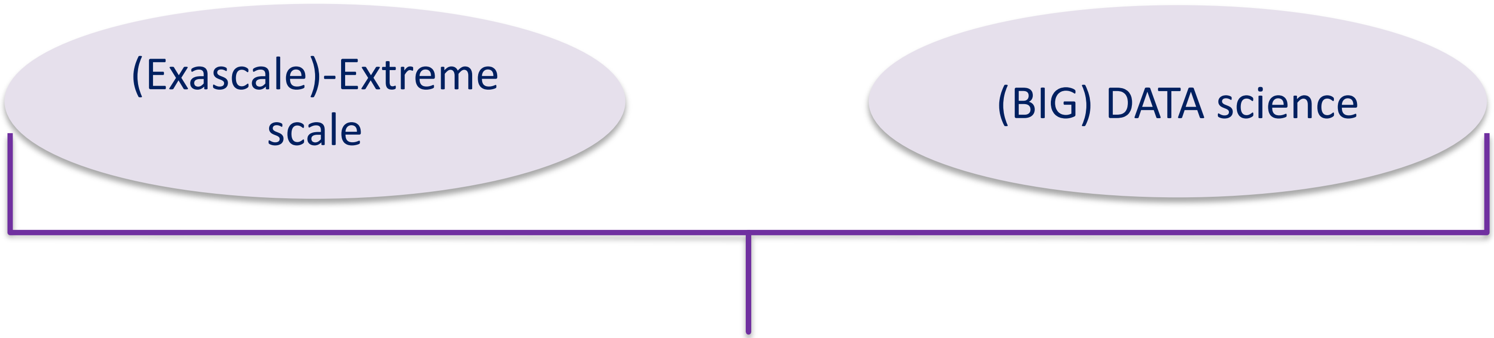
YML/XMP/StarPu tested on T2K in Japan, project FP3C

Outline

- Introduction
- Future computing, programming, and applications?
- **Combining Extreme Computing and ”Big Data”**
- Summary

Combining HPC and Data Science

Combining HPC and Big Data is the first step to use efficiently new and future supercomputers

