The Principles of HTC

Miron Livny
Wisconsin Institutes for Discovery
University of Wisconsin-Madison

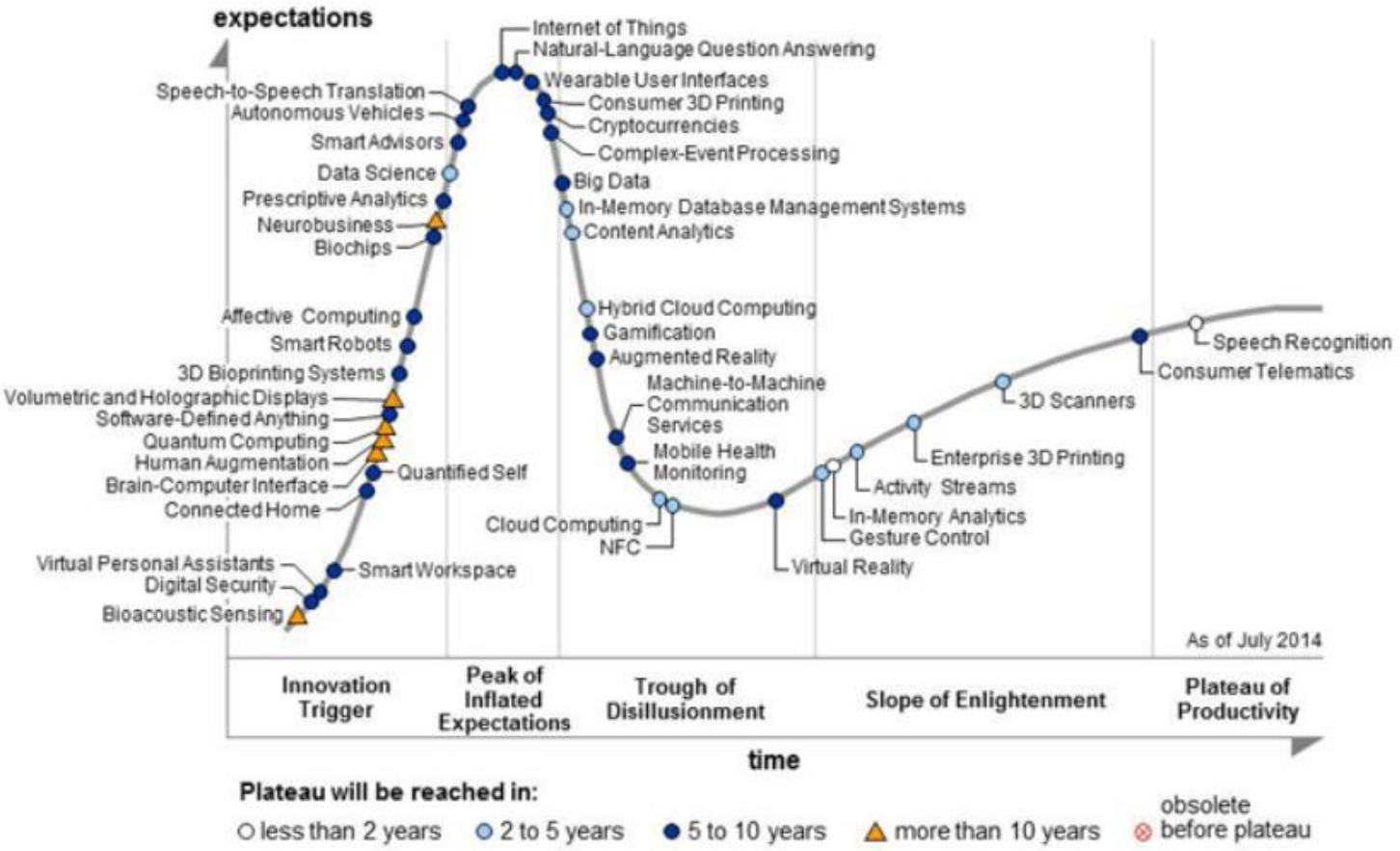
The words of Koheleth son of David, king in Jerusalem ~ 200 A.D.

