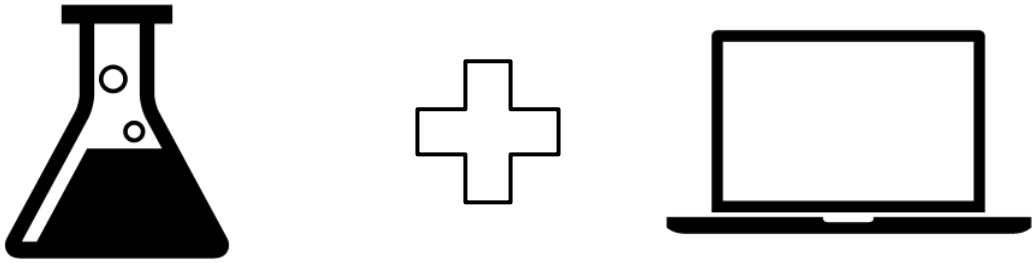
{Scientific,
Distributed,
Cloud}
Computing

# Scientific Computing

# Science Meets Computing



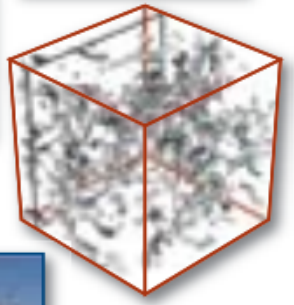
Using **computation** to support **scientific research** and **exploration**.

# Third and Fourth Paradigms

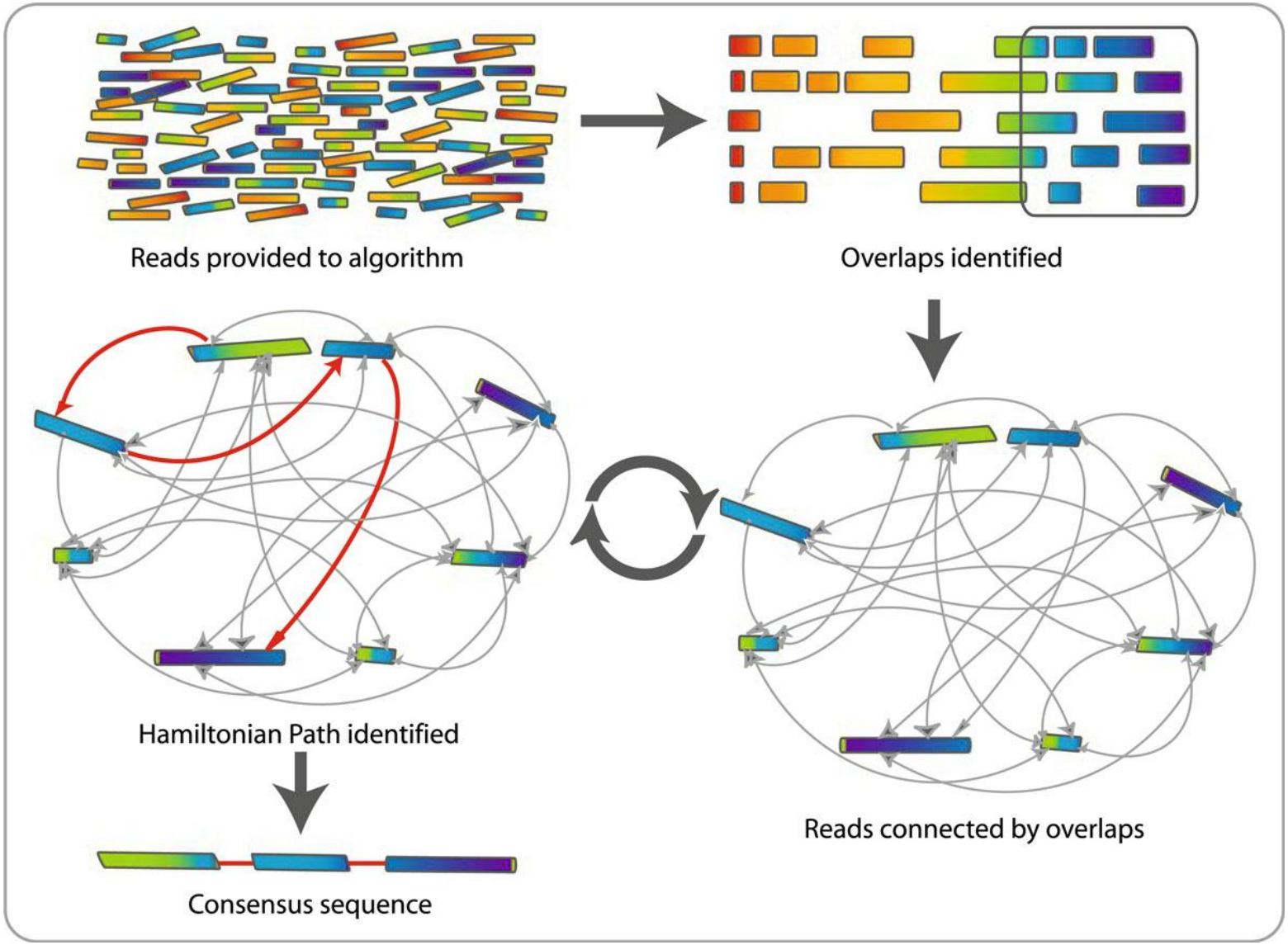
## Science Paradigms

- Thousand years ago:
  science was **empirical**
  *describing natural phenomena*
- Last few hundred years:
  **theoretical** branch
  *using models, generalizations*
- Last few decades:
  a **computational** branch
  *simulating complex phenomena*
- Today: **data exploration** (eScience)
  *unify theory, experiment, and simulation*
  - Data captured by instruments or generated by simulator
  - Processed by software
  - Information/knowledge stored in computer
  - Scientist analyzes database/files using data management and statistics