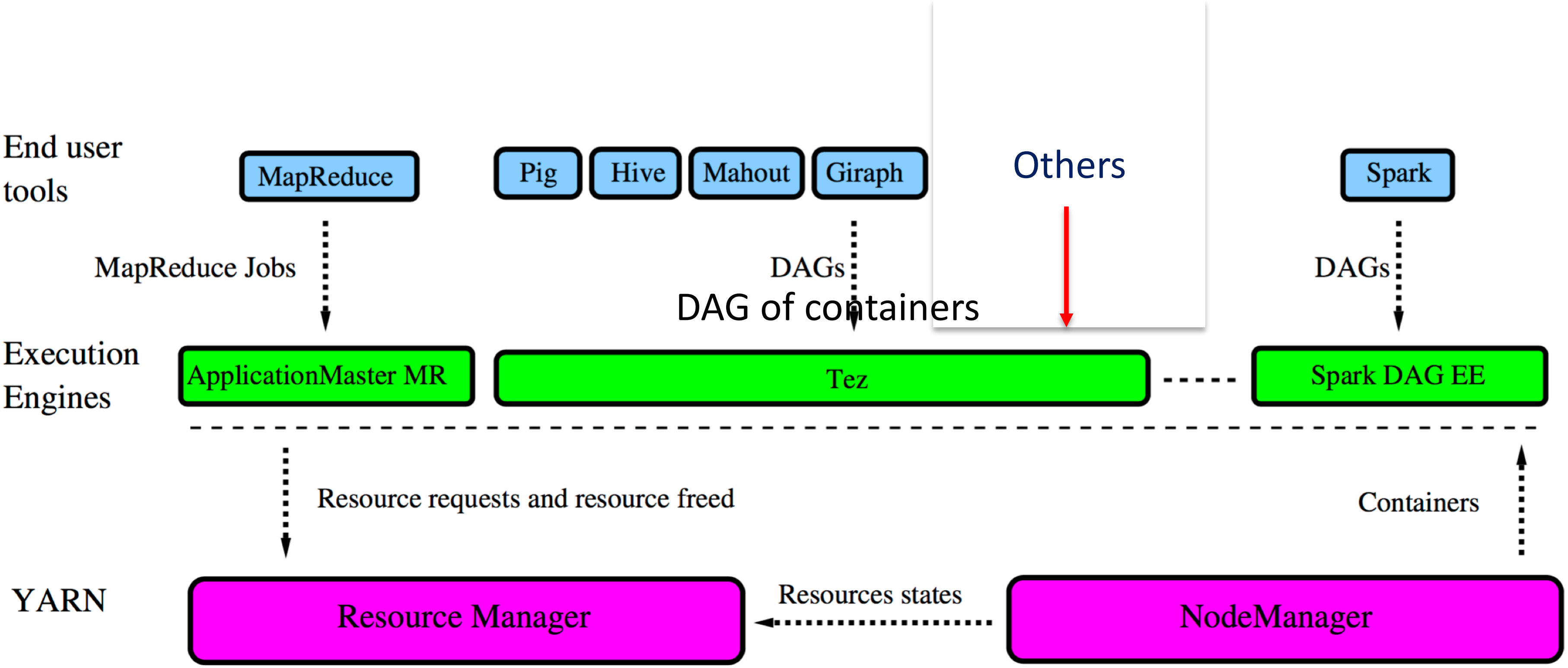


- Two different worlds with different hardware, organization,..
- They both have to use tasks, or component, or containers to encapsulate more local parallel/multi-level computations
- Distributed and multilevel programming, asking for smart scheduling and optimized data migrations

Existing example : Map-Reduce use to compute sparse matrix computation on the data “world”. Others parts of the algorithm are distributed on the “HPC” world, Ranking (PageRank,…..)

Existing Big Data world

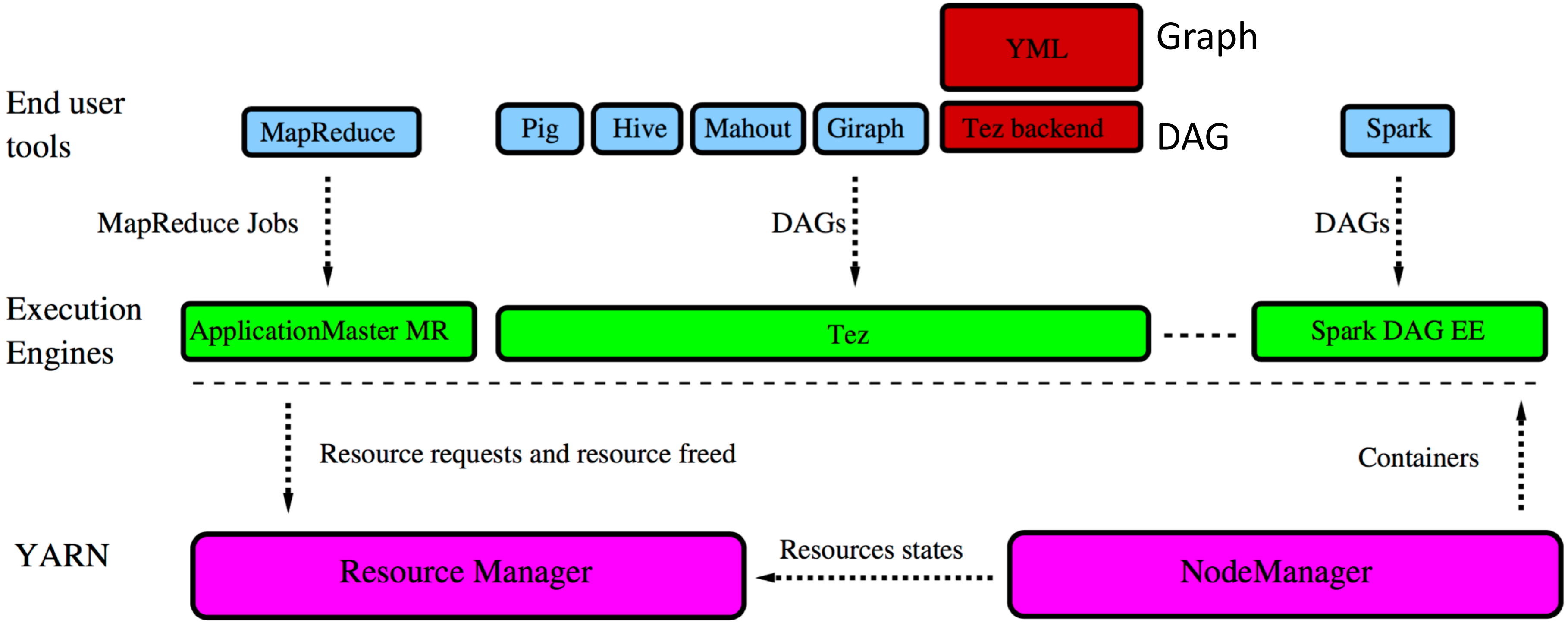
DAG of containers to use TEZ
Multilevel programming

