

# Parallel Computing: Multicore, Clusters and the Cloud

Tony Hey  
Vice President  
Microsoft Research



# Outline

**The Microsoft-Intel UPCRCs**

**Parallel Programming for Morts**

**Parallel Computing and the Cloud**

**Data-Intensive Science**

**The Future?**



# Microsoft-Intel Universal Parallel Computing Research Centers (UPCRCs)

- In 2008 Microsoft and Intel launched the **UPCRC** program to sponsor research into **multi-core** parallelism at the client
- Initial call to 25 CS Departments: two university centers selected
  - UPCRC at UC Berkeley
  - UPCRC at UIUC Illinois
- **Separate Microsoft Research initiative In Europe**
  - Joint Parallel Computing Institute at UPC Barcelona with Microsoft Research Cambridge
  - Initial focus on transactional memory

# UPCRCs Major Accomplishments

## UC Berkeley:

**12** faculty, **120** graduate students, **4** post-docs, **10** MSR interns, **300+** publications in journals, conferences, **6** Best Paper Awards, **20+** visits to/from MSFT, strong participation at the MSFT Academic Summits, **20+** lectures at MSFT, outreach activities, including courses and summer schools attended by **1100+**.

## UIUC:






**18** faculty, **45+** graduate students, **2** MSR interns, **20+** visits from/to MSFT, strong participation at the MSFT Faculty Summits, **42** UPCRC seminars, **60+** journal publications, **100+** conference presentations, outreach activities, including courses and summer schools with **750+** attendants.

# Structural Patterns for Parallel Composition

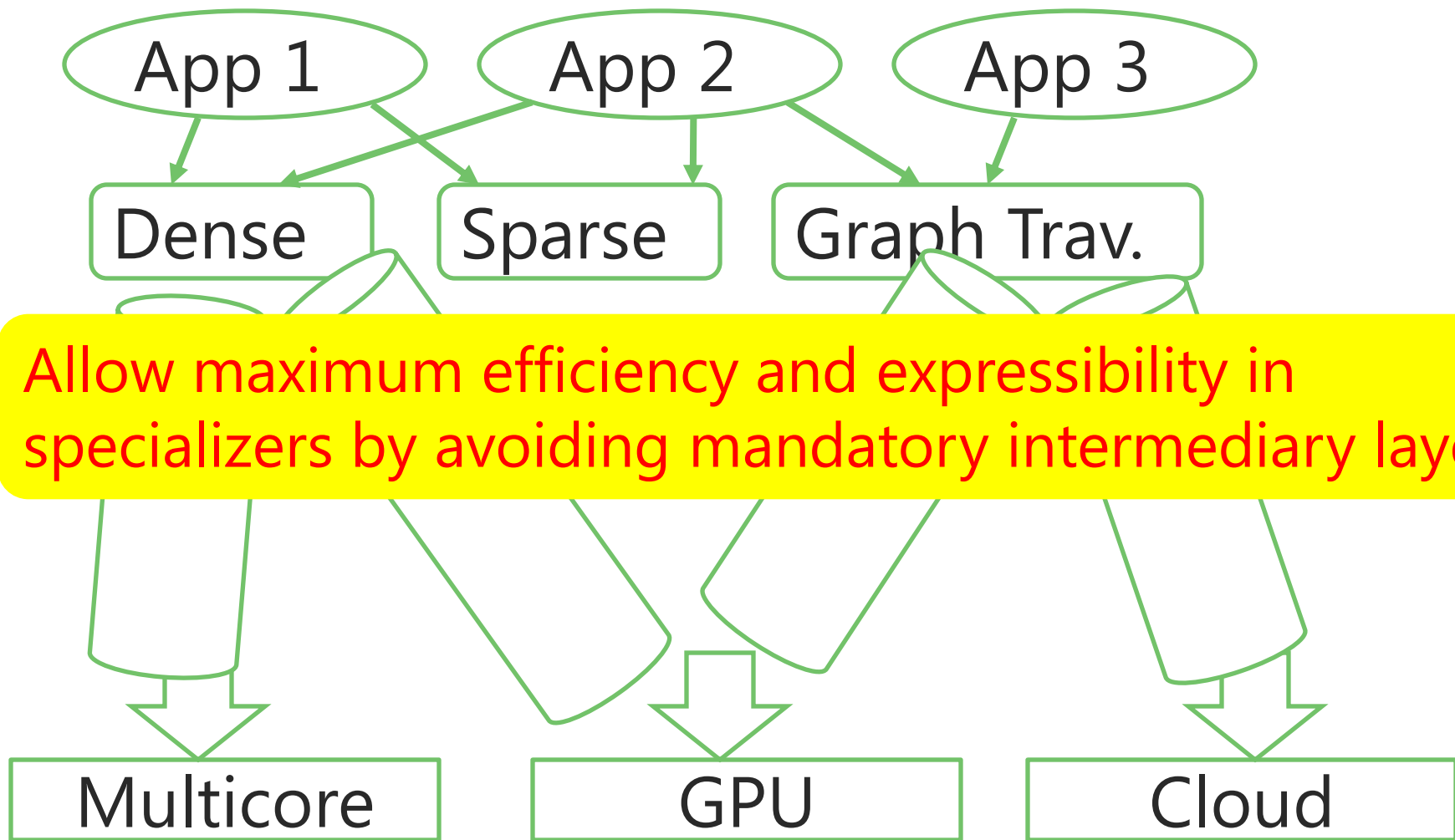
UPCRC Berkeley believes in relevance of computational and structural patterns at all levels of programming