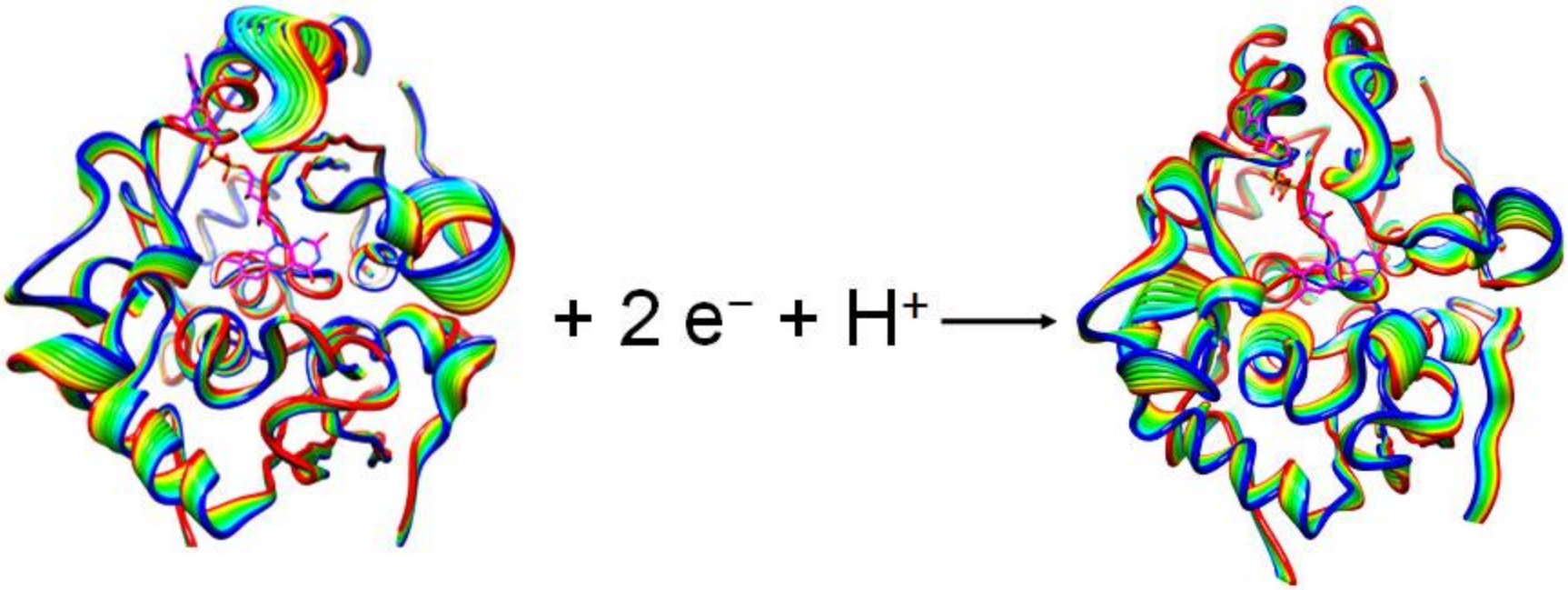
$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{4\pi G p}{3} - \mathrm{K}\frac{c^2}{a^2}$$