Only that shall happen
Which has happened,
Only that occur
Which has occurred;
There is nothing new
Beneath the sun!

Ecclesiastes, (,קֹהֶלֶת*Kohelet*, "son of David, and king in Jerusalem" alias Solomon, Wood engraving Gustave Doré (1832–1883)

Ecclesiastes Chapter 1 verse 9

We are driven by Principals (– Hype)



Source: Gartner (August 2014)

Perspectives on Grid Computing

(2010)

Uwe Schwiegelshohn Rosa M. Badia Marian Bubak Marco Danelutto Schahram Dustdar Fabrizio Gagliardi Alfred Geiger Ladislav Hluchy Dieter Kranzlmüller Erwin Laure Thierry Priol Alexander Reinefeld Michael Resch Andreas Reuter Otto Rienhoff Thomas Rüter Peter Sloot Domenico Talia Klaus Ullmann Ramin Yahyapour Gabriele von Voigt

We should not waste our time in redefining terms or key technologies: clusters, Grids, Clouds... What is in a name? Ian Foster recently quoted Miron Livny saying: "I was doing Cloud computing way before people called it Grid computing", referring to the ground breaking Condor technology. It is the Grid scientific paradigm that counts!

The paradigm shift of 70's – computing hardware packaged and sold in small units

Claims for "benefits" provided by Distributed Processing Systems

P.H. Enslow, *"What is a Distributed Data Processing System?"* Computer, January 1978

- High Availability and Reliability
- High System Performance
- Ease of Modular and Incremental Growth
- Automatic Load and Resource Sharing
- Good Response to Temporary Overloads
- Easy Expansion in Capacity and/or Function

Definitional Criteria for a Distributed Processing System

P.H. Enslow and T. G. Saponas *""Distributed and Decentralized Control in Fully Distributed Processing Systems"* Technical Report, 1981

- Multiplicity of resources
- Component interconnection
- **Unity of control**
- System transparency
- **Component autonomy**

Unity of Control

All the component of the system should be **unified** in their desire to achieve a **common goal**. This goal will determine the rules according to which each of these elements will be controlled.

Component autonomy

The components of the system, both the logical and physical, should be **autonomous** and are thus afforded the ability to refuse a request of service made by another element. However, in order to achieve the system's goals they have to interact in a **cooperative** manner and thus adhere to a common set of policies. These policies should be carried out by the control schemes of each element.

It is always a tradeoff

In 1983 I wrote a Ph.D. thesis –

"Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems"

http://www.cs.wisc.edu/condor/doc/livny-dissertation.pdf

When each resource has a it's own queue, when should I stay and wait and when should I move to another queue?

" … Since the early days of mankind the primary motivation for the establishment of *communities* has been the idea that by being part of an organized group the capabilities of an individual are improved. The great progress in the area of inter-computer communication led to the development of means by which stand-alone processing sub-systems can be integrated into multi-computer *'communities'*. … "

Miron Livny, " ***Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems*****.",**
Ph.D thesis, July 1983.

In 1985 we extended the scope of the distributed load balancing problem to include "ownership" of resources

Should I share and if I do with whom and when?

Now you have customers who are consumers, providers or both

What Did We Learn From Serving a Quarter of a Million Batch Jobs on a Cluster of Privately Owned Workstations

1992

Miron Livny

Computer Sciences Department
University of Wisconsin — Madison
Madison, Wisconsin
{miron@cs.wisc.edu}

User Prospective

- Maximize the capacity of resources accessible via a single interface
- Minimize overhead of accessing remote capacity
- Preserve local computation environment

Submit Locally and run Globally

(Here is the work and here are the resources I bring to the table)

In 1996 I introduced the distinction between High **Performance** Computing (HPC) and High **Throughput** Computing (**HTC**) in a seminar at the NASA Goddard Flight Center in and a month later at the European Laboratory for Particle Physics (CERN). In June of 1997 HPCWire published an interview on High Throughput Computing.