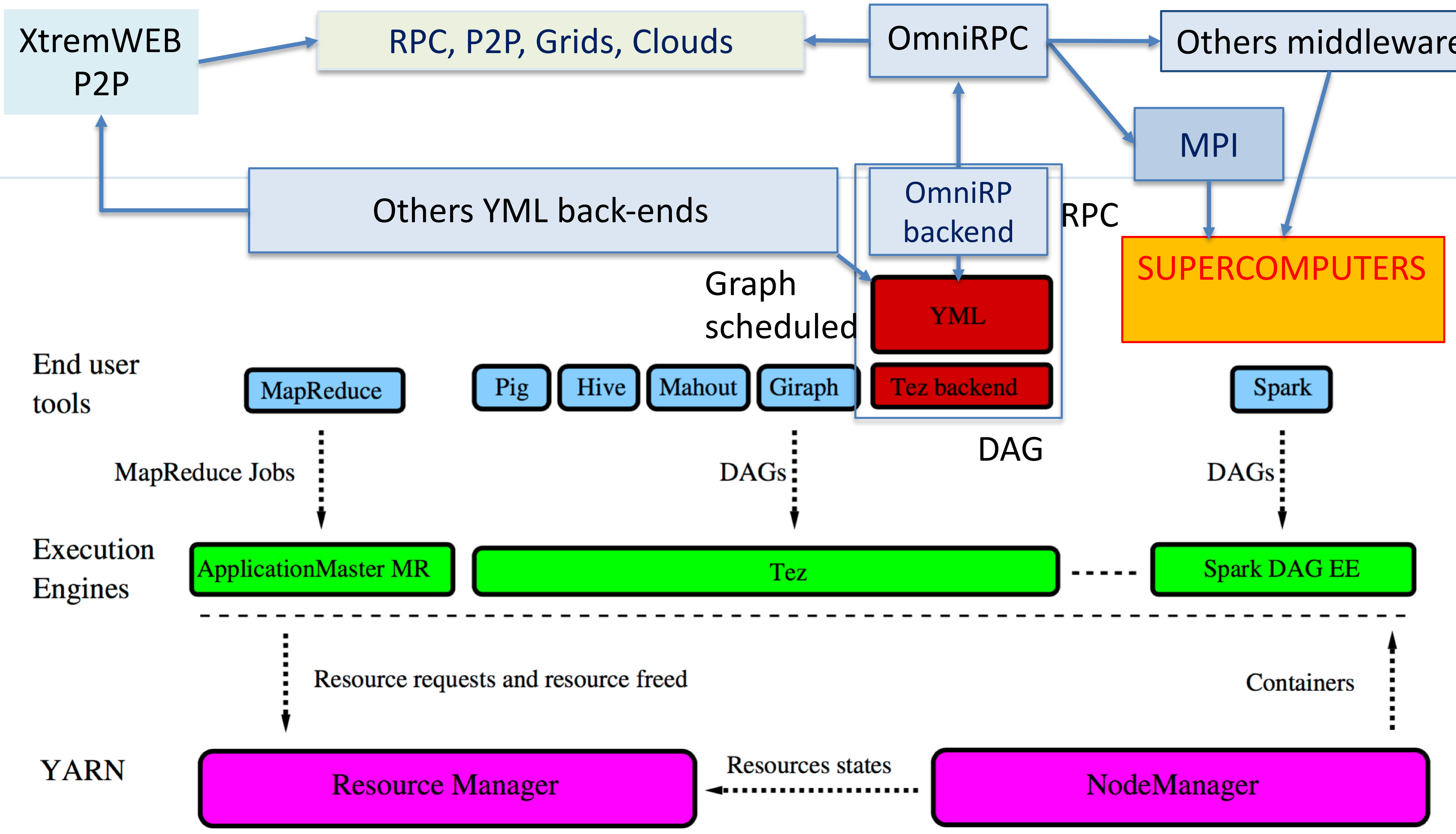


Adapted from Laurent Bobelin's slides

YML/TEZ backend for Big Data

Graph of TCC using YML : use to generate the DAG for TEZ





XtremWEB P2P

RPC, P2P, Grids, Clouds

OmniRPC

Others middleware

MPI

Others YML back-ends

OmniRP backend

RPC