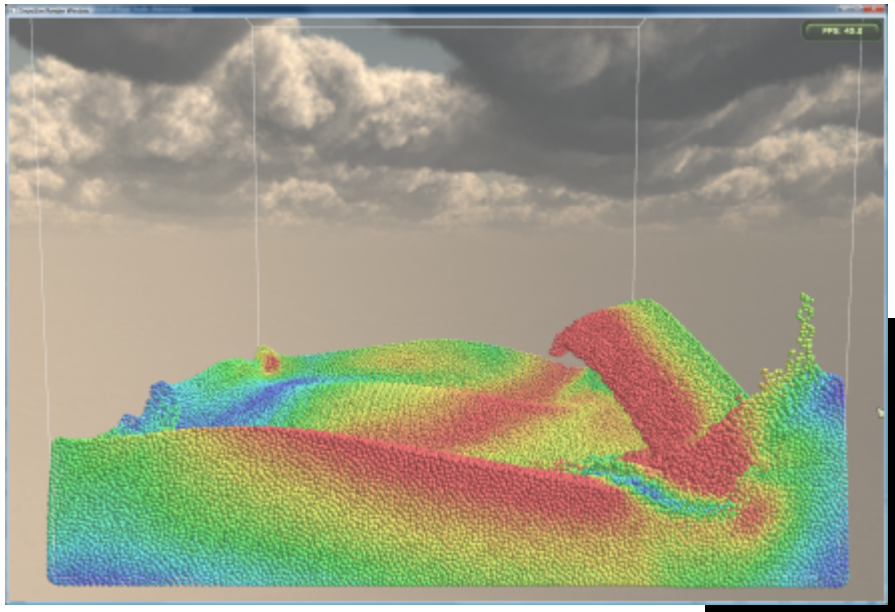
# Example: Genome Assembly

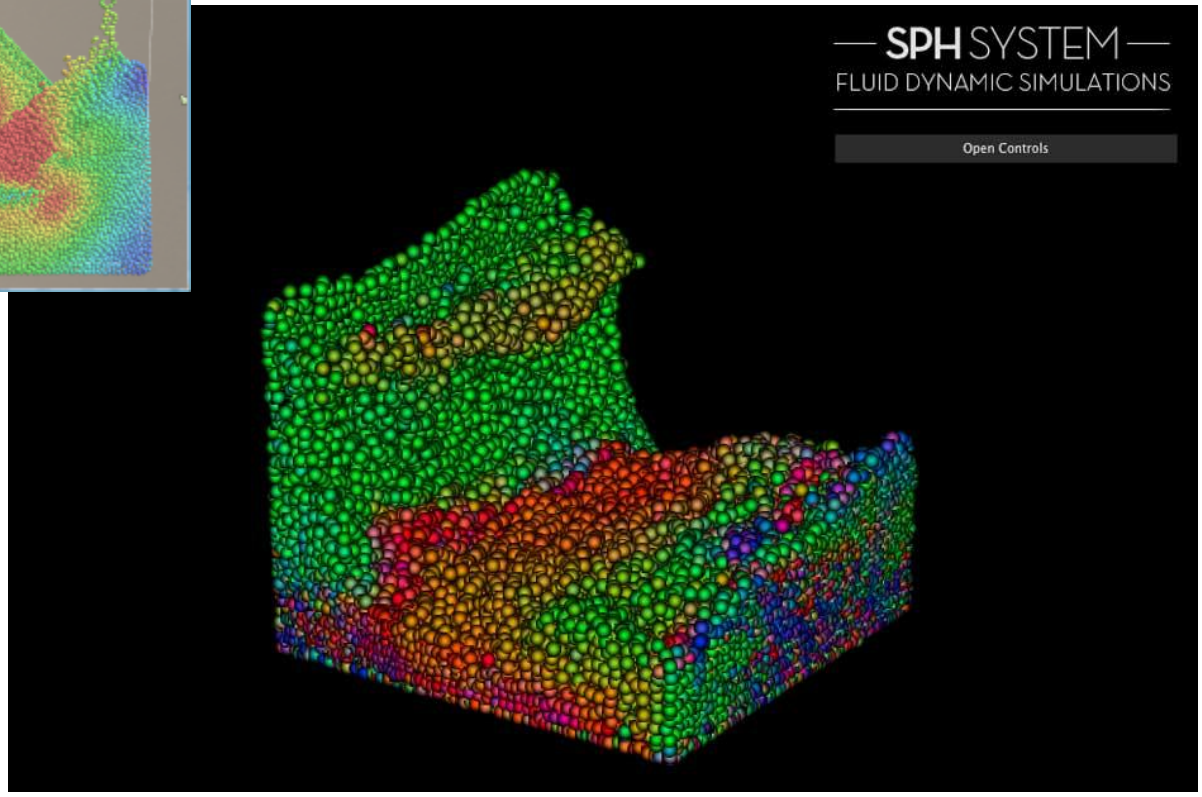


# **Example:** Molecular Dynamics

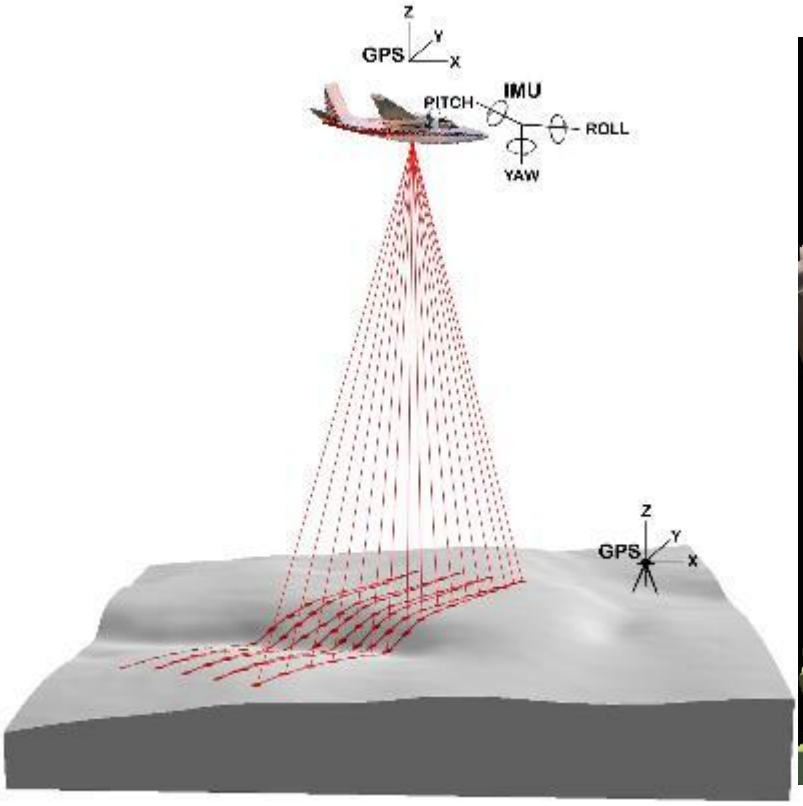


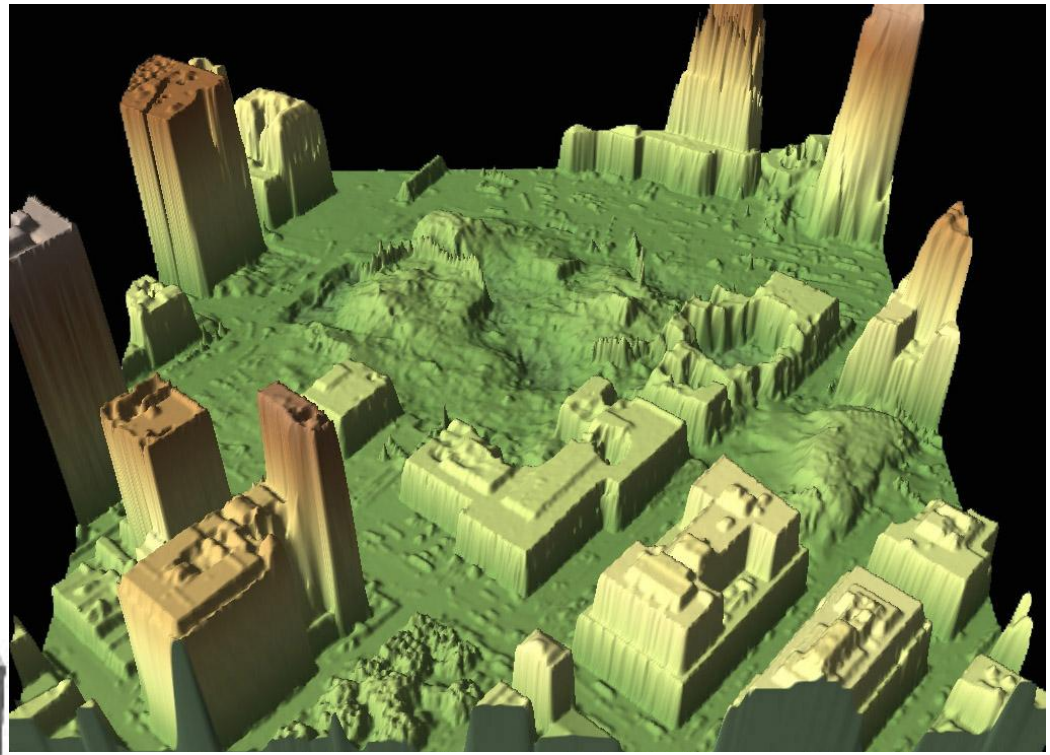
# Example: Particle Simulation



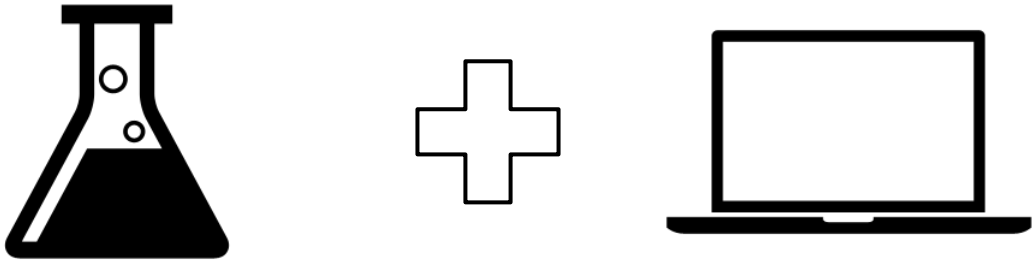