HIGH THROUGHPUT COMPUTING: AN INTERVIEW WITH MIRON LIVNY 06.27.97
by Alan Beck, editor in chief HPCwire

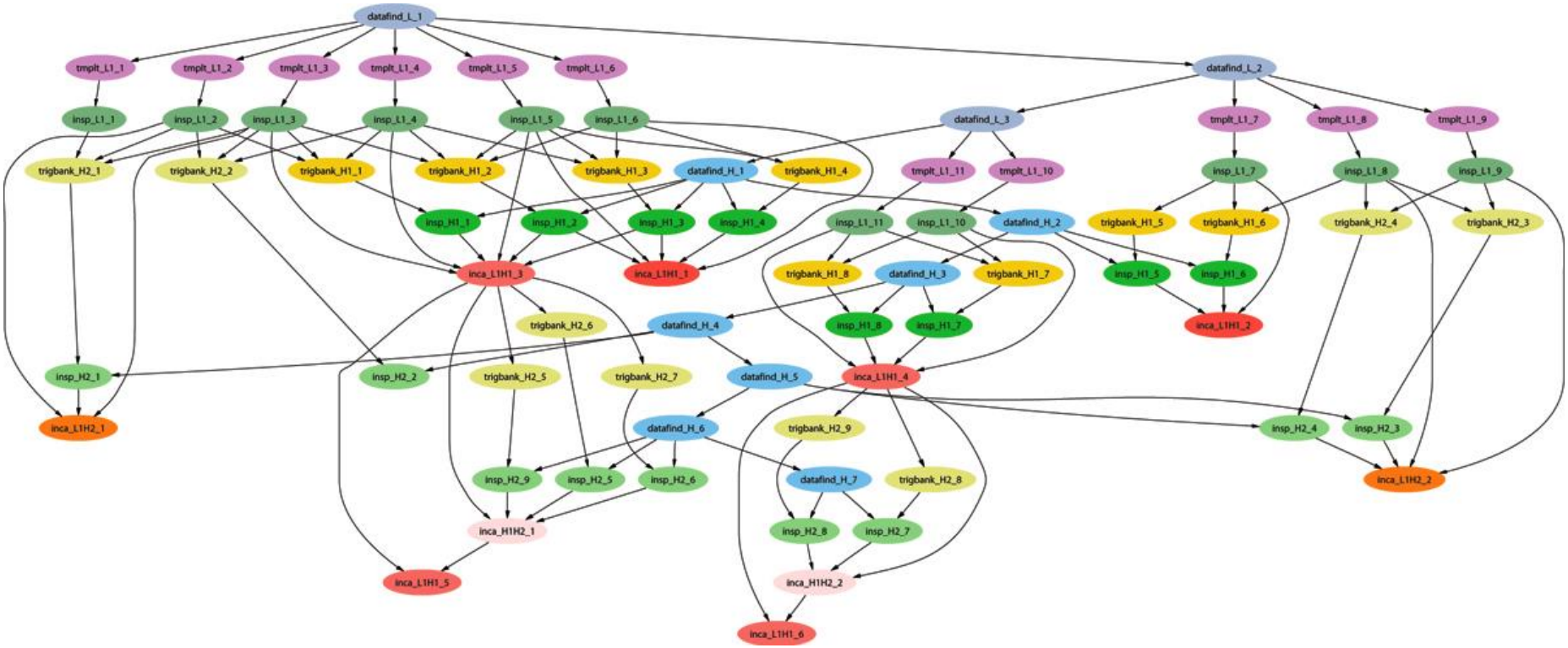
This month, NCSA's (National Center for Supercomputing Applications) Advanced Computing Group (ACG) will begin testing Condor, a software system developed at the University of Wisconsin that promises to expand computing capabilities through efficient capture of cycles on idle machines. The software, operating within an HTC (High Throughput Computing) rather than a traditional HPC (High Performance Computing) paradigm, organizes machines

