

(HT)Condor - Past and Future

Miron Livny

John P. Morgridge Professor of Computer Science

Wisconsin Institutes for Discovery

University of Wisconsin-Madison



3/100

CoBe '98

חי

has the value of
18

ערך	אות
1	א
2	ב
3	ג
4	ד
5	ה
6	ו
7	ז
8	ח
9	ט
10	י
20	כ
30	ל
40	מ

חי

means
alive

- Europe HTCondor Week #3
- (HT)Condor Week #18
- 12 years to the Open Science Grid (OSG)
- 21 years to High Throughput Computing (HTC)
- 32 years to the first production deployment of (HT)Condor
- 40 years since I started to work on distributed systems

Global Scientific Computing via a Flock of Condors

CERN 92

Miron Livny

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

MISSION

Give scientists effective and efficient access to large amounts of cheap (if possible free) CPU cycles and main memory storage

THE CHALLENGE

How to turn existing privately owned clusters of *workstations, farms, multiprocessors, and supercomputers* into an efficient and effective Global Computing Environment?

In other words, how to minimize wait while idle?

APPROACH

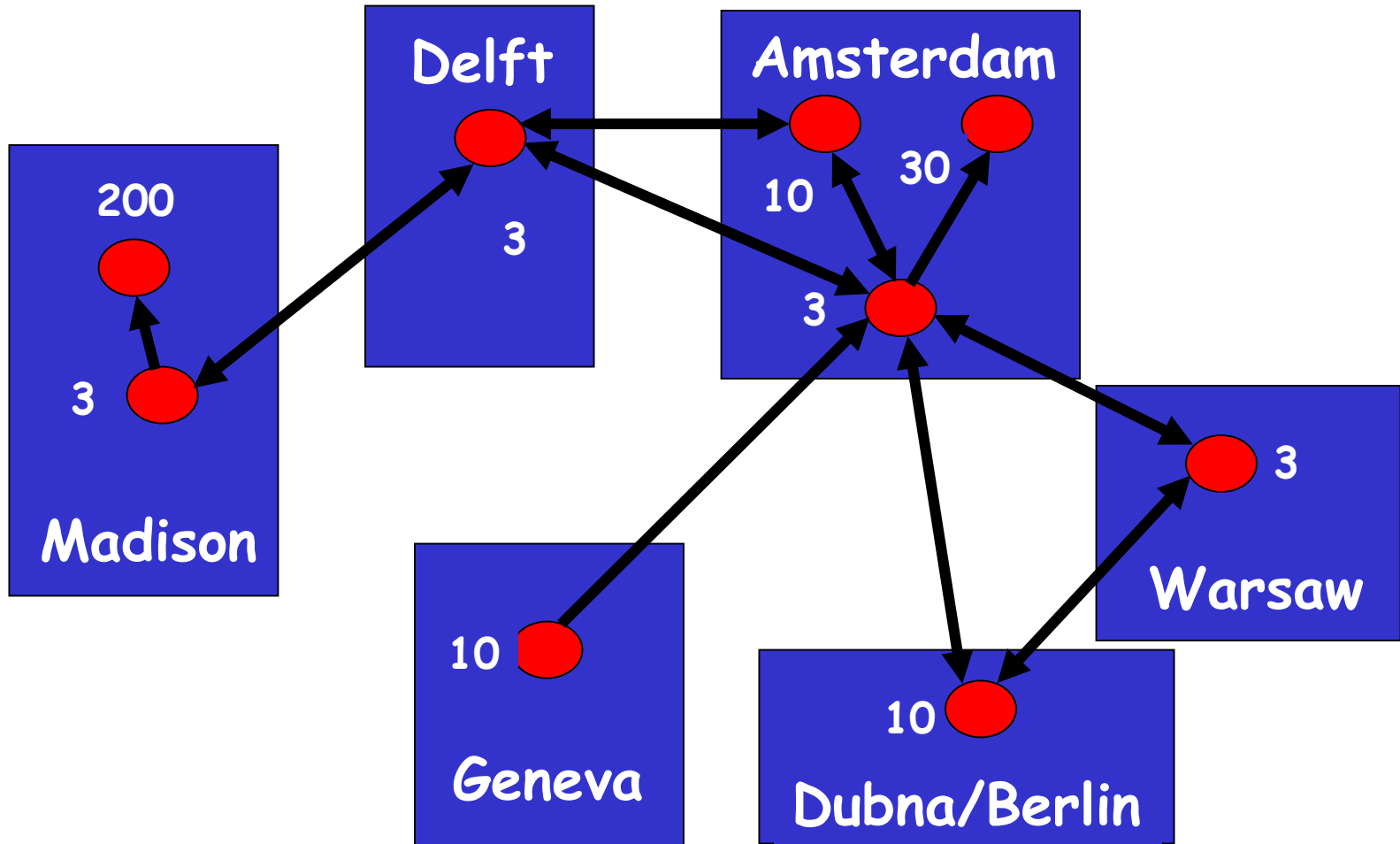
Use wide-area networks to transfer batch jobs between Condor systems

