Emerging Technologies

ACI-REF Virtual Residency – Aug 6-10, 2018

Dirk Colbry – FPGAs
Michigan State University – Director of HPC Studies

Mariya Vyushkova – Quantum Computing
Univ. of Notre Dame – Quantum Computing Research Specialist

Anita Schwartz – Singularity Containers
University of Delaware – Information Resource Consultant

Shawn Doughty – OnDemand Jupyter/RStudio
Tufts University – Sr Research Technology Spec

David Chin – Hadoop/Spark/MapReduce
Drexel University – Sr Systems Administrator

Field Programmable Gate Arrays (FPGA)

Dirk Colbry

Michigan State University

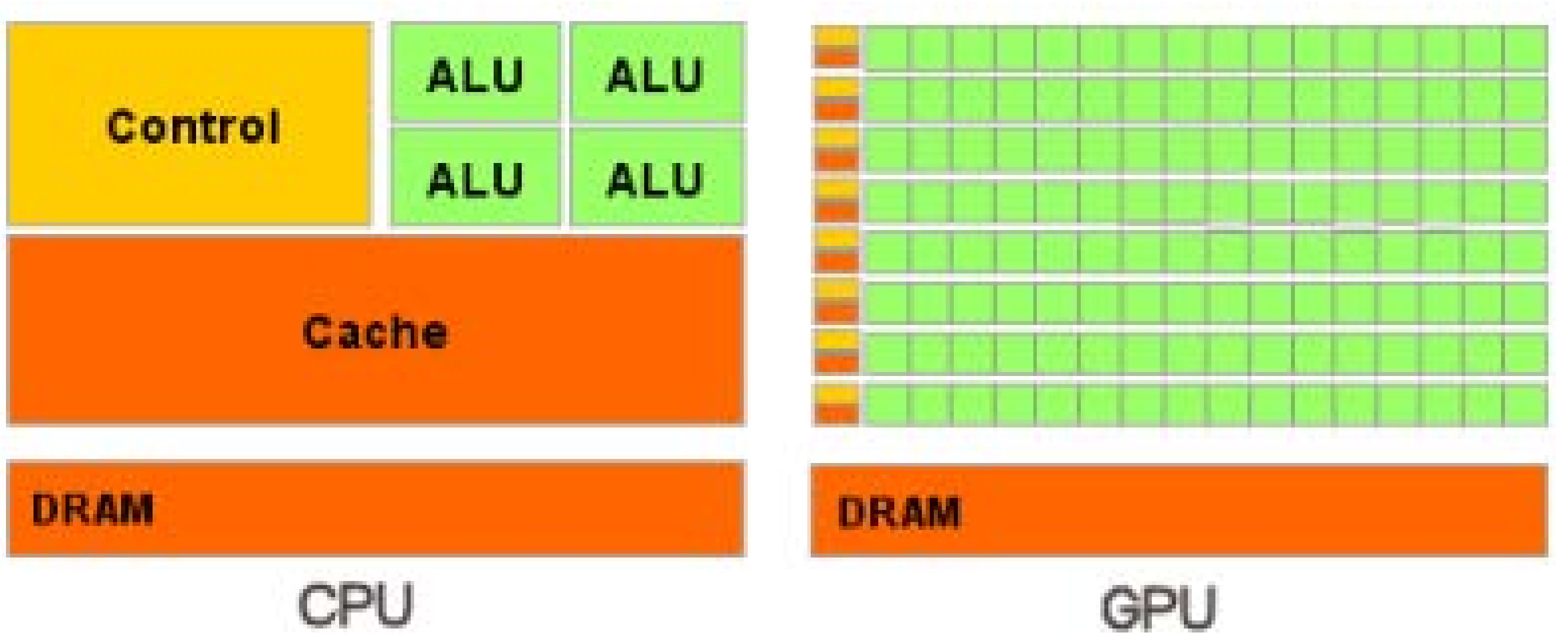
Why me?

- Taught an FPGA undergraduate lab way back in 1999-2002 when I was in graduate school.
- Did a sabbatical in 2015 with PixelVelocity, a company based in Ann Arbor that builds smart cameras that use FPGAs.