High Throughput Computing is a 24-7-365 activity and therefore requires automation

FLOPY ≠ (60*60*24*7*52)*FLOPS

Using Directed Acyclic Graphs (DAGs) to support declarative automation of interdependent tasks

Example of a LIGO Inspiral DAG (Workflow)



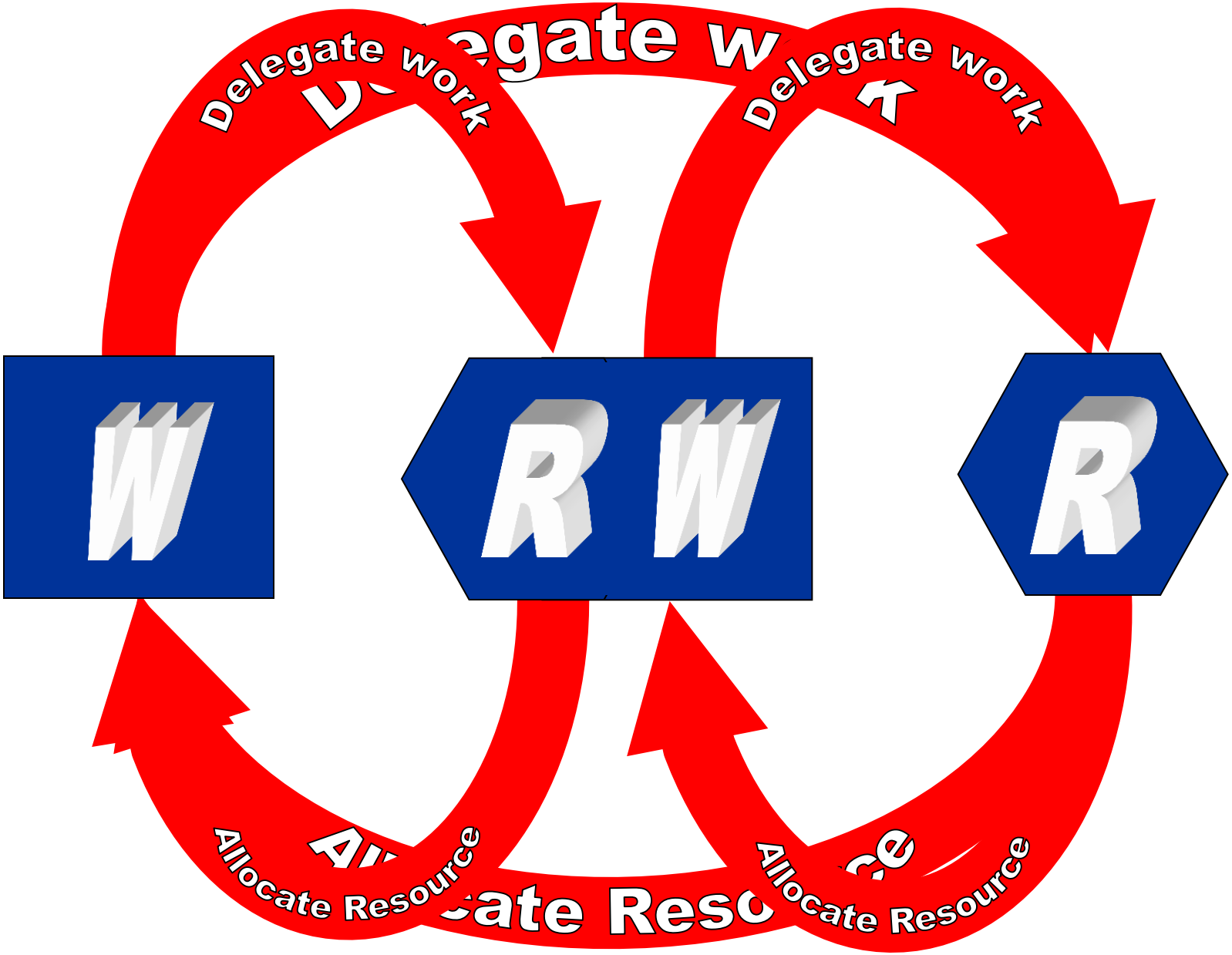
From: Stuart Anderson <anderson@ligo.caltech.edu>
Date: February 28, 2010 11:51:32 PM EST
To: Condor-LIGO mailing list <condorligo@aei.mpg.de>
Subject: [CondorLIGO] Largest LIGO workflow