- Boundaries of each Condor system will be determined by physical or administrative considerations

TWO EFFORTS

- UW CAMPUS**
Condor systems at Engineering, Statistics, and Computer Sciences
- INTERNATIONAL**
We have started a collaboration between CERN-SMC-NIKHEF-Univ. of Amsterdam, and University of Wisconsin-Madison

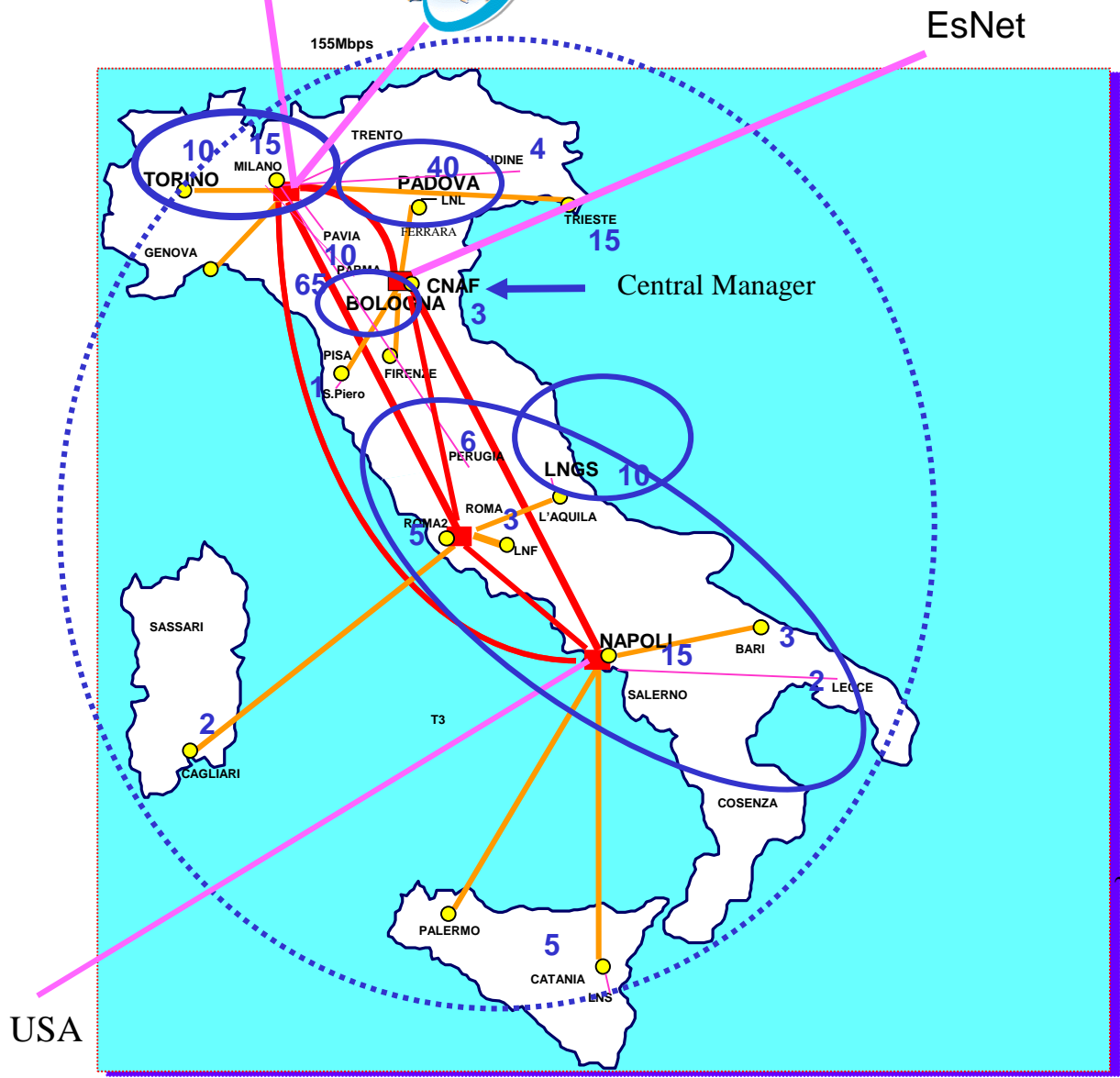
1994 Worldwide Flock of Condors



D. H. J Epema, Miron Livny, R. van Dantzig, X. Evers, and Jim Pruyne, "A Worldwide Flock of Condors : Load Sharing among Workstation Clusters" *Journal on Future Generations of Computer Systems*, Volume 12, 1996



INFN Condor Pool on WAN: checkpoint domain



GARR-B Topology

155 Mbps ATM based Network

● access points (PoP)

■ main transport nodes

○ CKPT domain # hosts

○ Default CKPT domain @ Cnaf

~180 machines ⇒ 500-1000 machines

6 ckpt servers ⇒ 25 ckpt servers

“The members of **OSG** are united by a commitment to promote the adoption and to advance the state of the art of *distributed high throughput computing (DHTC)* – *shared utilization of autonomous* resources where all the elements are optimized for maximizing computational throughput.”