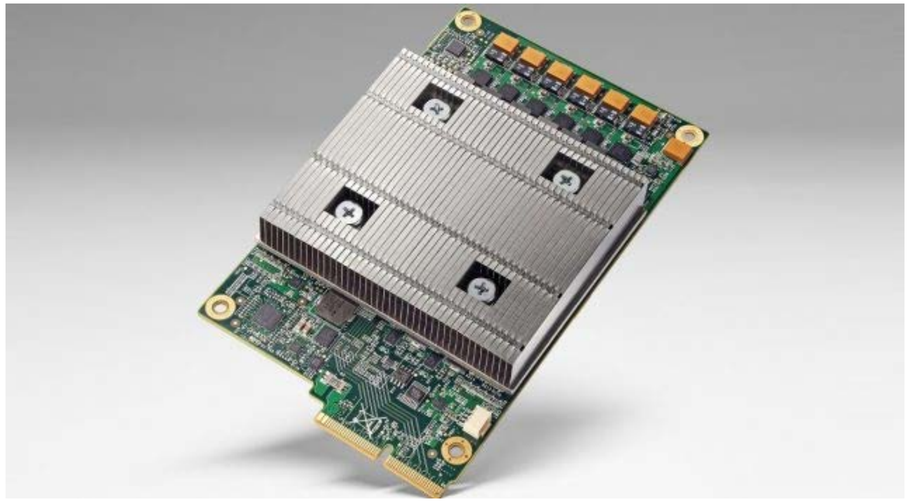
@Michigan State

- Purchased 2 nodes with a total of 6 Altera FPGAs
- Started a FPGA Taskforce for interested researchers
- Working with a Graduate Student on an independent study related to an optimization problem (function fitting) on FPGAs
- This fall we plan to conduct a Graduate Course to benchmark and test the 7 dwarves algorithms

Traditional Programmable GPUs and CPUs



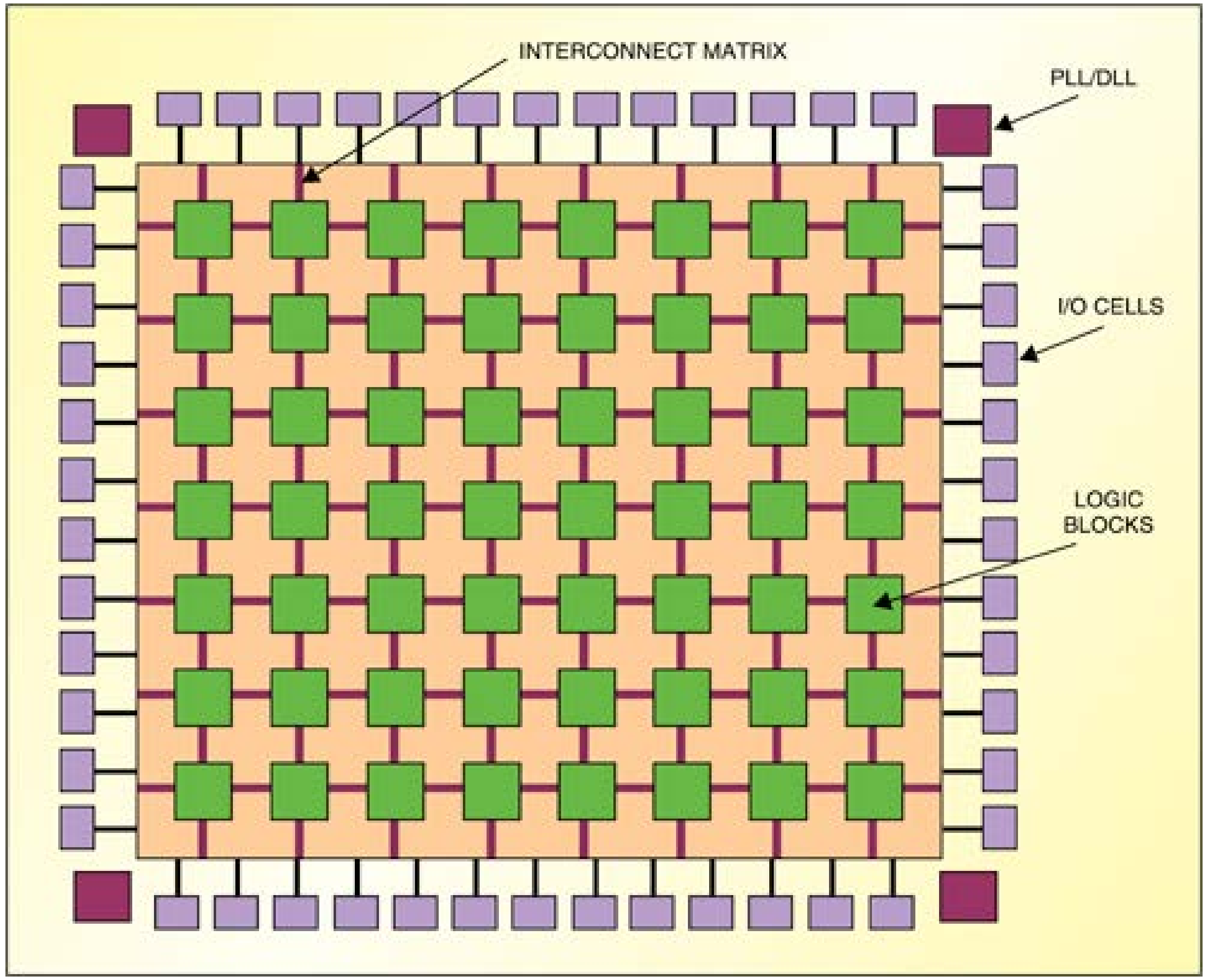
Application Specific Integrated Circuit (ASIC)