# Example: LIDAR



# Science Needs Computing



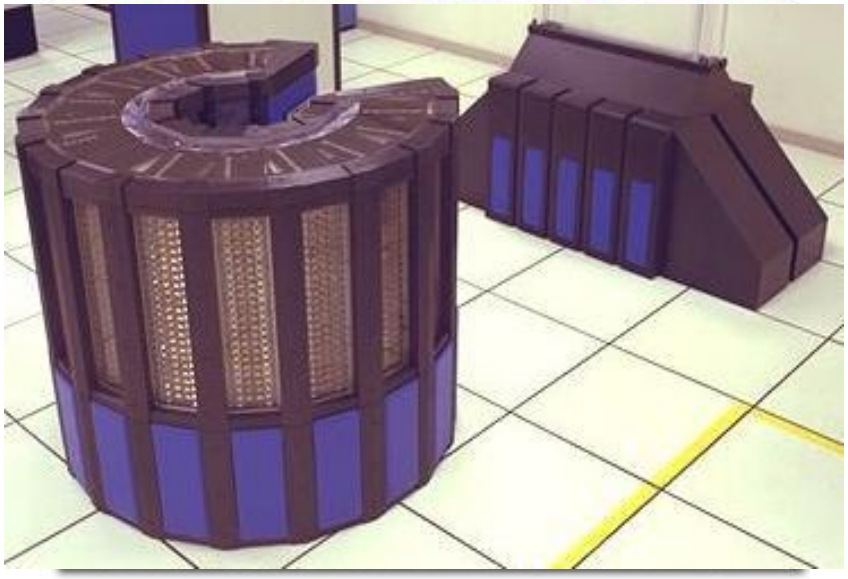
**Computation** not only **expedites scientific research**, but also makes certain **scientific inquiries possible**!

# Distributed Computing

# Oxen vs Chickens

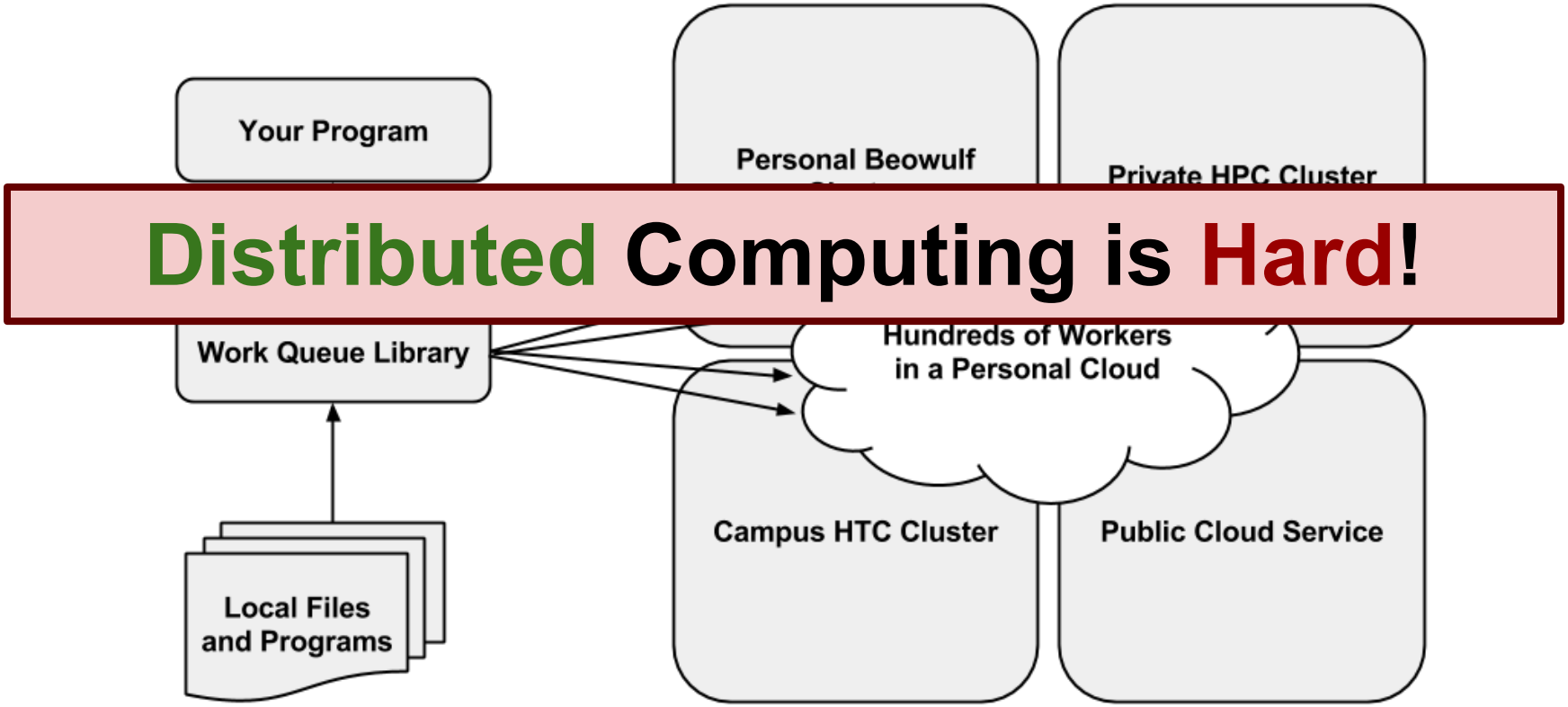
If you are plowing a field, which would you rather use?