1.42B core hours in 12 months!

Almost all jobs executed by the **OSG** leverage (HT)Condor technologies:

- Condor-G
- HTCondor-CE
- Basco
- Condor Collectors
- HTCondor overlays
- HTCondor pools

In the last 24 Hours

272,000 Jobs

3,289,000 CPU Hours

5,515,000 Transfers

512 TB Transfers

In the last 30 Days

9,973,000 Jobs

122,533,000 CPU Hours

205,077,000 Transfers

15,098 TB Transfers

In the last 12 Months

134,076,000 Jobs

1,433,825,000 CPU Hours

1,910,892,000 Transfers

165,000 TB Transfers

OSG delivered across 132 sites

Two weeks ago ...

Hello HTCondor Admin:

Intel has created a reference architecture that includes HTCondor for a Genomics Analysis computing platform that we'd like to promote on a publically-facing website.

Two questions:

1.) Do we have permission to display your Logo on the page with other open-source s/w packages? Example:

a. The links will point to your site & GitHub repo.

2.) Could you please send a high resolution version of your Logo,



SEE THE LATEST HEALTH & LIFE SCIENCES BLOG DISCUSSIONS

[Visit the blog](#)

Resources for Developers

Access open-source frameworks and tools for optimized genomics analytics, as well as code and reference architectures for implementation, benchmarking, and customization.



Genome Analysis Toolkit*

The industry leading analysis solution. Now bundled with key supporting components for easier implementation.

[Learn more >](#)
[Get GATK* >](#)



Cromwell

Broad's Workflow Description Language (WDL) - based execution management engine. Flexible, open source, and applicable for local or cloud operation.

[Learn more >](#)
[Get Cromwell >](#)



Genomics Kernel Library*

Optimized compute kernels for Intel® architecture under 64-bit Linux* & Mac OSX*. Includes Kernels for PairHMM, igzip compression, and otc_zlib compression.

[Learn more and get GKL >](#)



GenomicsDB

A specialized & highly optimized sparse data array for genomic variant data management.

[Learn more >](#)
[Get GenomicsDB >](#)



Lustre

Parallel file system for HPC big data workloads. Includes Intel® Manager for Lustre* software.



HTCondor

Workload management, job queueing, scheduling, prioritizing, resource monitoring, and resource management for HPC.



Intel® Omni-Path Architecture

High-performance data center communication architecture for low latency, low power consumption, and high throughput.



RSD Reference Architecture

Reference architecture details and performance benchmarking details in a best practices deployment white paper.

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

Ecclesiastes Chapter 1 verse 9

DISTRIBUTED COMPUTING BASICS FOR BIG DATA



RELATED BOOK

**Big Data For
Dummies**

By [Judith Hurwitz](#), [Alan Nugent](#), [Fern Halper](#), [Marcia Kaufman](#)

If your company is considering a big data project, it's important that you understand some distributed computing basics first. There isn't a single distributed computing model because computing resources can be distributed in many ways.

Claims for “benefits” provided by Distributed Processing Systems

- 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

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

Load Balancing
in
Homogeneous Broadcast Distributed Systems

by
Miron Livny and Myron Melman
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot, Israel

ABSTRACT

Three different load balancing algorithms for distributed systems that consist of a number of identical processors and a CSMA communication system are presented in this paper. Some of the properties of a multi-resource system process are demonstrated by Simulation is used as a means of measuring the interdependency between the performance of the distributed system and the behavior of the load balancing algorithm. The results of this study show the characteristics of the load

The assignment algorithm is motivated by the desire to achieve better overall performance relative to some selected metric of system performance. The strategy of the load balancing algorithm has a strong effect on the utilization of the system

**COMPUTER
NETWORK
PERFORMANCE
SYMPOSIUM**



Sponsored By SIGCOMM, SIGMETRICS & SIGOPS And The University Of Maryland

College Park, Maryland
April 13-14, 1982

Minimize **wait**
(job/task queued)
while Idle (a
resource that is
capable and willing to
serve the job/task is
running a lower
priority job/task)