Neural Networks
Tensor Processing Units (TPU)

Molecular Dynamics
GRAvity PipE (MDGRAPE)

Field Programmable Gate Array (FPGA)



Notes about FPGAs

- For some applications speeds may be significantly faster than GPU/CPU. However, they will probably never be faster than ASICs.
- Key to FPGA’s success will be flexibility.
- FPGAs make the problem of optimization much more complex. You now can change both the hardware and the software. (I,.e. you may be trading human time for
- Configuring the Hardware is Tricky. You have basically three options:
  - Draw the circuits yourself using (HDL)
  - Write code using a version of OpenCL
  - Download existing hardware “images”

Quantum Computing

Mariya Vyushkova

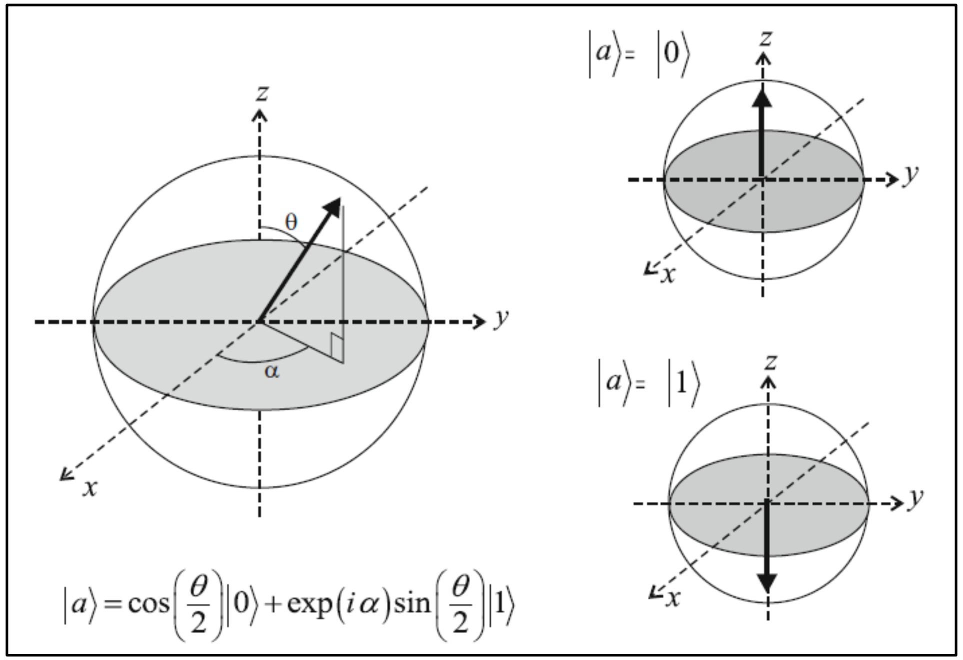
University of Notre Dame

ACIREF VR Workshop Panel: CI Emerging Technologies, August 10th, 2018

What is quantum computing? Qubit and spin

Quantum computation: information processing accomplished using quantum mechanical systems by harnessing the quantum mechanical phenomena such as superposition and entanglement

A **qubit (quantum bit)** is an elementary unit of quantum information.
Quantum mechanically, it is a simple two-state system.
Can be in a coherent **superposition** of both |0> and |1> states at the same time (in contrast to classical bits)
Multiple qubits can exhibit **quantum entanglement**



$|a\rangle = |0\rangle$
$|a\rangle = |1\rangle$
z
y
x
θ
α
$|a\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \exp(i\alpha)\sin\left(\frac{\theta}{2}\right)|1\rangle$

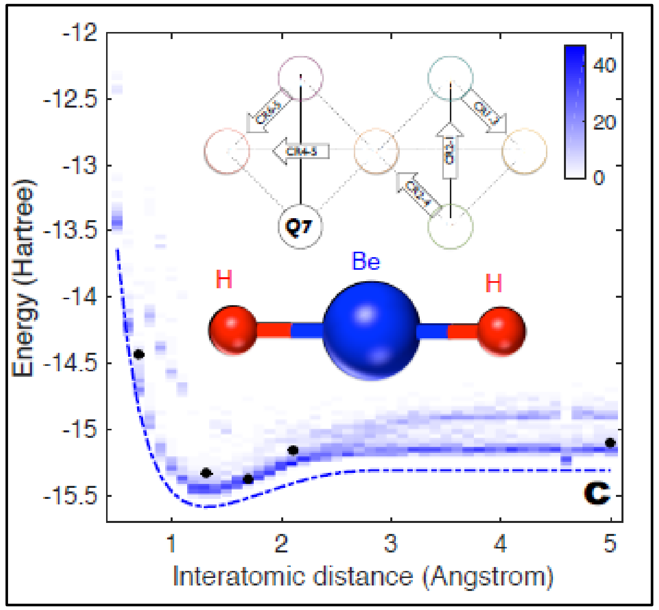
Qubit architectures: two-level atoms (IonQ); superconducting circuits (IBM Q, Rigetti, D-Wave); solid-state spin-based architectures (nuclear or electron spins as qubits)