**Two strong Oxen** or **1024 Chickens?**



# Distributed Computing

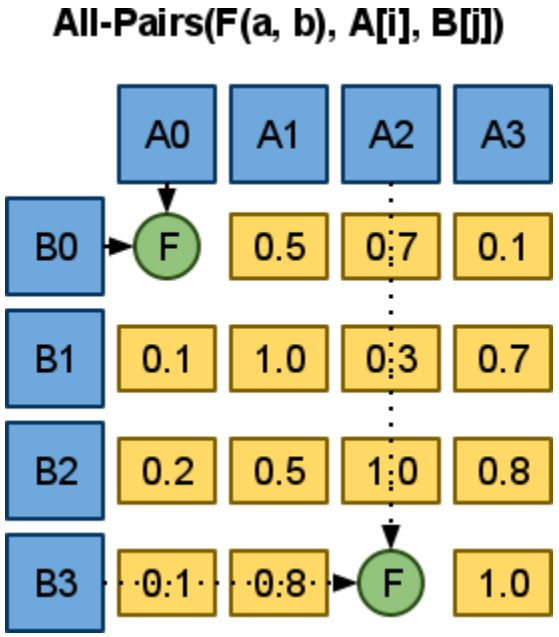
Applications can increase **throughput** by executing multiple tasks simultaneously on **different machines**




**Distributed Computing is Hard!**

# Abstractions: Common Patterns

**Structured** way of combining **small executables** into **parallel** graphs that can be **scaled** up to large sizes.




## Examples

All-Pairs, Wavefront, Map-Reduce

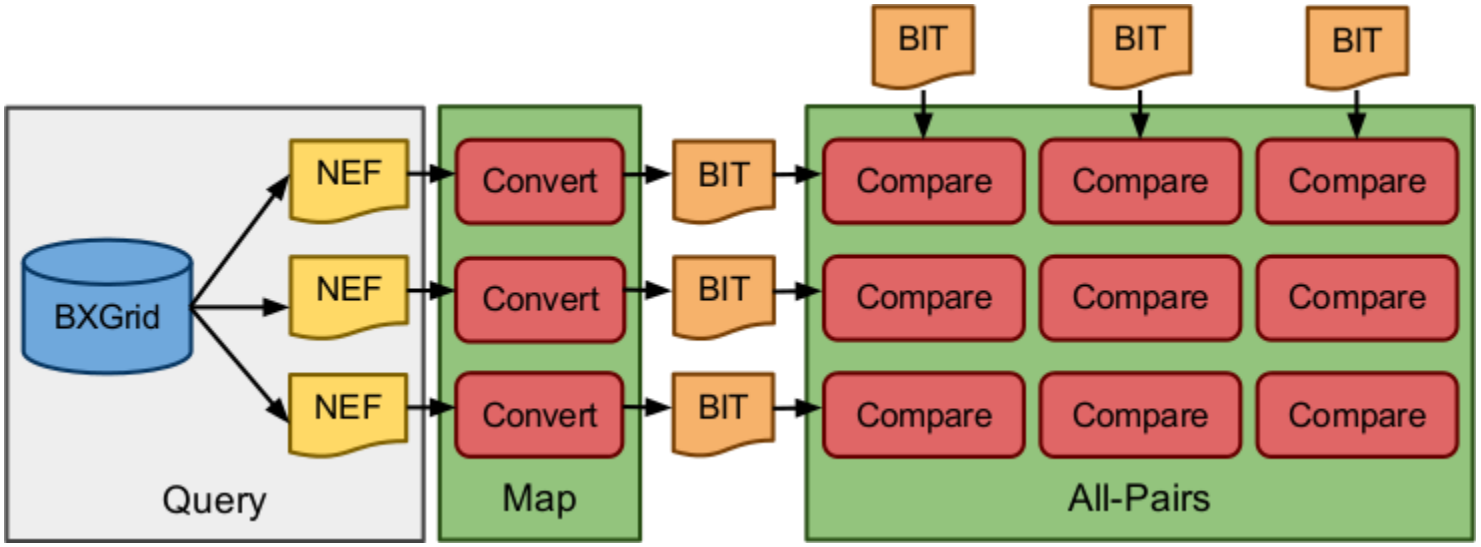
## Advantages

- Simple programming interface
- Hides details of distributed system

## Disadvantages

- Only addresses one type of computation
- Difficult to implement large sophisticated applications

# Biometrics Experiment: Overview




1. **Query:** Select and extract data from scientific repository
2. **Transcode:** Convert image data to new format suitable for analysis
3. **Comparison:** Perform All-Pairs computation on intermediate image data

# Workflows: Graphs

Organize computation as a **directed-acyclic graph** (**DAG**)

## Examples

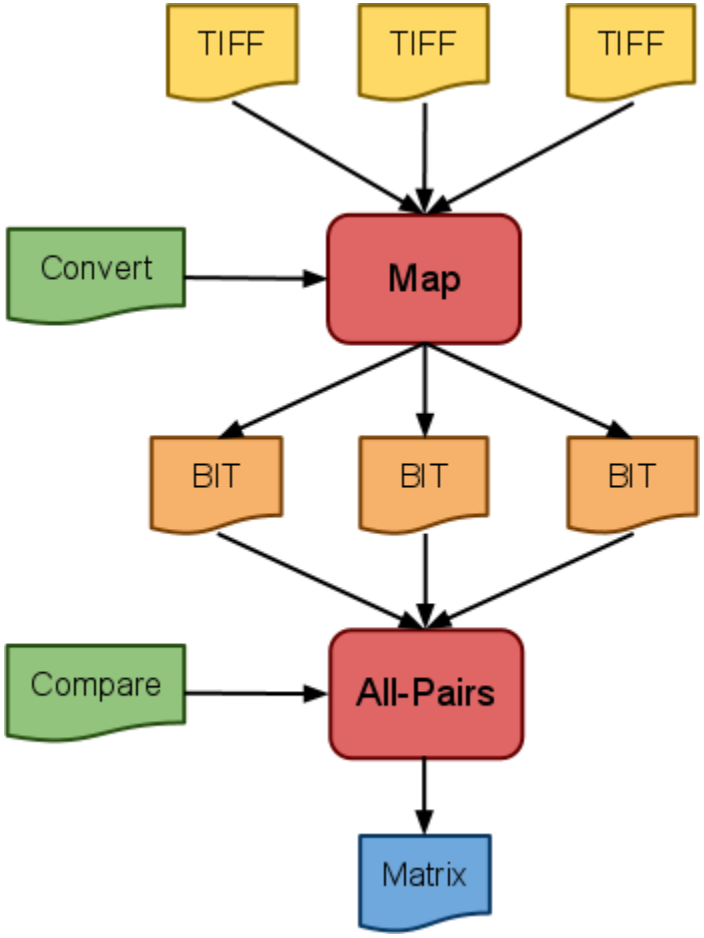
Pegasus, DAGMan, Dryad, Makeflow

## Advantages

- Exploit natural concurrency
- Program large applications
- Embed/implement abstracts in DAG