- From Domain level through to Efficiency level
- Patterns provide a good vocabulary for domain experts
- Patterns are also comprehensible to efficiency-level experts or hardware architects
- Patterns act as a *lingua franca* between the different projects
- Expanded Phil Colella's original 'Seven Dwarfs' computational kernels to 12, 13, 14 ...

# 'Heat Map' for the Berkeley Dwarfs

	Embed	SPEC	DB	Games	ML	CAD	HPC	 Health	 Image	 Speech	 Music	 Browser	
1 Finite State Mach.	Red	Red	Red	Yellow	Yellow	Yellow	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Red
2 Circuits	Red	Light Blue	Green	Light Blue	Green	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Red
3 Graph Algorithms	Red	Yellow	Yellow	Yellow	Red	Red	Light Blue	Red	Light Blue	Red	Green	Green	Green
4 Structured Grid	Red	Red	Light Blue	Yellow	Light Blue	Light Blue	Red	Light Blue	Red	Light Blue	Light Blue	Light Blue	Light Blue
5 Dense Matrix	Red	Red	Yellow	Red	Red	Red	Red	Light Blue	Red	Red	Red	Red	Light Blue
6 Sparse Matrix	Yellow	Yellow	Light Blue	Red	Red	Red	Red	Red	Light Blue	Light Blue	Red	Light Blue	Light Blue
7 Spectral (FFT)	Yellow	Light Blue	Light Blue	Yellow	Yellow	Yellow	Red	Light Blue	Green	Red	Red	Red	Red
8 Dynamic Prog	Yellow	Light Blue	Red	Light Blue	Red	Red	Light Blue	Light Blue	Light Blue	Yellow	Light Blue	Light Blue	Red
9 Particle Methods	Light Blue	Yellow	Light Blue	Yellow	Light Blue	Light Blue	Red	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue
10 Backtrack/ B&B	Light Blue	Light Blue	Yellow	Light Blue	Red	Red	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Yellow	Light Blue
11 Graphical Models	Light Blue	Light Blue	Yellow	Light Blue	Red	Light Blue	Light Blue	Light Blue	Light Blue	Light Blue	Red	Red	Light Blue
12 Unstructured Grid	Light Blue	Light Blue	Light Blue	Yellow	Yellow	Yellow	Red	Red	Light Blue	Light Blue	Red	Light Blue	Light Blue