Quantum programming: development of **quantum algorithms** and their implementation on a **quantum computing device**

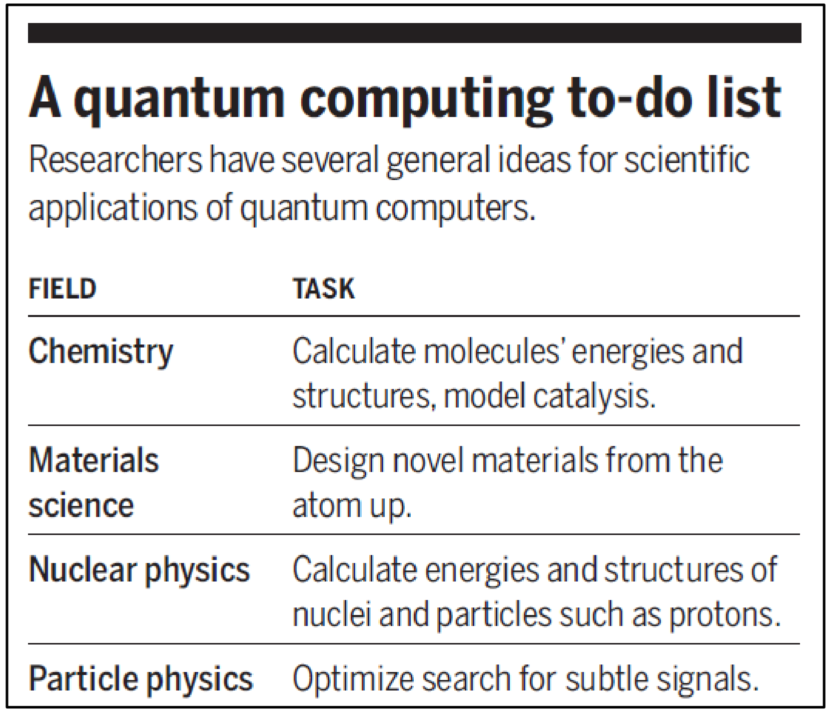
Figure from: S.K. Misra, in: Electron Spin Resonance (ESR) Based Quantum Computing. T. Takui, L. Berliner, G. Hanson (Eds.) Springer New York 2016, p. 1-23

Quantum Computing Applications

- ❑ Modeling or simulating inherently quantum mechanical processes in chemistry, materials science, nuclear physics, and particle physics
- ❑ Optimization problems (machine learning, social sciences, computational biology, weather prediction, space debris problem…)
- ❑ Integer factorization (Shor’s algorithm) – cybersecurity
- ❑ Searching an unsorted database (Grover’s algorithm)



Potential Energy Surface for BeH_2 molecule
A.Kandala et al. *Nature* **549** (2017) 242–246



A quantum computing to-do list

Researchers have several general ideas for scientific applications of quantum computers.

FIELD	TASK
Chemistry	Calculate molecules’ energies and structures, model catalysis.
Materials science	Design novel materials from the atom up.
Nuclear physics	Calculate energies and structures of nuclei and particles such as protons.
Particle physics	Optimize search for subtle signals.

A. Cho, *Science* **359**(2018)141-142

Emerging Technologies

Singularity

Anita Schwartz
Scientific Applications Consultant IV
August 10, 2019

University of Delaware

IT Client Support & Services

What is Singularity?

- Singularity is a container solution created by necessity for scientific and application driven workloads.*

- Singularity enables users to have full control of their environment to create containers that can be used to package entire scientific workflows, software and libraries, and even data.

* Due to many poorly written applications developed without the design to allow flexibility and portability for installation by others on different systems.

Why Singularity?

Singularity is different from other container solutions in it’s primary design goals and architecture:

- **Reproducible software stacks**
- **Mobility of compute**
- **Compatibility with complicated architectures**
- **Security model**

Benefits of Singularity

- Easy to use and well documented
- Repositories
- HPC application portability
- Flexibility
- Minimize support
- Overhead is minimal

Open OnDemand Web Portal for HPC

Shawn G. Doughty
Tufts University
shawn.doughty@tufts.edu

Features

- Open Source.
- NSF Funded. Based on OSC OnDemand Portal
- https://doi.org/10.21105/joss.00622
- Clientless web based access to HPC resources.
- File management, upload, download, view, edit.
- Command line access.
- Job management, multiple schedulers, slurm.
- Launch graphical desktops, applications (x11, web)

Implementation

- Practical to implement for small and large centers.
- RPM based install.
- Applicable to all research domains. Long tail of science.
- Web front end. Authentication.
- Training manual available for repurpose.
- Feedback from users has been positive.
- Web interface expected by users. Disruptive.