## Disadvantages

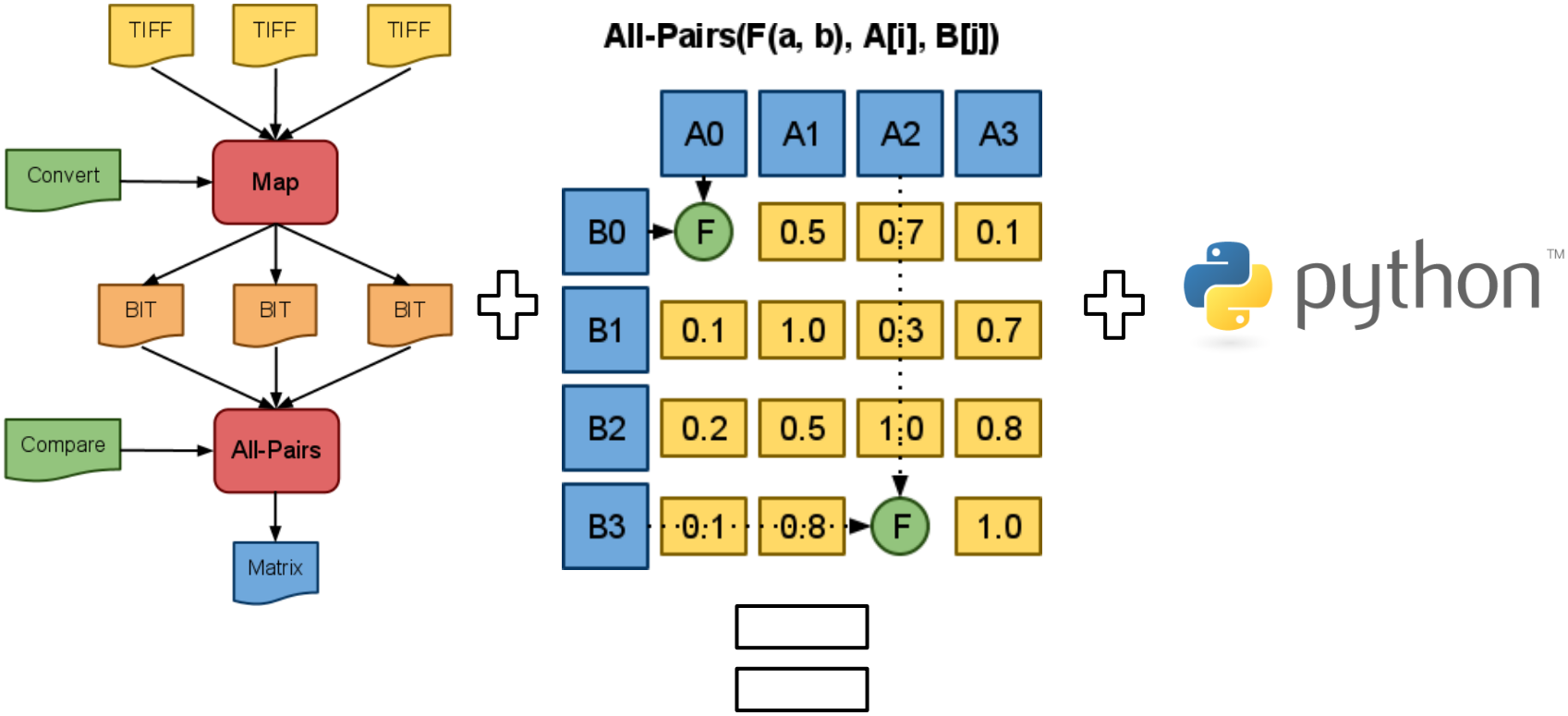
- Tedious, difficult to construct DAGs
- Too low level



# Biometrics Experiment: DAG


***Large workflows require many nodes!***

# Weaver: Workflow Compiler




***Simplified Distributed Programming!***

# Weaver: Features

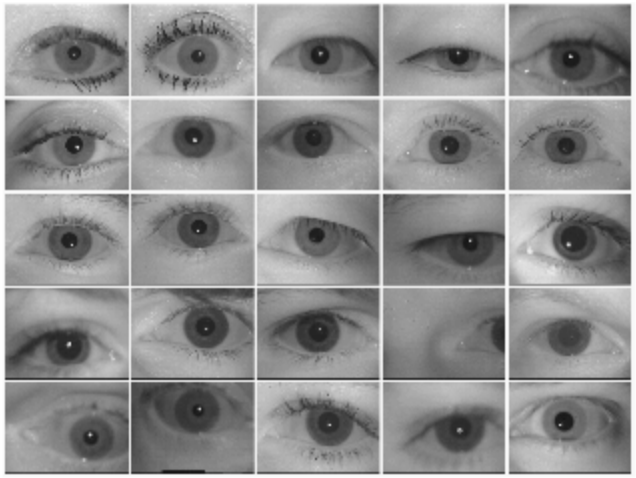
Weaver is a high-level compiler framework that allows users to construct distributed workflows

- Built on top of **Python** programming language
- Enable users to combine **abstractions** to construct workflows
- Applies various **compiler techniques** to workflow construction
- Includes additional utilities such as **linkers** and **profilers** to provide complete programming toolchain

# Biometrics Experiment: Weaver

```
 1  db      = MySQLDataset('db', 'biometrics', 'irises')
 2  irises  = Query(db, db.c.state == 'Enrolled',
 3                      Or(db.c.color == 'Blue',
 4                         db.c.color == 'Green'))
 5
 6  convert = ParseFunction(
 7              'convert_iris_to_template {IN} {OUT}')
 8  compare = ParseFunction(
 9              'compare_iris_templates {IN} > {OUT}')
10
11  bits    = Map(convert, irises, '{BASE_WOEXT}.bit')
12  results = AllPairs(compare, bits, bits)
13  table   = Merge(results, 'table.txt')
```