SUPERCOMPUTERS

Graph scheduled

YML

End user tools

MapReduce

Pig

Hive

Mahout

Giraph

Tez backend

Spark

DAG

MapReduce Jobs

DAGs

DAGs

Execution Engines

ApplicationMaster MR

Tez

Spark DAG EE

Resource requests and resource freed

Containers

YARN

Resource Manager

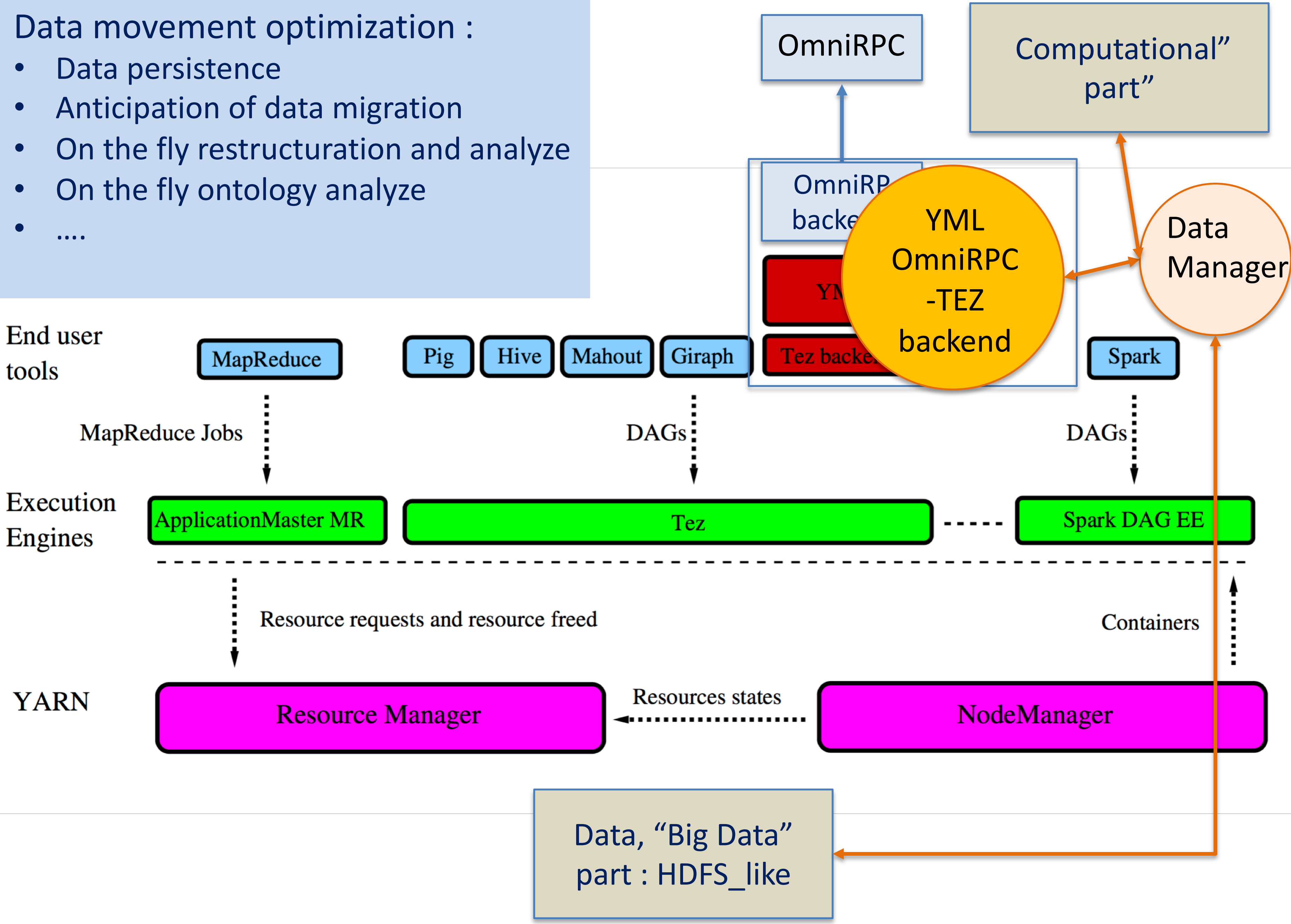
Resources states

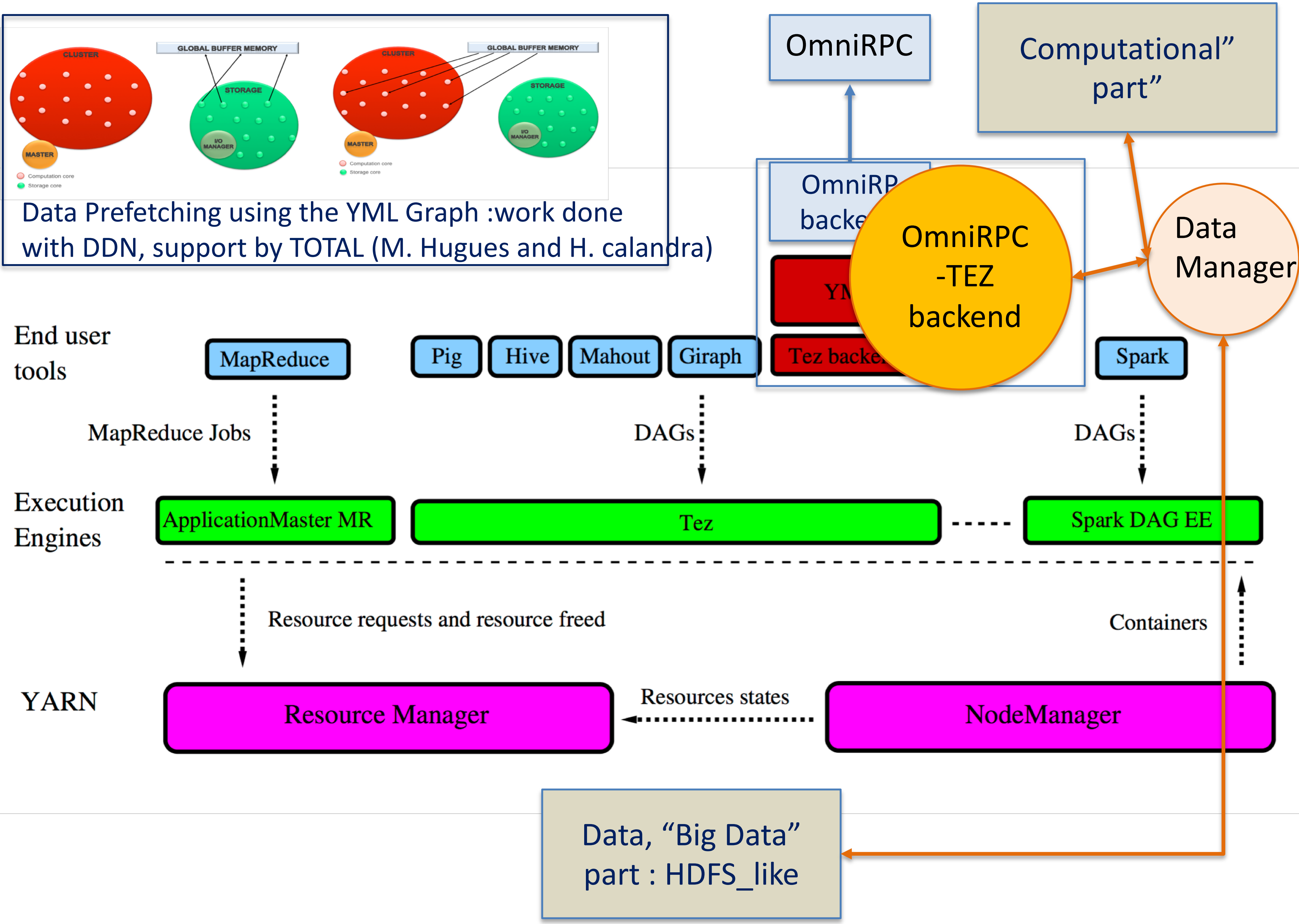
NodeManager

Convergence between HPC and Big Data

Data movement optimization :

- Data persistence
- Anticipation of data migration
- On the fly restructuration and analyze
- On the fly ontology analyze
- ….