Community

- Website. http://ondemand.org/
- Mailing List.
- Read the Docs.
- GitHub.

Future

- Standard web based interface for HPC sites.
- Community contributions.
- Sustainable development model?
- Project ready to take off.
- Let’s help make it happen.

ACI-REF VR '18

Panel: Hadoop, MapReduce, Apache Spark, and Big Data

David Chin, PhD <dwc62@drexel.edu>
Sr. Systems Admin, Drexel University Research Computing

Outline

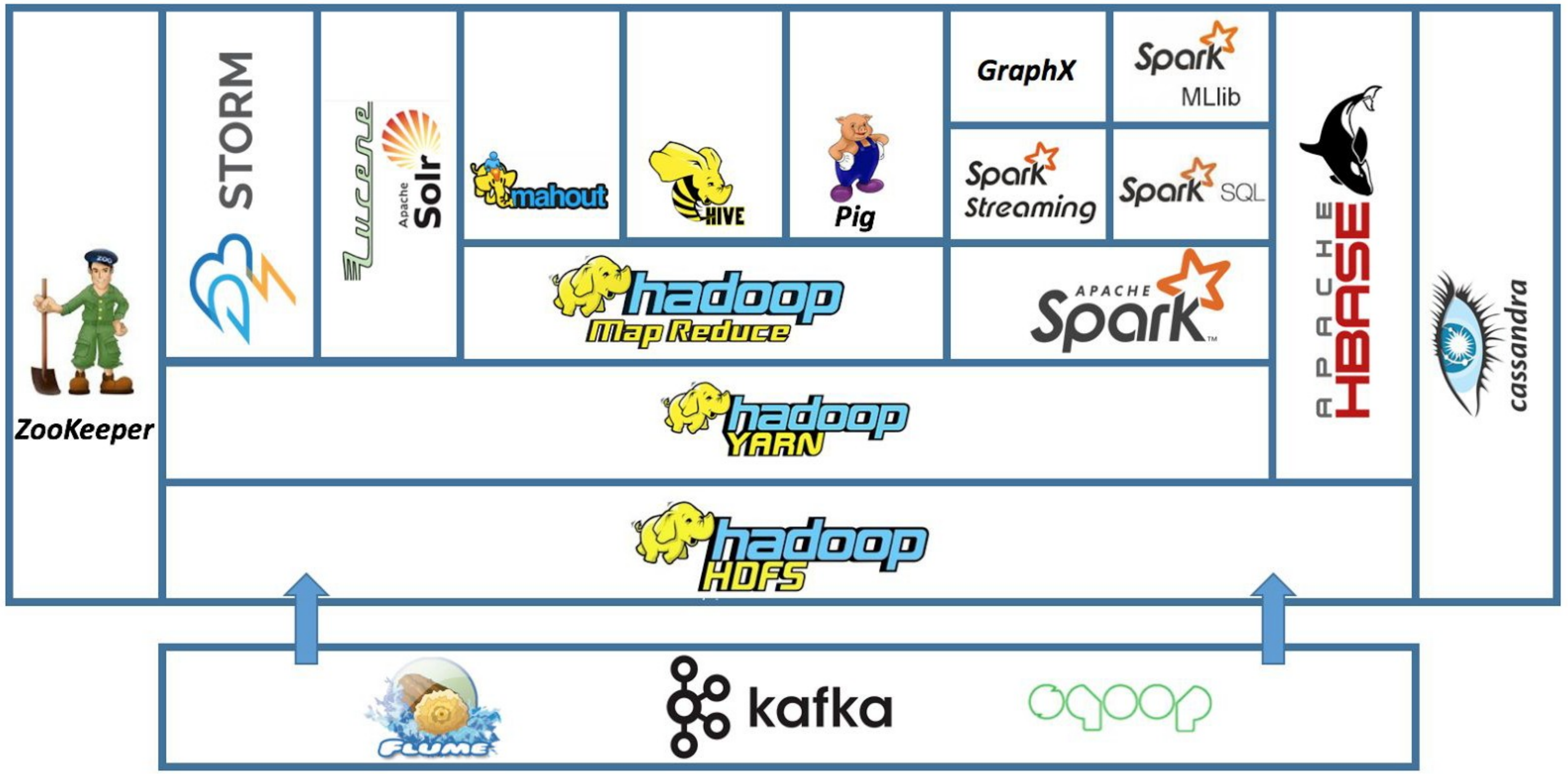
- Overview of Hadoop zoo (ecosystem)
- Outline of Hadoop technology/architecture
- Challenges in running Hadoop in traditional HPC

Overview

The HadoopZoo

- Collection of softwa
  problems involving l
  (using commodity h
- HDFS (Hadoop Dist
  - Handles distributio
- YARN
  - Resource manager -
    distribution of tasks
- MapReduce, Spark
  - Computation frame