Here are some numbers you ask about for LIGO's use of DAGs to manage large data analysis tasks:

1) DAG Instance--one condor_dagman process: **196,862**.
2) DAG Workflow--launched from a single condor_submit_dag but may include multiple automatic sub- or spliced DAGs: **1,120,659**.
3) DAG Analysis--multiple instances of condor_submit_dag to analyze a common dataset with results combined into a single coherent scientific result: **6,200,000**.
4) DAG Total--sum over all instances of condor dagman run: **O(100M)**.

P.S. These are lower bounds as I did not perform an exhaustive survey/search, but they are probably close.

Resource Allocation
(resource -> job)
vs.
Work Delegation
(job -> resource)


Delegate work
Delegate work
Allocate Resource
Allocate Resource
W
R W
R

Resource Allocation

A limited assignment of the "ownership" of a resource

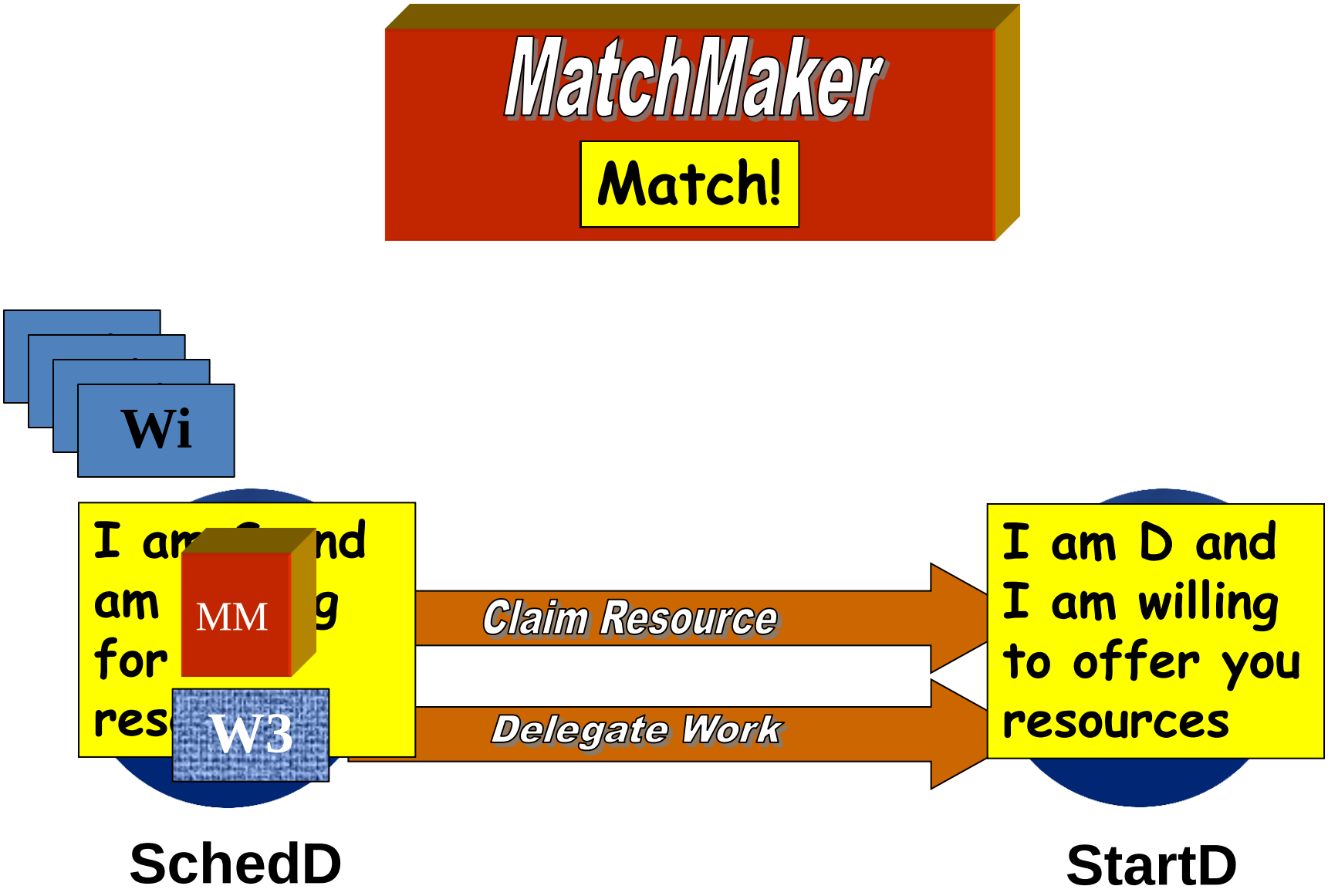
- Owner is charged for allocation regardless of actual consumption
- Owner can allocate resource to others
- Owner has the right and means to revoke an allocation
- Allocation is governed by an "agreement" between the client and the owner
- Allocation is a "lease"
- Tree of allocations