# Specializers: Pattern-specific and platform-specific compilers (SEJITS)



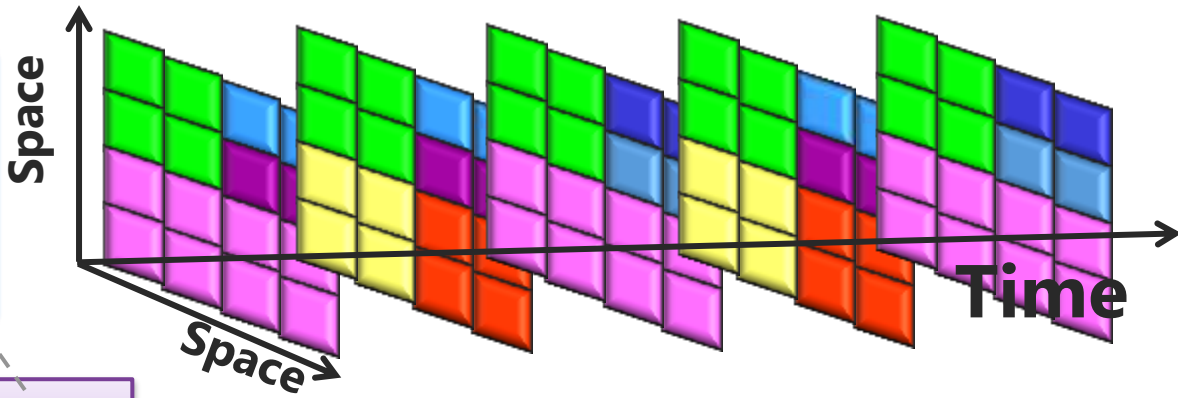
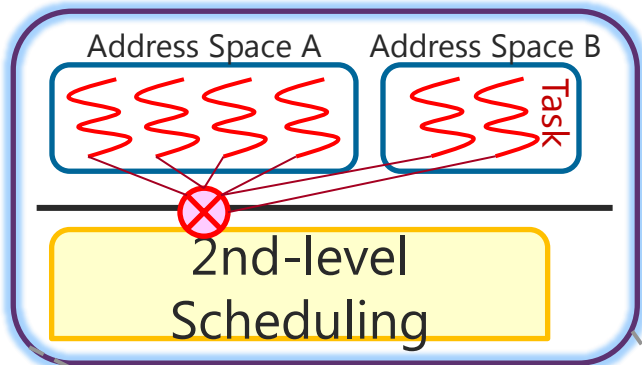
Allow maximum efficiency and expressibility in specializers by avoiding mandatory intermediary layers