# Transcoding Workflow



# Weaver: Contribution

**DAGs** are the **assembly language** of distributed computing:

Provide mechanism for construction and executing large distributed applications

**Abstractions** are the **SIMD** instructions:

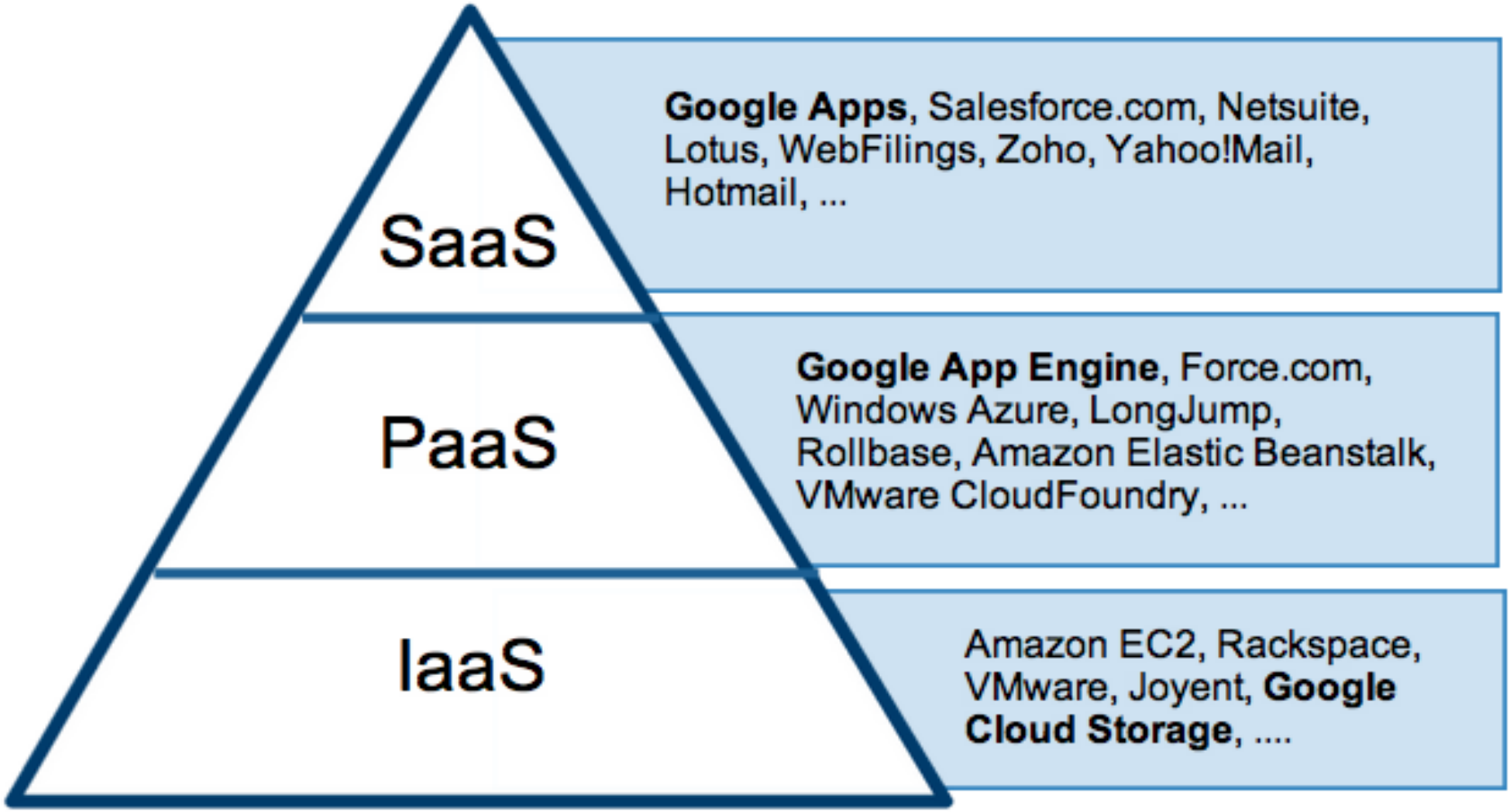
Provide powerful and compact way to express a common pattern of computation

*We need a **compiler** that allows us to take advantage of both in building large **distributed applications**!*

# Cloud Computing

# Cloud Computing

## Cloud Computing as Gartner Sees It




Source: Gartner AADI Summit Dec 2009

# **Renting** (aka Out-sourcing)

**Renting** resources and services (including a data center or compute cluster!)

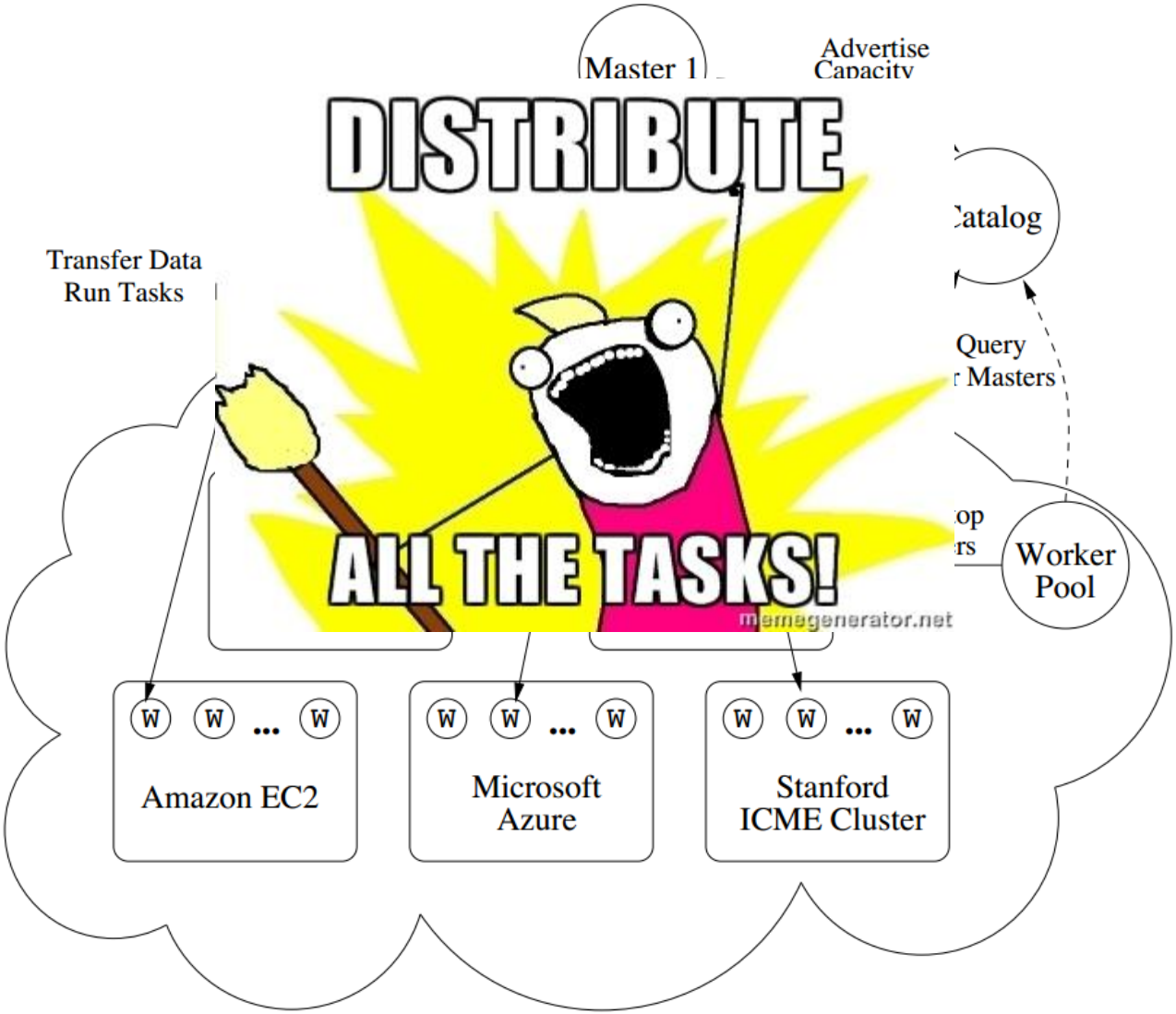
- **Scalability:**
  Ability to add more resources in order to increase performance