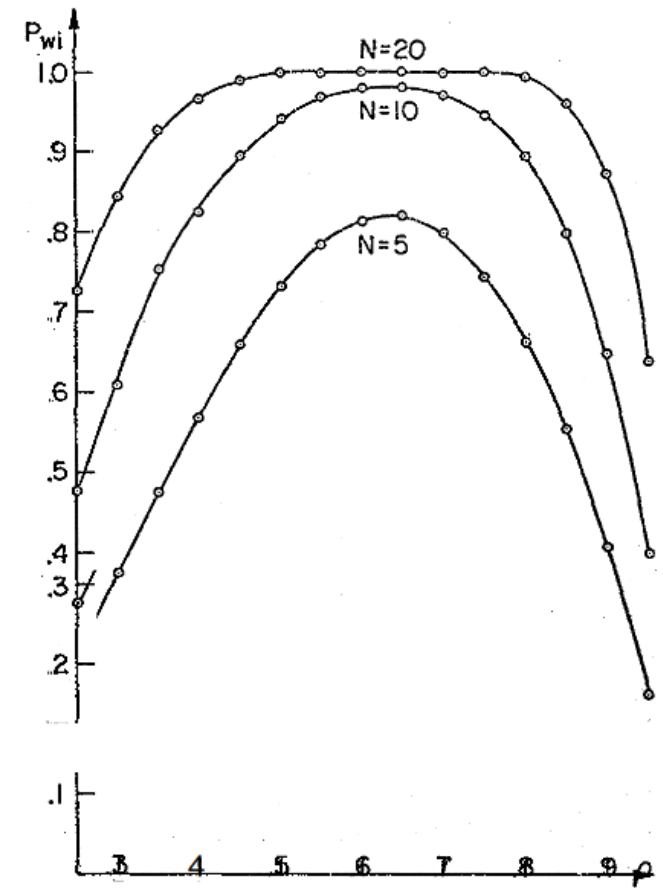


figure: 1 P_{wi} as a function of ρ

Submit locally (queue and manage your jobs/tasks locally; leverage your local resources) **and** **run globally** (acquire any resource that is capable and willing to run your job/task)

- **Job owner identity is local**
 - Owner identity should never “travel” with the job to execution site
 - Owner attributes are local
- **Name spaces are local**
 - File names are locally defined
- **Resource acquisition is local**
 - Submission site (local) is responsible for the acquisition of all resources

“external” forces moved us away from this “pure” local centric view of the distributed computing environment.

With the help of capabilities (short lived tokens) and reassignment of responsibilities we are committed to regain local control.

Handing users with money (real or funny) to acquire commutating resources helps us move (push) in this positive direction.

In **1996** I introduced the distinction between High **Performance** Computing (**HPC**) and High **Throughput** Computing (**HTC**) in a seminar at NASA Goddard Flight Center and a month later at 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
by Alan Beck, editor in chief

06.27.97
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

Future Directions for NSF Advanced Computing Infrastructure to Support U.S. Science and Engineering in 2017-2020

AUTHORS

Committee on Future Directions for NSF Advanced Computing Infrastructure to Support U.S. Science in 2017-2020; Computer Science and Telecommunications Board; Division on Engineering and Physical Sciences; National Academies of Sciences, Engineering, and Medicine

“... many fields today rely on high-throughput computing for discovery.”

“Many fields increasingly rely on high-throughput computing”

High Throughput Computing
requires **automation** as it
is a **24-7-365** activity that
involves large numbers of jobs

$FLOPY \neq (60*60*24*7*52)*FLOPS$

$100K \text{ Hours} * 1 \text{ Job} \neq 1 \text{ H} * 100K \text{ J}$

**HTC is about many jobs,
many users, many
servers, many sites and
(potentially) long running
workflows**

HTCondor uses a matchmaking process to dynamically **acquire** resources.

HTCondor uses a matchmaking process to **provision** them to queued jobs.

HTCondor launches jobs via a task delegation protocol.