# Communication-avoiding algorithms (Jim Demmel)

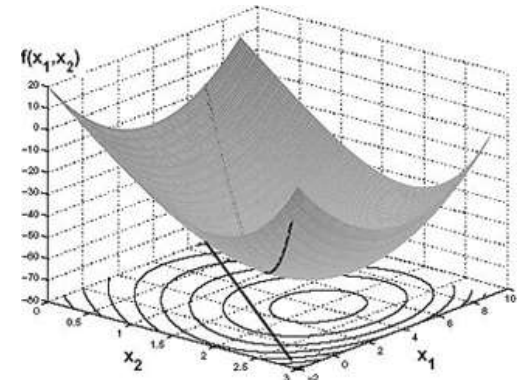
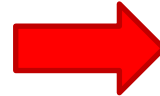
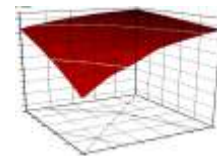
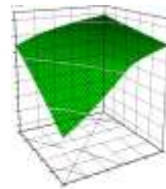
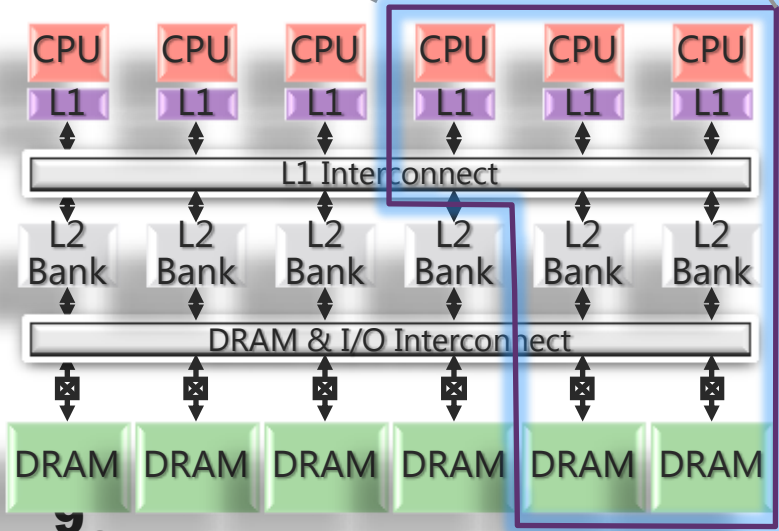
- Possible to reduce communication to theoretical minimum in various linear algebra computations
  - Parallel:  $O(1)$  or  $O(\log p)$  messages to take  $k$  steps, not  $O(k)$  or  $O(k \log p)$
  - Sequential: move data through memory once, not  $O(k)$  times
  - Lots of speed up possible (modeled and measured)
- Lots of related work
  - Some ideas go back to 1960s, some new
  - Rising cost of communication forcing us to reorganize linear algebra (among other things!)



# Convex Optimization for Resource Management (Sarah Bird, Burton Smith)



Tessellation Kernel (Partition Support)



# UIUC UPCRC - AvaScholar: Immersive Environment for Education

## AvaScholar Instructor

Real-Time Deformable Stereo and  
Shape-from-Motion Reconstruction of  
Instructor and Visual Aids



## AvaScholar Student

Real-Time Agglomeration of  
Demographics, Engagement and  
Confusion of Remote Students



# UPCRC Barcelona UPC

- **16 papers** in peer-reviewed Conferences, Journals and Workshops.
- **Best paper awards** in ICPE (paper on the publicly available TM benchmark suite) and in GLSVLSI (paper on detailed circuit design of a practical TM-cache)
- **First PhD graduates:** Sasa Tomic, Ferad Zyulkyarov and Nehir Sonmez
- **FP7 VELOX FP7** project on Transactional Memory included implementation and evaluation of AMDs Advanced Synchronization Facility (ASF) Hardware TM.
- **Transactional Memory:** Atomic Dataflow Model with best of optimistic concurrency and dataflow programming. Promising results with a game server application. Concluded work on dynamically adapting per-core hardware priorities to load-balance sibling threads for the STM2 framework.
- **Collaboration with Koç University** on TM based data race detection.
- **Low-power vector processors:** first release of benchmark software. New applications in the benchmark, speech synthesis and recognition, face recognition, TPC-H and artificial intelligence.

# Outline

**The Microsoft-Intel UPCRCs**

**Parallel Programming for Morts**

**Parallel Computing and the Cloud**

**Data-Intensive Science**

**The Future?**



# HPC and the 'Excluded Middle'

- **Consumer applications**
  - Access to diverse software plus a wide array of devices and a rich and responsive software developer ecosystem
- **Top 500 HPC applications**
  - Experienced computational scientists needed to create and maintain applications in energy, environment, climate change, computational fluid and structural dynamics, biological models ...
- **The 'Excluded Middle'**
  - Between ubiquitous consumer software and specialized high-end HPC applications is the world of day-to-day computational science problems - 'the Bottom 50,000 HPC applications'
  - The Excluded Middle is a "no man's land" that not only lacks the ready availability of application software but also simple parallel programming tools and methodologies for 'ordinary' developers.
  - The result has been limited uptake of computational science by many scientists and by companies of all sizes, large, medium and small.

