From Principles to Capabilities -
the Birth and Evolution of
High Throughput Computing

Miron Livny
Wisconsin Institutes for Discovery
Madison-Wisconsin
The lessons of the past and the illusions of predictions
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 Chapter 1 verse 9

The Talmud says in the name of Rabbi Yochanan, "Since the destruction of the Temple, prophecy has been taken from prophets and given to fools and children."

(Baba Batra 12b)
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 and a month later at the European Laboratory for Particle Physics (CERN). In June of 1997 HPCWire published an interview on High Throughput Computing.

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...
Why HTC?

For many experimental scientists, scientific progress and quality of research are strongly linked to computing **throughput**. In other words, they are less concerned about **instantaneous** computing power. Instead, what matters to them is the amount of computing they can harness over a day, a month or a year --- they measure computing power in units of scenarios per **day**, wind patterns per **week**, instructions sets per **month**, or crystal configurations per **year**.
High Throughput Computing is a 24-7-365 activity

\[ \text{FLOPY} \neq (60 \times 60 \times 24 \times 7 \times 52) \times \text{FLOPS} \]
“The members of the 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.”
Scientific Computing for the 21st Century

Workshop on HPC and Super-computing for Future Science Applications

June 6, 2013

Richard Carlson

Richard.Carlson@science.doe.gov
Traditional Scientific Computing Issues

• Tussle between High Performance Computing and High Throughput Computing
  – Capability vs Capacity
• Tussle between Grid / Cloud / Distributed computing
  – What are the differences between grid and cloud
• Tussle between hardware ownership and software services
  – Who owns and manages the hardware vs the deployed services
• Tussle between basic research and sustained deployment activities
  – How to balance research with sustainability
In 1978 I fell in love with the problem of load balancing in distributed systems
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


- 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
BASICS OF A M/M/1 SYSTEM

Expected # of customers is \( \frac{1}{1-\rho} \), where \( \rho = \frac{\lambda}{\mu} \) is the utilization.

When utilization is 80%, you wait on the average 4 units for every unit of service.
When utilization is 80%, you wait on the average 4 units for every unit of service.

When utilization is 80%, 25% of the time a customer is waiting for service while a server is idle.
Wait while Idle (WwI) in m*M/M/1
In 1983 I wrote a Ph.D. thesis –

“Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems”

Should I stay or should I move?
“... 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’. ... “

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}
Submit Locally and run Globally

(Here is the work and here are the resources I bring to the table)
Global Scientific Computing via a Flock of Condors

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

Miron Livny

Computer Sciences Department
University of Wisconsin — Madison
Madison, Wisconsin
{miron@cs.wisc.edu}
1994 Worldwide Flock of Condors

Madison

Delft

Amsterdam

Warsaw

Geneva

Dubna/Berlin
HTC on the UW campus

- 710 million hours
- 90 million hours
- .03 million hours

Desktop
UW-Madison CHTC
Open Science Grid
Subject: Meeting request
From: Michael Gofman <michael.gofman@gmail.com>
Date: Thu, 16 May 2013 11:47:50 -0500
To: MIRON LIVNY <MIRON@cs.wisc.edu>

Dear Miron,

I am an assistant professor of finance at UW-Madison. I did my Phd at the University of Chicago and master degrees at the Tel Aviv University.

In the last couple months I was using HTC resources that you developed to compute optimal financial architecture.

I would like to meet with you and tell you more about my project as well to thank you personally for developing this amazing platform.

Yours,
Michael
Experimental Computer Science where you and other scientists are the
2006 SIGMOD Test of Time Award

BIRCH: An Efficient Data Clustering Method for Very Large Databases

Tian Zhang (University of Wisconsin, Madison), Raghu Ramakrishnan (University of Wisconsin, Madison), and Miron Livny (University of Wisconsin, Madison)

The paper introduces a novel, scalable, simple yet effective technique for clustering large multi-dimensional datasets, based on core database management system technology (indexing). It has had significant research impact and has influenced commercial products.
Dear Professor Livny,

I'm writing to you as I wish to invite you to a panel we're organizing at the next ECCS 2012 on "Experiments in Computer Science: Are Traditional Experimental Principles Enough?"

I was present during your ECSS presentation last year in Milan on "Experimental Computer Science and Computing Infrastructures" and, actually, was the person who asked you about a more scientifically oriented notion of experiment.