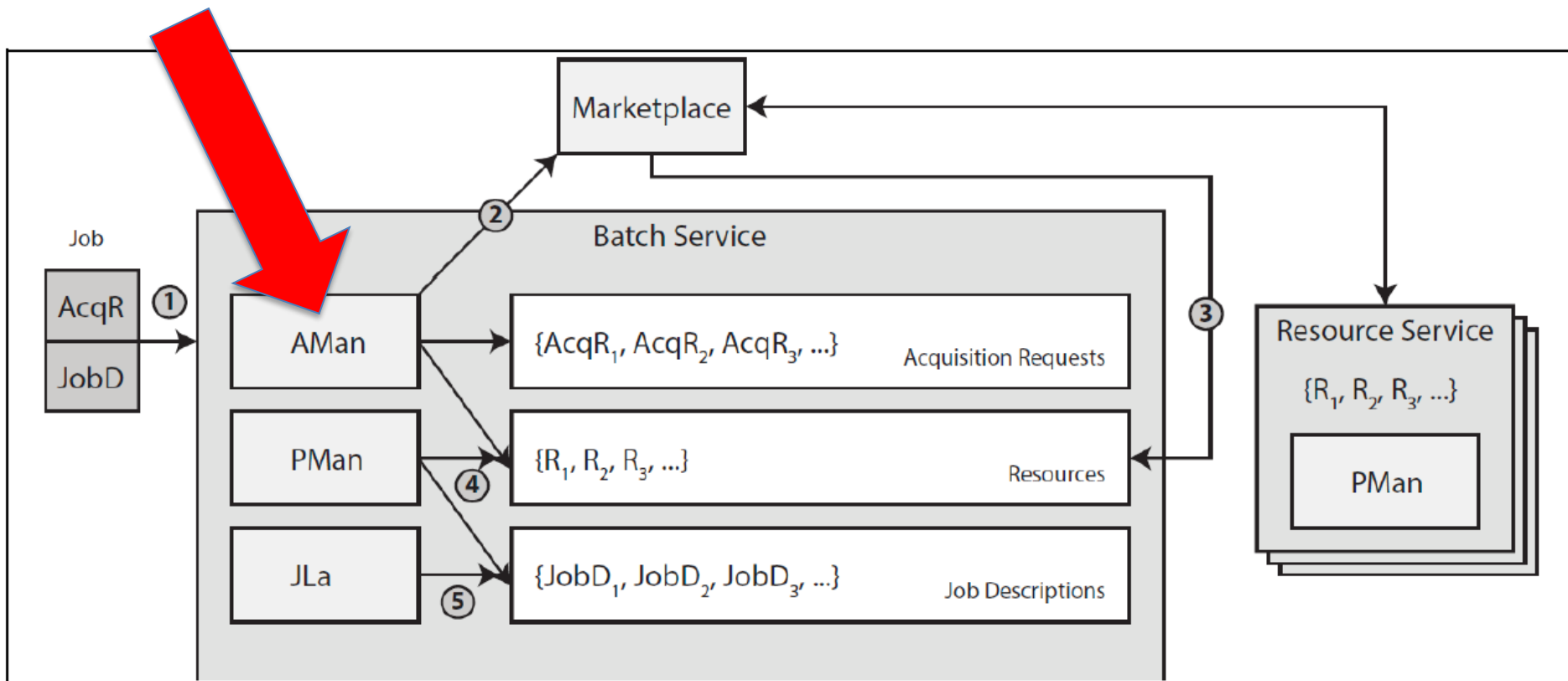


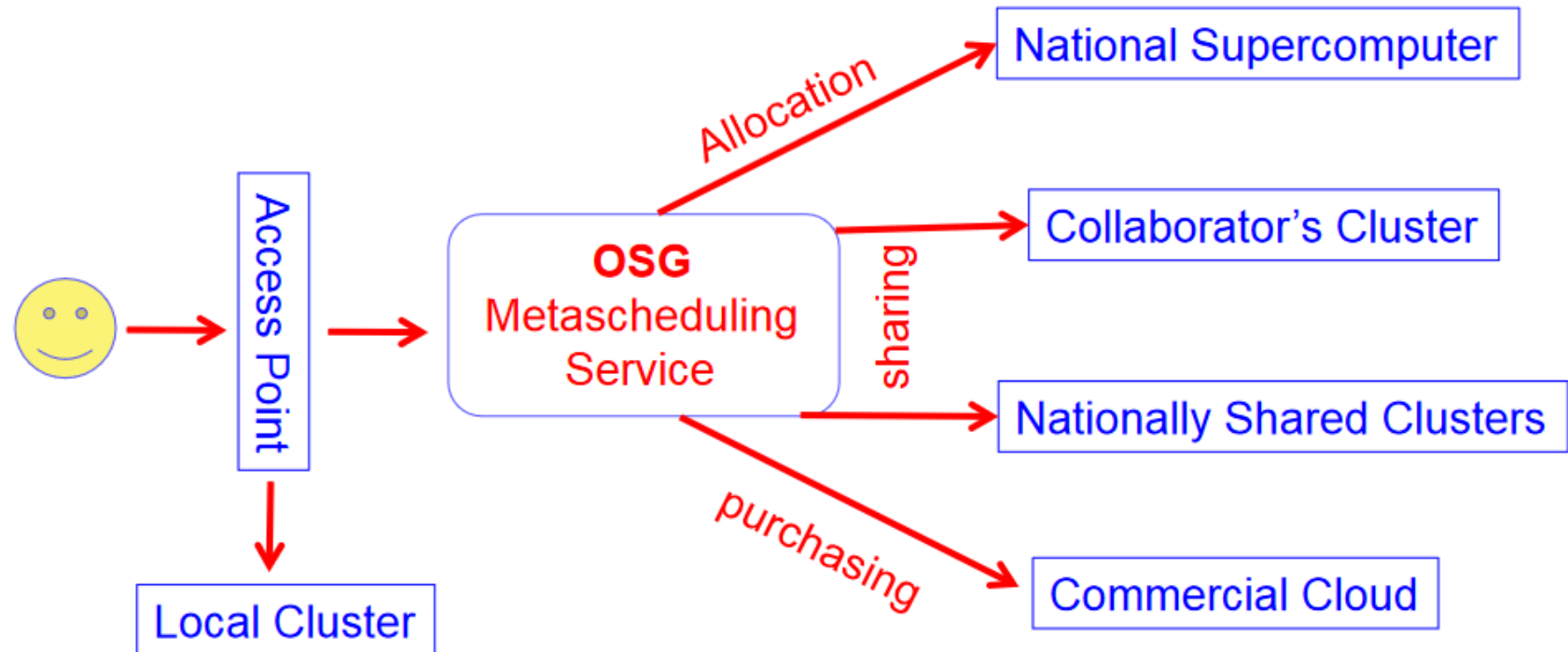
Figure 1: The HTC Framework: (1) A user submits job, composed of an AcqR and a JobD. (2) The AMan requests a resource compatible with an AcqR. (3) The MarketPlace offers a compatible Resource to the batch service. (4) The Pman selects a job description and Resource to send to (5) the JLa, which runs the job.