**After Dan Reed's CACM Blog – see:**

**[www.hpcdan.org/reeds\\_ruminations/2010/10/hpc-and-the-excluded-middle.html](http://www.hpcdan.org/reeds_ruminations/2010/10/hpc-and-the-excluded-middle.html)**

# Einstein, Elvis and Mort

**Goal: Simplify parallel programming for 'ordinary mortals'**

## **Start with Abstractions**

- Parallel speedups for data parallel computations
- Independent loops
- Coarse-grained task parallelism
- Patterns, not primitives
- DAG model

## **Only later teach how to optimize applications**

- Need to look below abstractions to understand performance  
e.g. caching behavior, bandwidth and latencies

# Parallel Programming with ...

## .NET

Task Parallel Library

PLINQ

Samples for C#, VB & F#

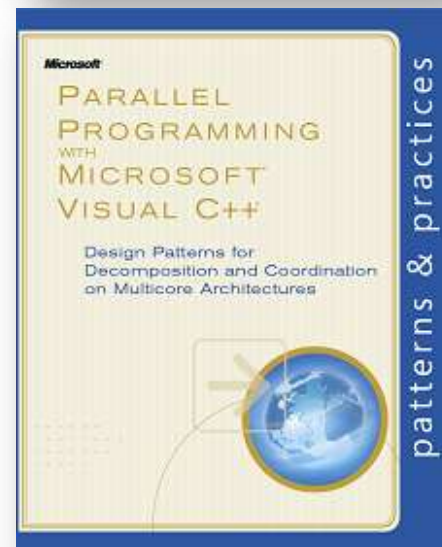
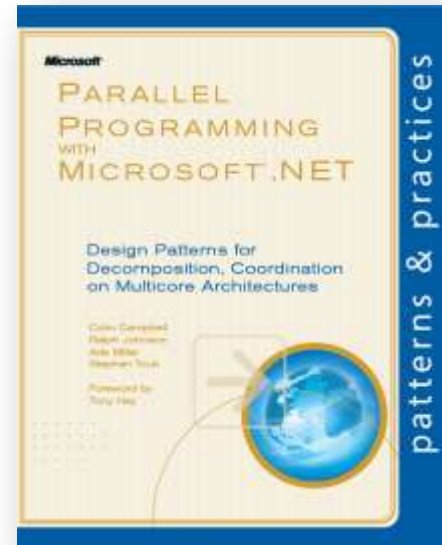
<http://parallelpatterns.codeplex.com/>