Hadoop Zoo (Ecosystem)



Source: http://blog.newtechways.com/2017/10/apache-hadoop-ecosystem.html

Disclaimer

- I am no expert
  - Exposure via Coursera Big Data series taught by SDSC, Machine Learning series by Andrew Ng
  - Hacked a couple versions of Apache Spark to partially integrate with Univa Grid Engine
- I will drop “Apache” from the front of a bunch of the following software names

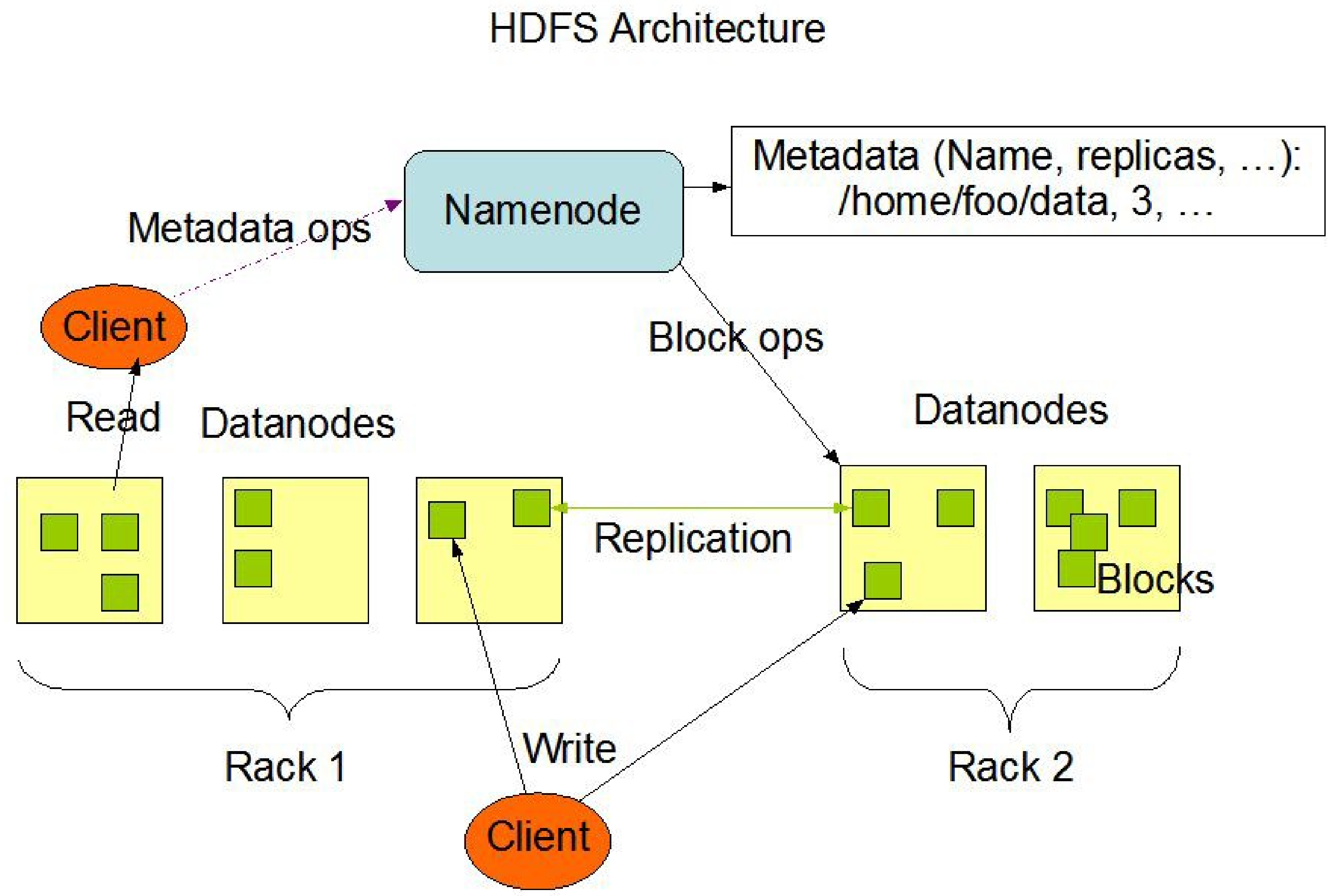
Hadoop Ecosystem

Hadoop Distributed FS (HDFS)

- Distributed, scalable, portable **non-POSIX** file system for the Hadoop framework written in Java
- BUT has shell commands and API that provide similar functionality to “real” filesystems
- Splits data to the local disks on multiple nodes
    - Allows handling of “big data”
    - Designed for immutable files; may not handle concurrent writes

HDFS Architecture

- master/slave architecture
  - Single **NameNode**, a master server that manages the file system namespace and regulates access to files by clients
  - A number of **DataNodes**, usually one per node in the cluster, which manage storage attached to the nodes that they run on
  - HDFS exposes a file system namespace and allows user data to be storedin files.
  - Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes.