HTCondor 101

- Jobs are submitted to the HTCondor **SchedD**
- A job can be a **Container** or a **VM**
- The **SchedD** can **Flock** to additional **Matchmakers**
- The **SchedD** can delegate a job for execution to a **HTCondor StartD**
- The **SchedD** can delegate a job for execution to a another **Batch system**.
- The **SchedD** can delegate a job for execution to a **Grid Compute Element (CE)**
- The **SchedD** can delegate a job for execution to a **Commercial Cloud**

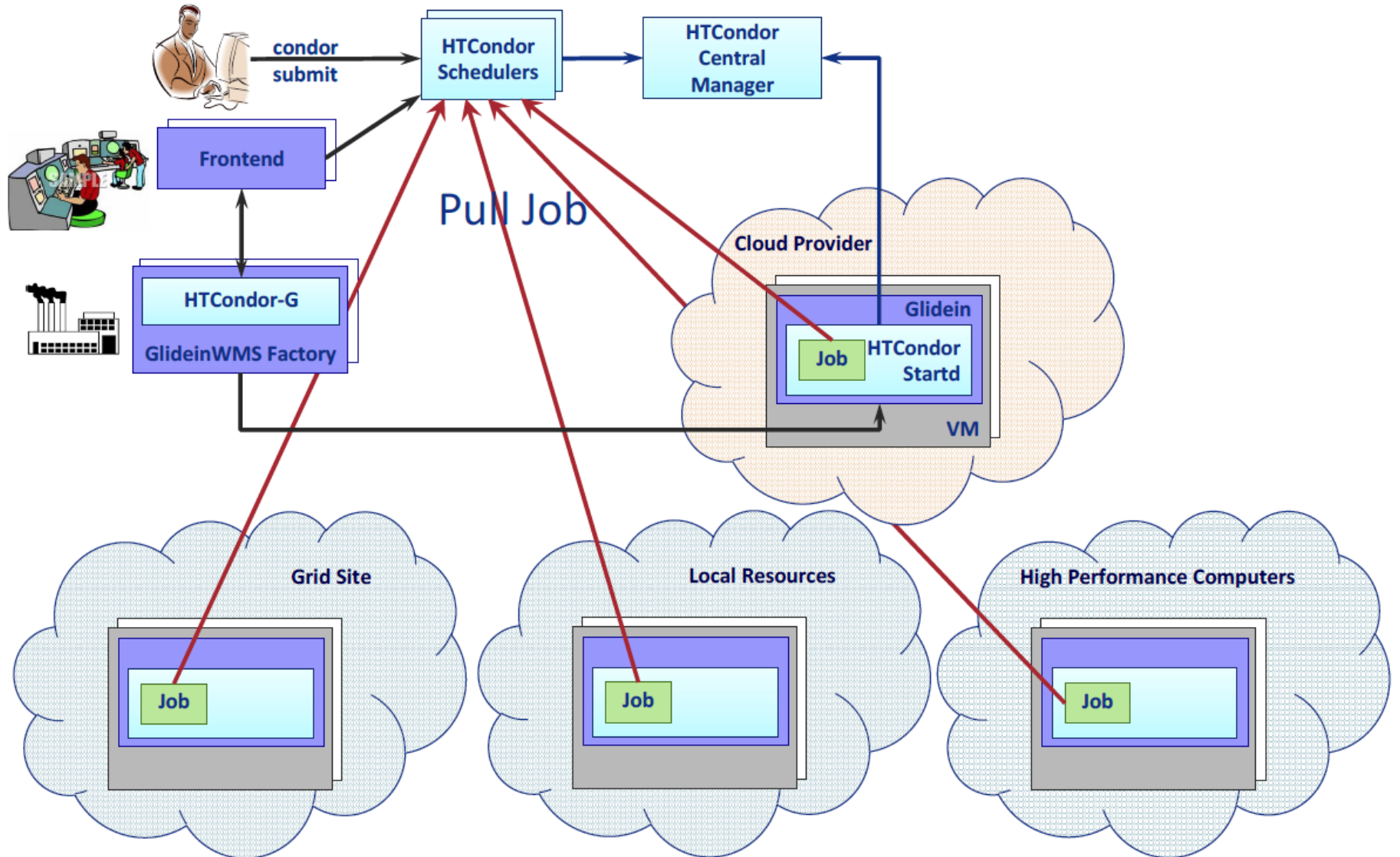


The OSG Vision