## Visual C++

Asynchronous Agents Library

Parallel Patterns Library

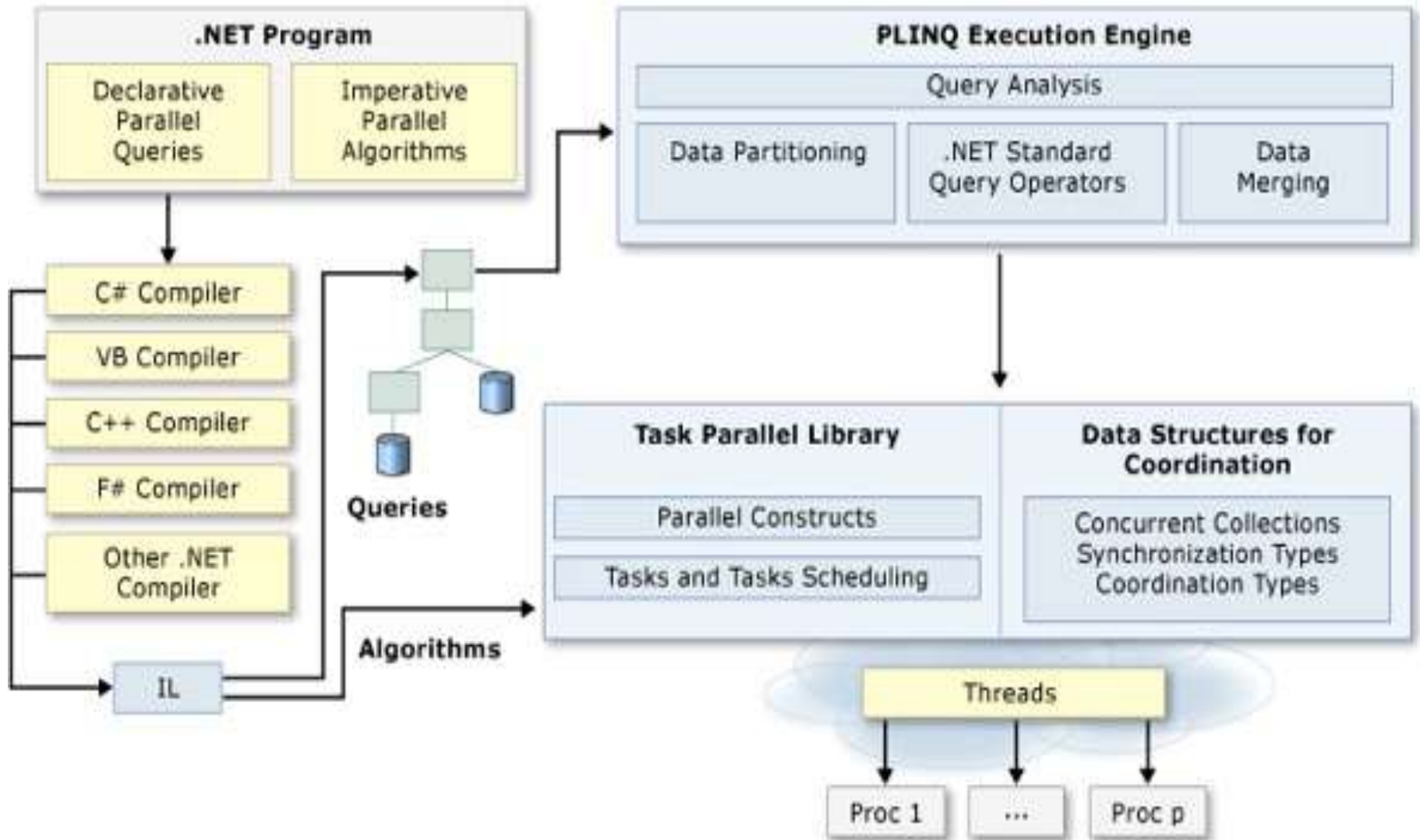
<http://parallelpatternscpp.codeplex.com/>