Source: https://hadoop.apache.org/docs/r3.1.1/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

HDFS on the command line

Example HDFS shell operations:

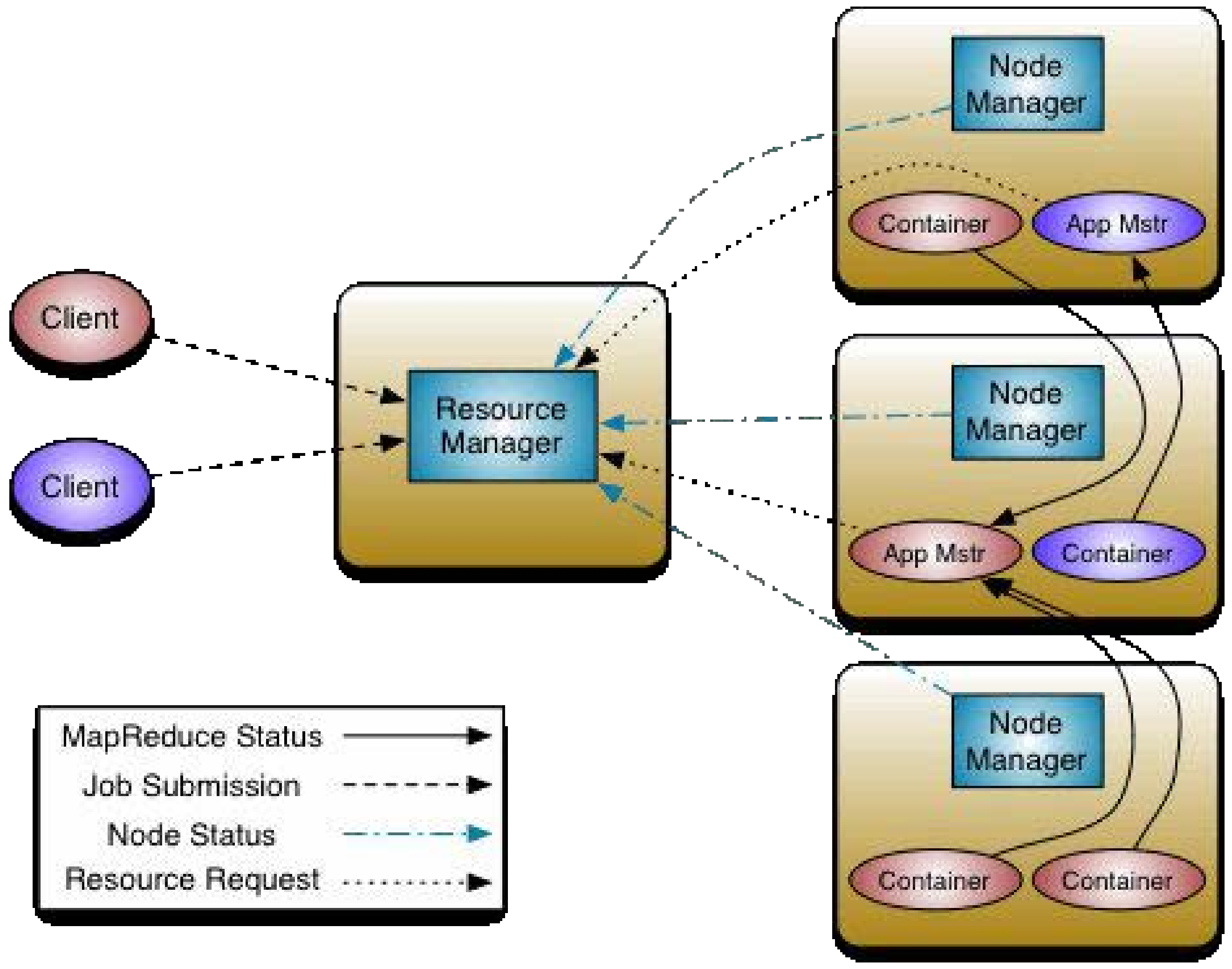
```
$ hadoop fs -mkdir /hdfs/myname/foo

$ hadoop fs -put ~/bigdata.csv /hdfs/myname/foo

$ hadoop fs -ls /hdfs/myname/foo
```

YARN (Yet Another Resource Negotiator)

- Resource manager and job scheduler
  - This is one of the sticking points in integrating the Hadoop ecosystem into typical HPC clusters
  - Tasks may take different amounts of time => YARN coordinates
- Scheduler has a pluggable policy, e.g.
  - CapacityScheduler
  - FairScheduler
- Job submission done within the code for computation (example to follow)



Source: https://hadoop.apache.org/docs/r3.1.1/hadoop-yarn/hadoop-yarn-site/YARN.html