OSG integrates computing across different resource types and business models to allow Campus IT to offer a maximally flexible HTC environment to your researchers.

HEPCloud – glideinWMS and HTCondor



SC16 CMS Demonstrator

Target: generate 1 Billion events in
48 hours during Supercomputing 2016 on
Google Cloud via HEPCloud

35% filter efficiency = stage out 380
million events → 150 TB output

Double the size of global CMS computing
resources

CMS Higgs Event - credit: CERN
https://commons.wikimedia.org/wiki/File:CMS_Higgs-event.jpg

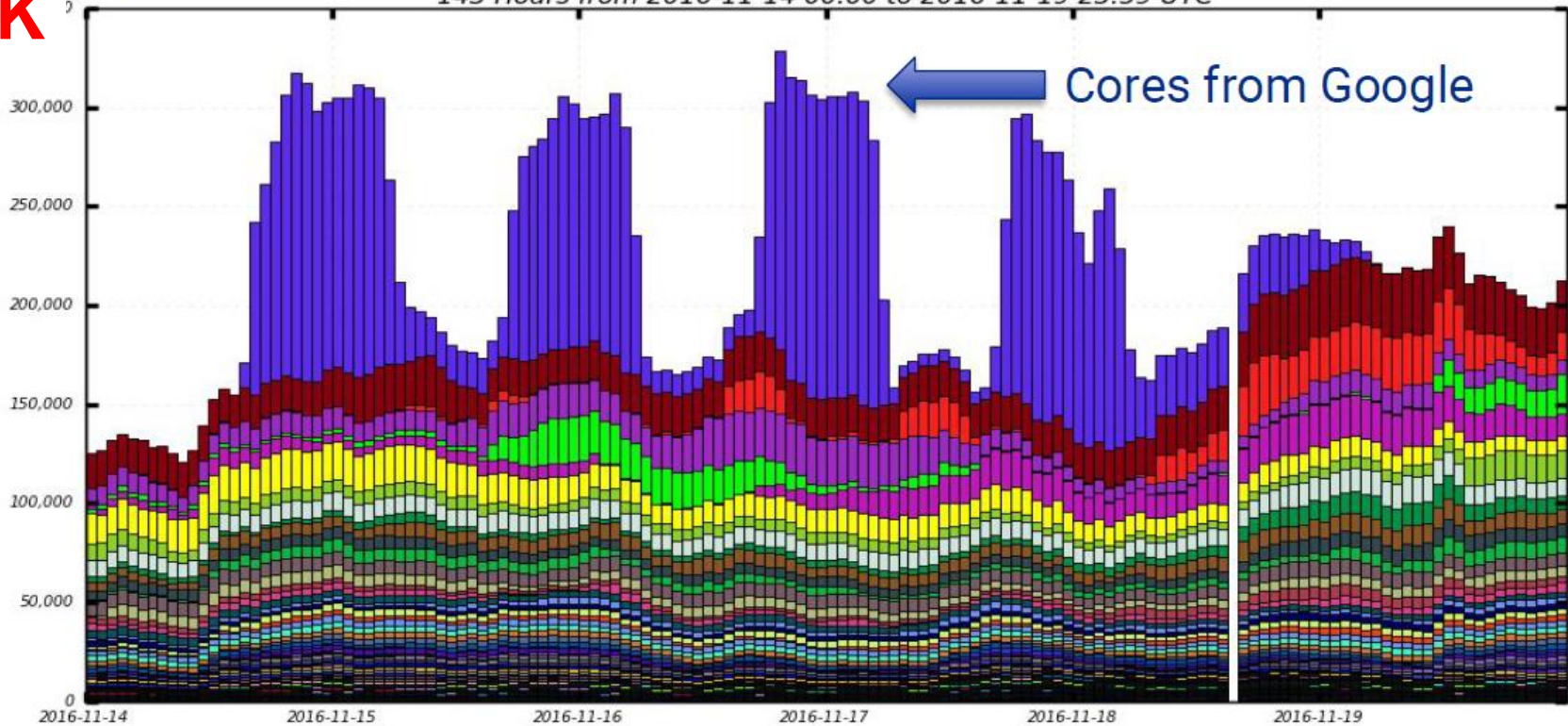


All Managed by HTCondor!