Microsoft  
**patterns &  
practices**



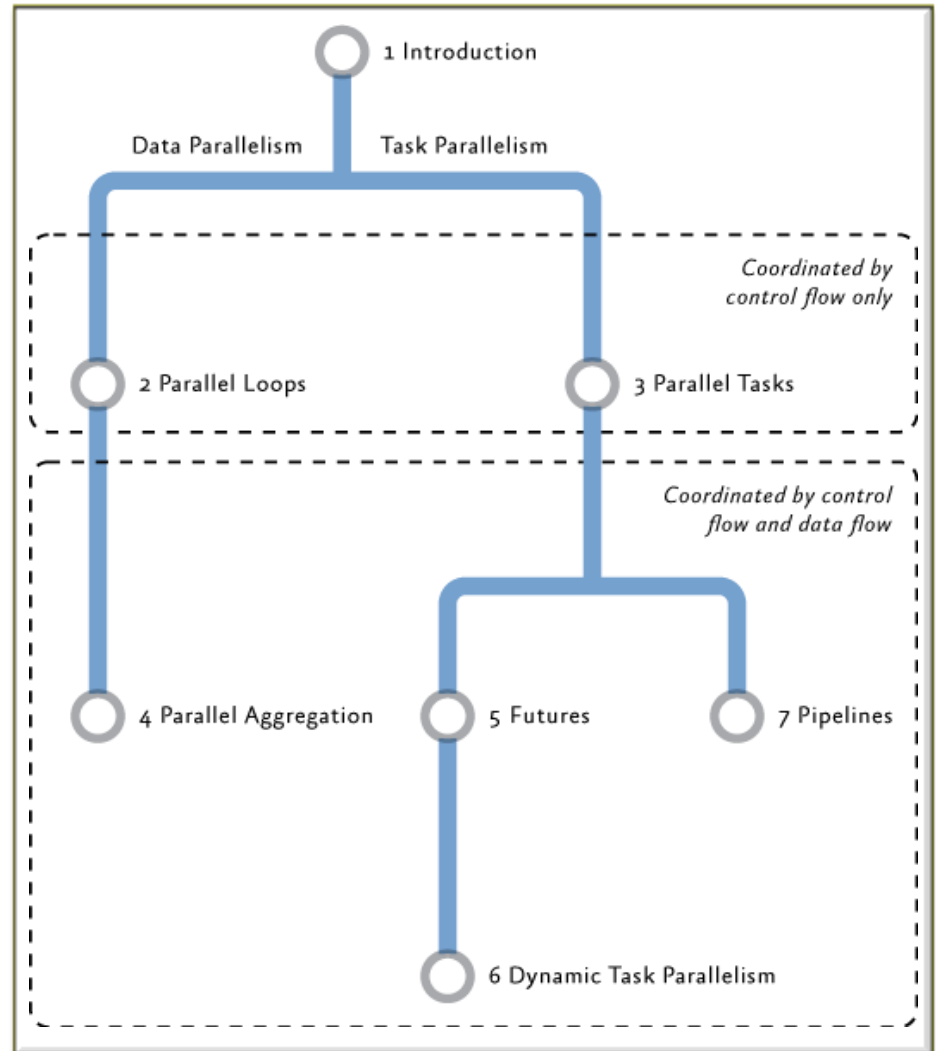
# Based on .NET CLR and Libraries





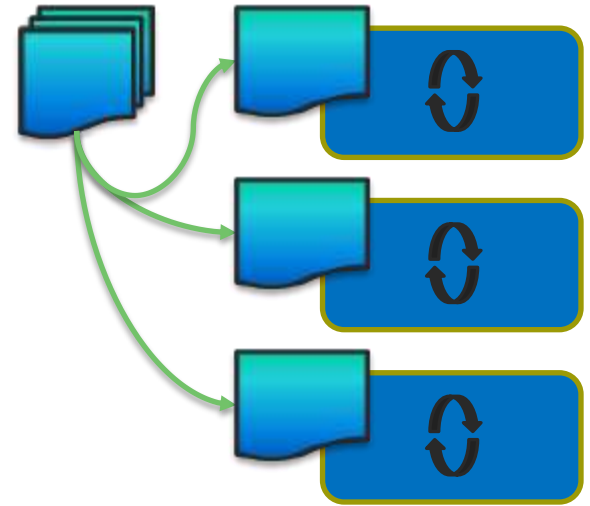
# Finding Potential Parallelism

- **Tasks vs. Data**
- **Control Flow**
- **Control and Data Flow**



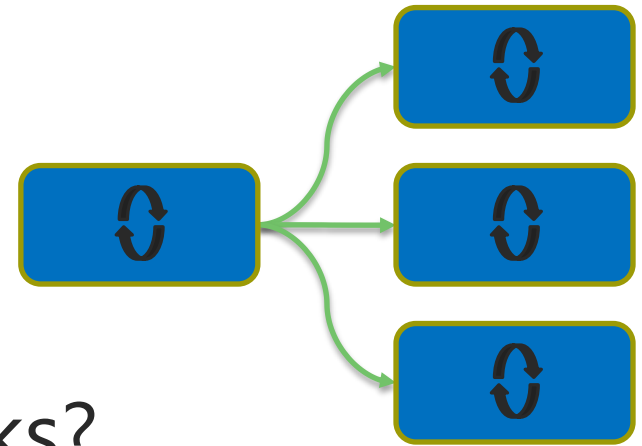
# Data Parallelism

- Data “chunk” size?
  - Too big – under utilization
  - Too small – thrashing
- Chunk layout?
  - Cache and cache line size
  - False cache sharing
- Data dependencies?