Data Prefetching using the YML Graph :work done with DDN, support by TOTAL (M. Hugues and H. calandra)

OmniRPC

Computational" part"

OmniRP backe

YM

Tez backe

OmniRPC -TEZ backend

Data Manager

End user tools

MapReduce

Pig

Hive

Mahout

Giraph

Spark

MapReduce Jobs

DAGs

DAGs

Execution Engines

ApplicationMaster MR

Tez

Spark DAG EE

Resource requests and resource freed

Containers

YARN

Resource Manager

Resources states

NodeManager

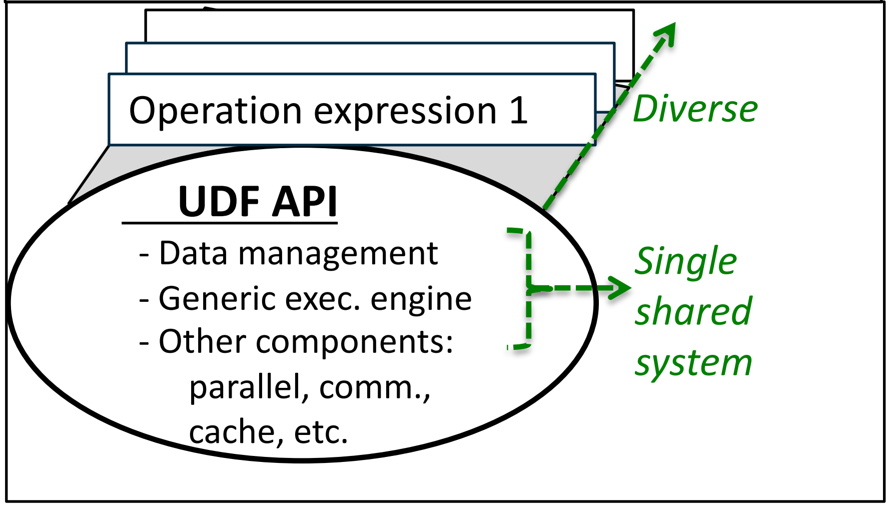
Data, "Big Data" part : HDFS_like

Another Possibility:
Separating Data Management from Data Analyses

Customized Solutions

```
For each operation P  Do
    Develop P's :
     - Data management            ---> Redundant   X
     - Expression execution       ---> Diverse
     - Other components:          ---> Redundant   X
          parallel,
          communication
          cache,
          etc.
End For
```

User-defined Functions (UDF)