- **Elasticity:**
  Ability to add and remove resources on-demand

***Made possible due to virtualization***

# Rent-a-Supercomputer

# Hybrid Computing

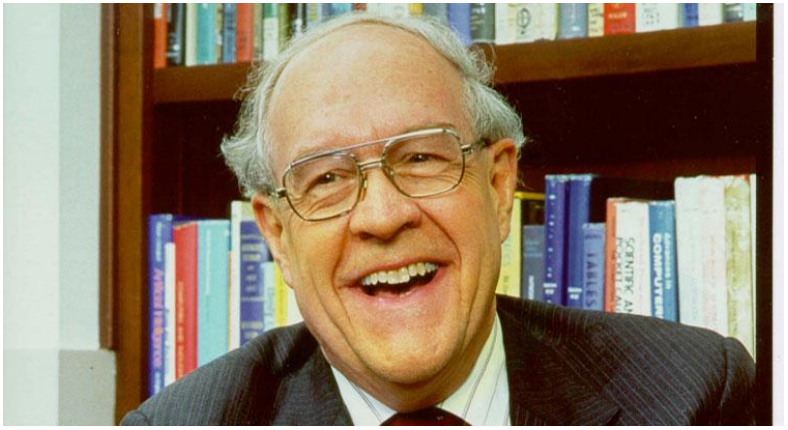


# Closing

# Computer Scientist == Toolsmith

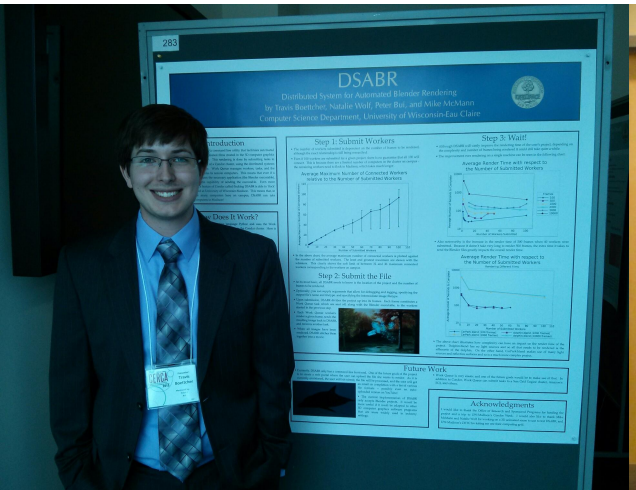
*"If the **computer scientist** is a **toolsmith**, and if our delight is to fashion power tools and amplifiers for minds, we must partner with those who will use our tools, those whose intelligences we hope to amplify."*

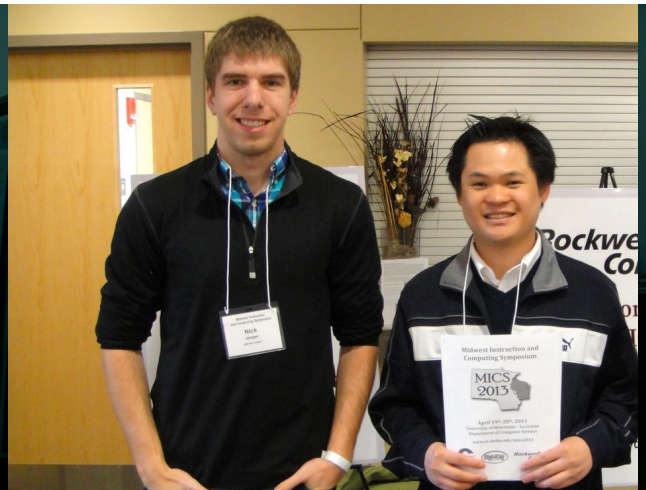
**- Fred Brooks**

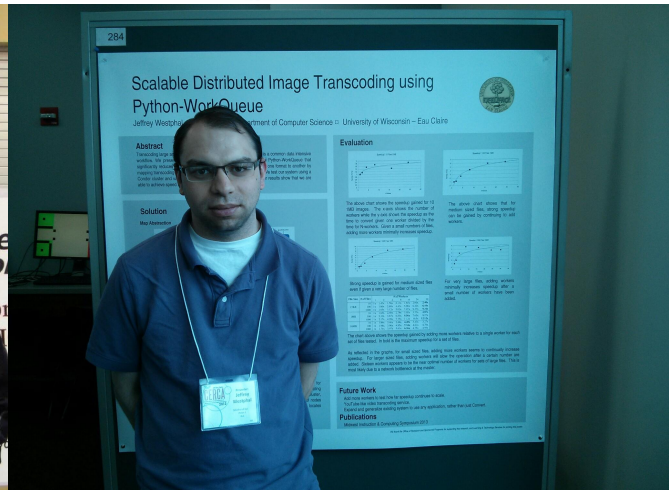


# Enlist Now!

- **Undergraduate Research**
  - Independent Study
  - Funded Research
  - Just for kicks!

# Questions?


**Peter Bui**

**Office:** Phillips 131

**EMail:** buipj@uwec.edu

**WWW:** http://cs.uwec.edu/~buipj

**IRC:** pnutzh4x0r