I must confess that your talk, and the discussion I had with some colleagues after, was ones of the driving forces behind the organization of this panel and a pre-summit workshop (also on experiments in computer science So it would be really fantastic if you would be interested in participating in the panel.
Edsger Dijkstra once stated:

"Computer science is no more about computers than astronomy is about telescopes."

Research Methods for Science By Michael P. Marder page 14. Published by Cambridge University Press
Abstract. We examine the philosophical disputes among computer scientists concerning methodological, ontological, and epistemological questions: Is computer science a branch of mathematics, an engineering discipline, or a natural science? Should knowledge about the behavior of programs proceed deductively or empirically? Are computer programs on a par with mathematical objects, with mere data, or with mental processes? We conclude that distinct positions taken in regard to these questions emanate from distinct sets of received beliefs or paradigms within the discipline:

Real and hard Computer Science problems are exposed when you do it for “real”
MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

Abstract

MapReduce is a programming model that provides a platform for processing and analyzing large data sets. Users specify a map function to perform a mapping from tuples of key-value pairs to groups of intermediate key-value pairs, and a reduce function to merge the intermediate key-value pairs that map to the same intermediate key. The run-time system then partitions the input data, schedules execution across a set of machines, and manages the required inter-machine communication. This allows programmers without experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

The MapReduce implementation relies on an in-house cluster management system that is responsible for distributing and running user tasks on a large collection of shared machines. Though not the focus of this paper, the cluster management system is similar in spirit to other systems such as Condor [16].

BAD-FS [5] has a very different programming model from MapReduce, and unlike MapReduce, is targeted to the execution of jobs across a wide-area network. However, there are two fundamental similarities. (1) Both systems use redundant execution to recover from data loss caused by failures. (2) Both use locality-aware scheduling to reduce the amount of data sent across congested network links.

To appear in OSDI 2004
You have Impact!
“Why are you leaving academia and taking a job in industry?”

“I want to have impact!”
In the words of Mike Carey

“I left academia for industry because I was drawn to the idea of getting more direct access to real problems - from customers and challenges encountered while building commercial-grade software - because I felt like I was in somewhat of a mode of inventing and solving problems, at least w.r.t. some of the things I'd been working on. Sure, that was leading to many written/submitted/accepted papers, but it was somehow less than satisfying after awhile.”
Solving “real-life” end-to-end problems makes you hype resistance
Gartner Hype Cycle

- Peak of Inflated Expectations
- Plateau of Productivity
- Slope of Enlightenment
- Trough of Disillusionment
- Technology Trigger
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!
How do we prepare for the HTC needs of 2020?
Scientific Collaborations at Extreme-Scales:

dV/dt - Accelerating the Rate of Progress towards Extreme Scale Collaborative Science

Collaboration of five institutions – ANL, ISI, UCSD, UND and UW Funded by the Advanced Scientific Computing Research (ASCR) program of the DOE Office of Science
“Using planning as the unifying concept for this project, we will develop and evaluate by means of at-scale experimentation novel algorithms and software architectures that will make it less labor intensive for a scientist to find the appropriate computing resources, acquire those resources, deploy the desired applications and data on these resources, and then manage them as the applications run. The proposed research will advance the understanding of resource management within a collaboration in the areas of: trust, planning for resource provisioning, and workload, computer, data, and network resource management.”
“Over the last 15 years, Condor has evolved from a concept to an essential component of U.S. and international cyberinfrastructure supporting a wide range of research, education, and outreach communities. The Condor team is among the top two or three cyberinfrastructure development teams in the country. In spite of their success, this proposal shows them to be committed to rapid development of new capabilities to assure that Condor remains a competitive offering. Within the NSF portfolio of computational and data-intensive cyberinfrastructure offerings, the High Throughput Computing Condor software system ranks with the NSF High Performance Computing centers in importance for supporting NSF researchers.”

A recent anonymous NSF review
“... a mix of continuous changes in technologies, user and application requirements, and the business model of computing capacity acquisition will continue to pose new challenges and opportunities to the effectiveness of scientific HTC. ... we have identified six key challenge areas that we believe will drive HTC technologies innovation in the next five years.”

- Evolving resource acquisition models
- Hardware complexity
- Widely disparate use cases
- Data intensive computing
- Black-box applications
- Scalability
Thank you HPDC!