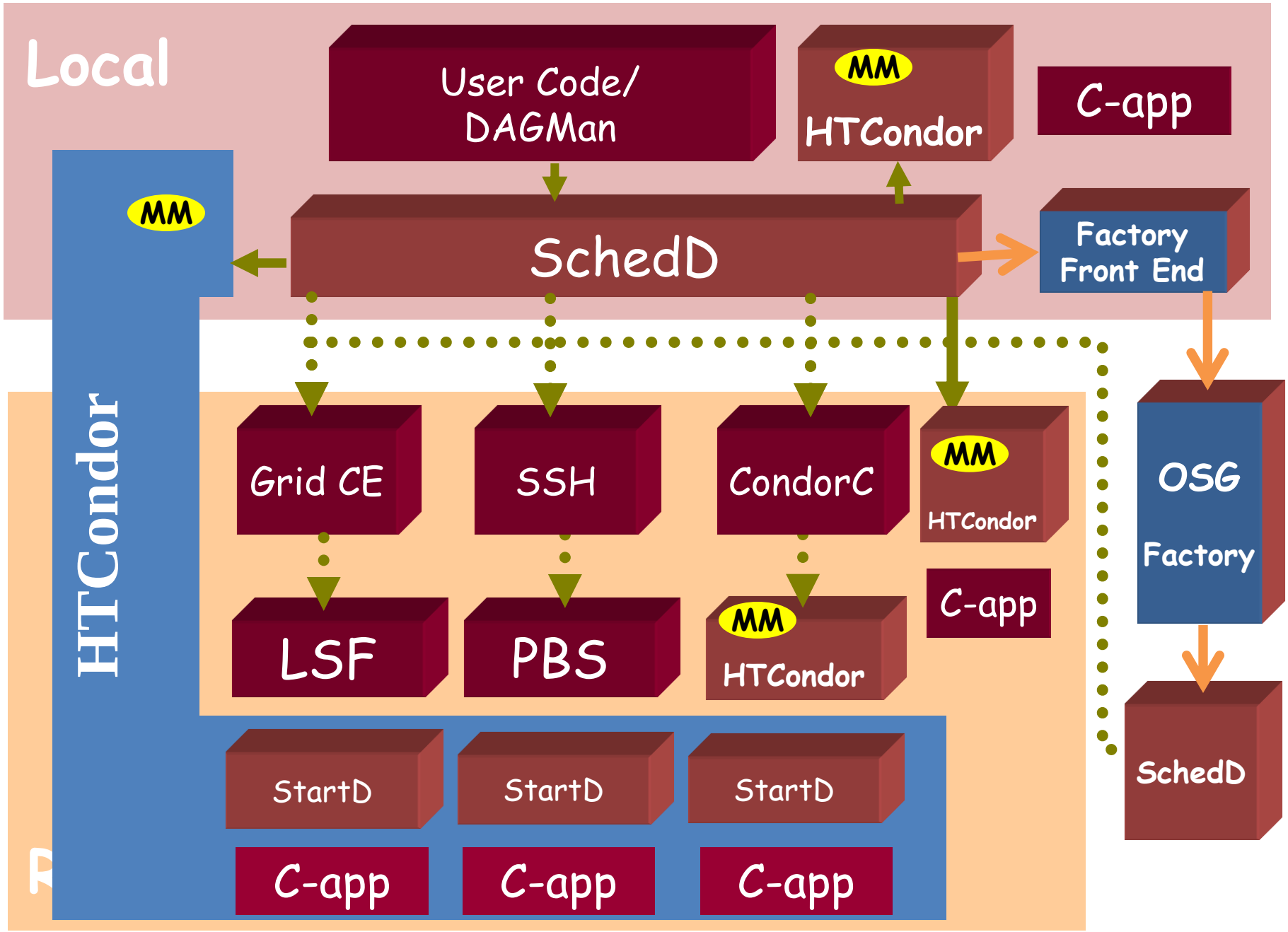
Work Delegation

A limited assignment of the responsibility to perform the work

- Delegation involved a definition of these "responsibilities"
- Responsibilities my be further delegated
- Delegation consumes resources
- Delegation is a "lease"
- Tree of delegations

HTCondor uses a two phase matchmaking process to first allocate a collection of resources to a requestor and then to select a task to be delegated for execution within the constraints of these resources


MatchMaker
Match!
Wi
I am
am
for
res
MM
W3
Claim Resource
Delegate Work
I am D and
I am willing
to offer you
resources
SchedD
StartD


Local
User Code/
DAGMan
MM
HTCondor
C-app
MM
SchedD
Factory
Front End
HTCondor
Grid CE
SSH
CondorC
MM
HTCondor
OSG
Factory
C-app
LSF
PBS
MM
HTCondor
SchedD
StartD
StartD
StartD
C-app
C-app
C-app

The OSG GildeIn factory uses a HTCondor SchedD as a resource provisioning agent on behalf of the (local) SchedD. The factory decides when, from where and for how long to keep an acquired resource.

The paradigm shift of 00's – computing capacity sold on demand for short time periods.