MapReduce

- "MapReduce: Simplified Data Processing on Large Clusters", J. Dean and S. Ghemawat (2004)
- For those familiar with MPI, this is similar to scatter-gather pattern
- Map
  - Applies operation(s) on distributed data
- Reduce
  - Collects distributed results and aggregates them in some way

MapReduce example

From the MapReduce Tutorial word count example. Only snippets.

```
public class WordCount {
    public static class TokenizerMapper extends Mapper<...> {
        ...
    }
    public static class IntSumReducer extends Reducer<...> {
        ...
    }
    //  ... continued ...
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

And run:

```
$ bin/hadoop jar wc.jar WordCount \
/user/joe/wordcount/input /user/joe/wordcount/output
```

How would Hadoop fit into typical HPC cluster?

With some kluges

Hadoop is the whole cluster

- Hadoop is meant to control the entire cluster
  - HDFS handles the distributed storage
  - YARN handles the distribution of tasks to available storage/compute resources
  - MapReduce or Spark jobs are submitted to the cluster

Parallel file system

- What if you already have a parallel file system?
  - There are replacements for HDFS
  - Intel Lustre (RIP), now Whamcloud again: Hadoop Adapter for Lustre
    https://github.com/whamcloud/lustre-connector-for-hadoop
    - Assumes Hadoop 2.3.0 installed non-securely (i.e. run as root)
  - CephFS: Hadoop CephFS Plugin
    http://docs.ceph.com/docs/mimic/cephfs/hadoop/
    - Requires Hadoop 1.1.x stable series; latest Hadoop is 3.1.1
  - GPFS: IBM Spectrum Scale
    https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.1.1/com.ibm.spectrum.scale.v4r11.adv.doc/bl1adv_hadoop.htm

Resource mgmt & job scheduling

- What if you already have a resource manager & job scheduler?
    - Generally, solution is to take the HPC job, and launch an ad hoc Hadoop cluster on the nodes assigned to that job
- Univa Grid Engine has Hadoop integration (for Cloudera Distribution for Hadoop (CDH) 3) http://www.gridengine.eu/gridengineaddons/86-cloudera-slides-about-univa-grid-engine-uge-hadoop-integration

- SLURM etc.
  - LLNL Magpie https://github.com/LLNL/magpie
    - Set of bash scripts etc. to run Hadoop zoo in HPC
    - Supports: Hadoop, Spark, Hbase, Storm, Pig, Mahout, Phoenix, Kafka, Tachyon, Zeppelin, and Zookeeper
    - Supports: Slurm, Moab, Torque, and LSF
    - Supports: Lustre, or other shared file system
  - Harvard Faculty of Arts & Sciences Research Computing (FASRC) https://github.com/fasrc/hpc-hadoop-mapreduce (old - Nov 2013)
    - One big bash script; works for SLURM & LSF; ignores HDFS for existing file system

- Slurm etc. (continued)
    - OpenSFS Lustre User Group 2013 talk by Intel (Castain, Kulkarni, and Xu) “MapReduce and Lustre: Running Hadoop in a HPC Environment” http://cdn.opensfs.org/wp-content/uploads/2013/10/CLUG2013_Running-Hadoop_final.pdf
        - Intel Distribution of Hadoop + Proprietary Hadoop-JSON Translator + SLURM JSON API
        - Also talks about Lustre replacing HDFS

Apache Spark

What is Apache Spark?

- Apache Spark is a fast and general-purpose cluster computing system https://spark.apache.org/docs/latest/index.html
  - Not very informative
- Key difference between Hadoop MapReduce and Spark:
  - Hadoop MapReduce reads from and writes to node-local disk
  - Apache Spark does it in memory, and writes only at the end
- Also, Apache Spark has a standalone mode (no HDFS, no YARN) which is the basis of the following kluge

Spark in standalone mode

- https://spark.apache.org/docs/latest/spark-standalone.html
- No YARN (or Mesos)
  - Start master and worker daemons by hand, or by using launch (bash) scripts

Spark in Univa Grid Engine

- Kluge
  - Modify launch scripts to read job environment to infer nodes being used, job & scratch dirs
  - I have done this for 2 different versions of Spark to run in Univa, **but** it only works single node (i.e. not very big data)
    - Write parallel environment (PE) scripts to read the node list, job directory, scratch directory, and set the appropriate environment variables used by the Spark launch scripts
    - https://github.com/prehensilecode/spark/blob/prehensilecode/sbin/pescript_spark2start.sh

Conclusion

1. Multiple solutions for using Hadoop frameworks in HPC: various resource managers & schedulers, various file systems
2. Klugey, and may not support latest Hadoop

Questions

Questions

How practical will it be to implement your technology topic in an HPC/CI center with 5-10 staff?

Questions

In which research domains is this technology most applicable?

Questions

How can smaller HPC/CI centers gain access to this technology?

Questions

How easily can the training material for these technologies be implemented by CI center staffs of various sizes?

Questions

What is your technology topic’s potential for disruption?

Could it create a technology research divide?