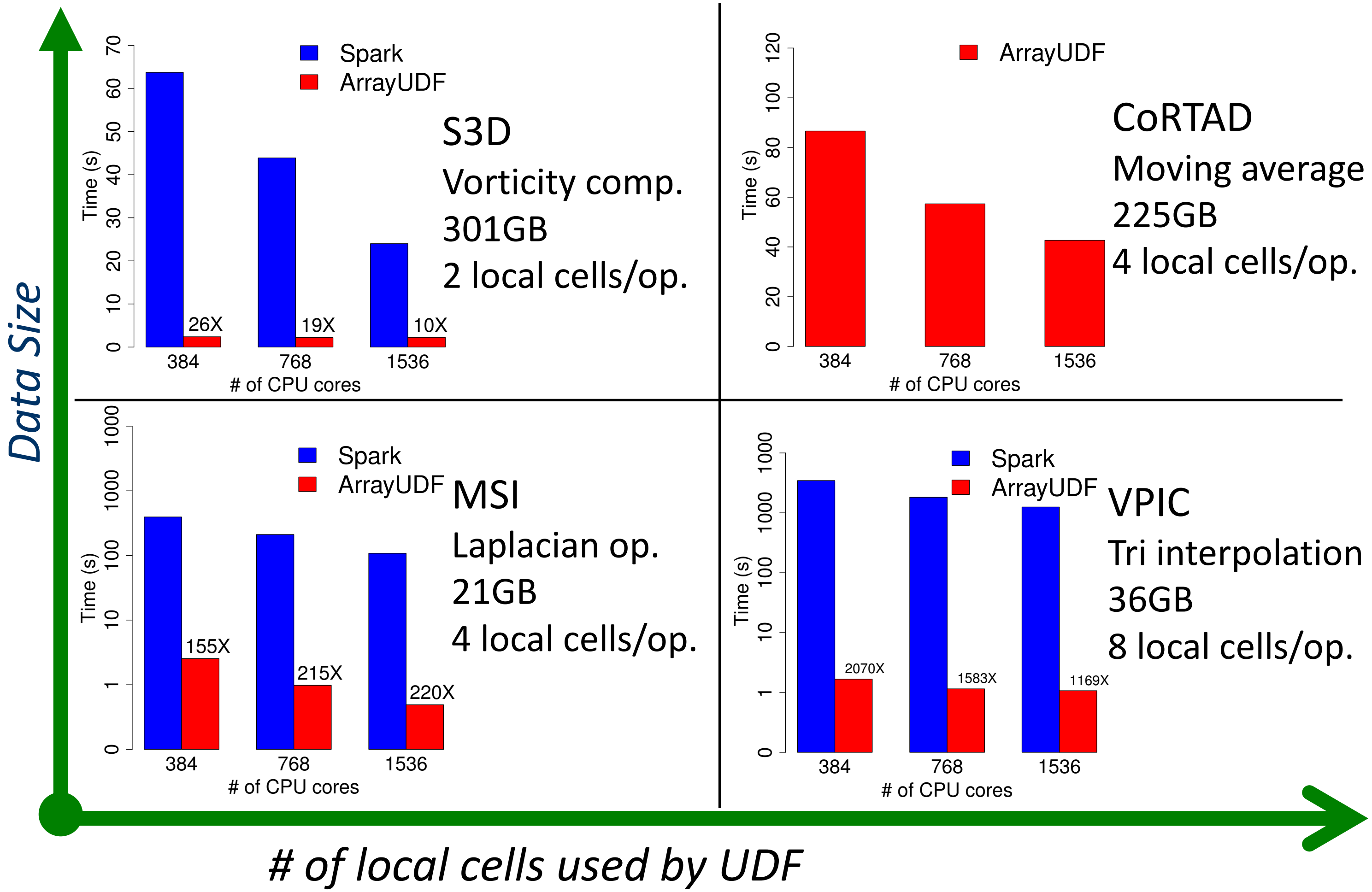


Scientific data analyses typically are custom programs ✗

Big Data systems separate data management from data analyses

Example: Scientific Data Services with array data model and ArrayUDF
https://bitbucket.org/arrayudf/arrayudf

Comparison with Spark – Common Scientific Data Analysis Tasks



S3D
Vorticity comp.
301GB
2 local cells/op.

CoRTAD
Moving average
225GB
4 local cells/op.

MSI
Laplacian op.
21GB
4 local cells/op.

VPIC
Tri interpolation
36GB
8 local cells/op.

Spark experiences out-of-memory:

- large data size

- more local cells

We observed ArrayUDF is 2070X faster

Outline

- Introduction
- Future computing, programming, and applications?
- Combining Extreme Computing and ”Big Data”
- **Summary**

New applications for HPC-Big Data

Using :

- Stochastic processes
- Bayesian statistic
- Graphs (decomposition, analysis,..)
- Game theory
- Classification
- Regression
- Others "classical Machine Learning algorithms
- AI

- Linear algebra for those news machines and applications
- Applied mathematics
- "new algebras" (CERFACS, TOTAL,..

A lot of scientific applications may have an "optimization" specification

Simulation versus observation :

Formula to results
versus
Data to formula

Combining HPC and “Big Data”

- HPC and Big Data are now on distributed and parallel environments
- These environments may not be (completely) integrated for economic and applications criteria : even if architectures would be sometime interleaved.
- How to program :
  - Graphs of task/components/containers (large granularity) : **YML**, Legion, Swift, Tensorflow…..
  - Each “task” may be a parallel encapsulate program
  - Data parallelism on each (cluster of) processor (Global array, PGAS, **XMP**)
- Mixing distributed and parallel programming
- Data migration (anticipation and persistence)
- **A smart scheduler may optimize ”HPC computation”, “Big Data”, and data migrations inside and between the two worlds, with respect of existing and future architectures.**
- Then ,we must first combine HPC and “Big Data” to open the road to future new applications and “extreme” machine learning.

Questions?

Contact information:

John Wu John.Wu@nersc.gov

SDM group

http://crd.lbl.gov/sdm/