Novartis Uses AWS to Conduct 39 Years of Computational Chemistry In 9 Hours

Amazon Web Services

Subscribe 30,460

969 views

Add to Share More 10 0

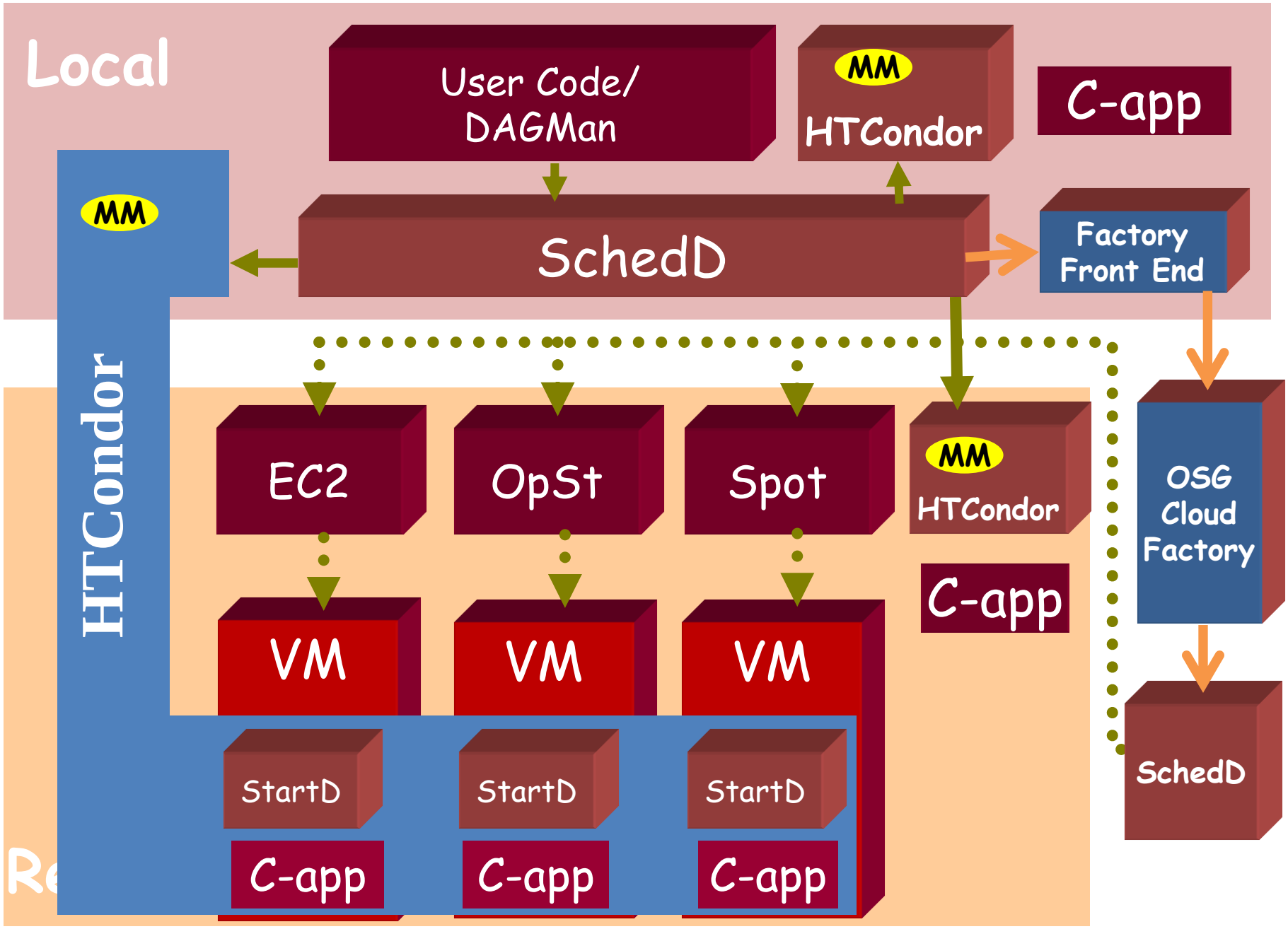
Published on Jul 17, 2014

Learn how Novartis compressed 39 years of computational chemistry into 9 hours with AWS, screening 10 million compounds and identifying three promising candidates for about $4,000. Novartis Institutes for Biomedical Research's (NIBR) purpose is to cure, care, and provide medicines that treat and prevent

http://bit.ly/1oe4m0c

More than 10K servers (80K cores) provided by CycleComputing and all managed by HTCondor

Are EC2 Spot instances a Grid, a Cloud or just a Distributed HTC System where resources come and go at (local) will?


Local
User Code/
DAGMan
MM
HTCondor
C-app
MM
SchedD
Factory
Front End
HTCondor
EC2
OpSt
Spot
MM
HTCondor
C-app
OSG
Cloud
Factory
VM
VM
VM
StartD
StartD
StartD
C-app
C-app
C-app
SchedD

Using HTCondor and the OSG GlideIn Factory, ONE local submit machine may be managing 100K jobs on 10K remote machines

Timeline of projects