350K

Dashboard

Running Job Cores
143 Hours from 2016-11-14 00:00 to 2016-11-19 23:59 UTC



T3_US_HEP_Cloud
T3_US_NotreDame
T2_US_Nebraska

T1_US_FNAL
T2_CH_CERN
T2_US_Caltech

T0_CH_CERN
T2_DE_DESY
T2_US_Purdue

T2_US_Wisconsin
T2_US_Florida
T2_US_MIT

T2_CH_CERN_HLT
T1_IT_CNAF
T2_US_UCSD



CHTC Home

About

Our Approach

Our Customers

Our Staff

Our Open Positions

How To's

Get Started

Get Help

All User Guides

Use Our HTC Submit Node

Run Your First HTC Jobs

HTC for MatLab, Python or R

HT CENTER FOR HIGH THROUGHPUT COMPUTING

Office Hours!
Tues/Thurs, 3-4:30pm
Wed, 9:30-11:30am
[Click for details](#)



The Center for High Throughput Computing (CHTC) supports a variety of scalable computing resources and services for UW-affiliated researchers and their collaborators, including high-throughput computing (HTC) *and*, tightly-couple computations (e.g. message passing interface, or "MPI"), high-memory, and GPUs. CHTC compute systems and personnel are funded by the National Science Foundation, the National Institutes of Health, the Department of Energy (DOE), the National Science Foundation (NSF), the Morgridge Institute for

CHTC Quick Facts	Jul'13- Jun'14	Jul'14- Jun'15	Jul'15- Jun'16	Jul'15- Jun'16
Million Hours Served	121	265	325	5
Research Projects	126	188	205	5

**A crowded and growing space of
distributed execution
environments that can benefit
from the capabilities offered by
HTCondor**



ELSEVIER

Future Generation Computer Systems

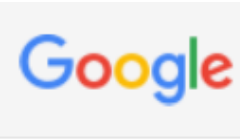
Volume 12, Issue 1, May 1996, Pages 67-85



Paper

Interfacing Condor and PVM to harness the cycles of workstation clusters

Jim Pruyne✉, Miron Livny



Scholar

About 97 results (0.02 sec)

All citations

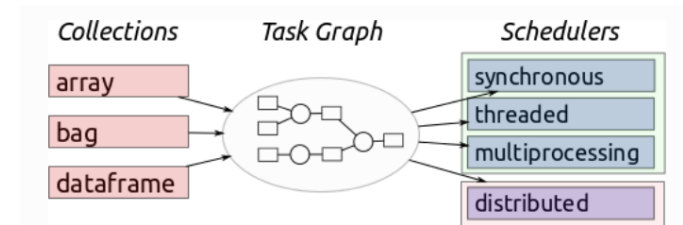
[Interfacing Condor and PVM to harness the cycles of workstation clusters](#)

Search within citation articles



The road from PVM to DASK

- In the 80's we added support for resource management primitives (**AddHost**, **DelHost** and **Spawn**) of Parallel Virtual Machine (PVM) message passing environment
- Today we working on adding support for dynamic (on the fly) addition and removal of worker nodes to **Dask**



2 CARMI

The Condor Application Resource Management Interface (CARMI) provides services for writing parallel applications in an environment with dynamic resources. CARMI, therefore, allows an application to exploit new resources which become available at runtime, and aids an application in detecting and managing resource loss. These services can be used in a dedicated environment to utilize resources which are freed by other applications, or to allow a scheduling mechanism to revoke resources from a running application. In an opportunistic environment, CARMI permits an application to grow to new resources as they become available, and cope with resources being reclaimed by their owner.

J. Pruyne and M. Livny, "Parallel Processing on Dynamic Resources with CARMI," in *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph (eds.), Springer-Verlag, 1995.

How to expose all shared resources that require protection (allocation) to application (user) software and control (configuration)?

- 1. Request/Acquire**
- 2. Schedule/Provision**
- 3. Allocate**
- 4. Protect**
- 5. Monitor**
- 6. Reclaim**
- 7. Account**

Miron Livny

*John P. Morgridge Professor of Computer Science
Director Center for High Throughput Computing
University of Wisconsin Madison*

It is all about Storage Management, stupid!

(Allocate **B** bytes for **T** time units)

- We do not have tools to manage storage allocations
- We do not have tools to schedule storage allocations
- We do not have protocols to request allocation of storage
- We do not have means to deal with “sorry no storage available now”
- We do not know how to manage, use and reclaim opportunistic storage
- We do not know how to budget for storage space
- We do not know how to schedule access to storage devices