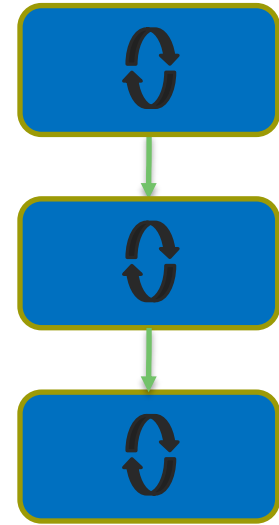
# Task Parallelism

- Enough tasks?
  - Too many – thrashing
  - Too few – under utilization
- Work per task?
  - Small workloads
  - Variable workloads
- Dependencies between tasks?
  - Removable
  - Separable
  - Read only or read/write



# Control and Data Flow

- Task constraints
  - Temporal:  $A \rightarrow B$
  - Simultaneous:  $A \leftrightarrow B$
  - None:  $A \ B$
- External constraints
  - I/O read or write order
  - Message or list output order
- Linear and irregular orderings
  - Pipeline
  - Futures
  - Dynamic Tasks



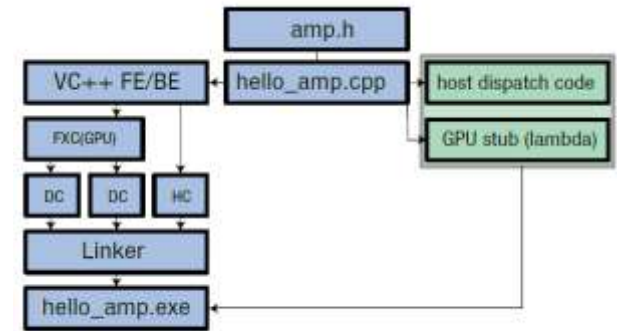
# C++ AMP (Accelerated Massive Parallelism)

AMP is a parallel programming model for heterogeneous computing on Windows

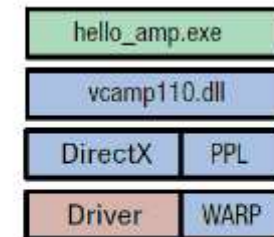
- Cross-vendor, open specification
- Generic data parallel API
- Extends modern C++
- Builds on PPL (Parallel Patterns Library)

## Example:

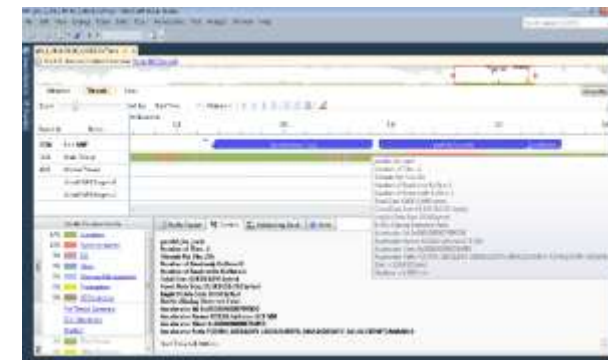
```
#include <amp.h>
array_view<particle_t, 1> particles_av(n, particles);
for( int step = 0; step < NSTEPS; step++ )
{
    parallel_for_each(particles_av.extent, [=] (index<1> i) restrict(amp){
        particles_av[i].ax = 0;
        particles_av[i].ay = 0;
        for (int j = 0; j < n; j++) {
            apply_force( particles_av[i], particles_av[j] );
        }
    });
    parallel_for_each(particles_av.extent, [=] (index<1> i) restrict(amp){
        move( particles_av[i], size );
    });
    particles_av.synchronize();
}
```



C++ AMP compilation and execution



C++ AMP software stack



# Parallel Processing Language Features

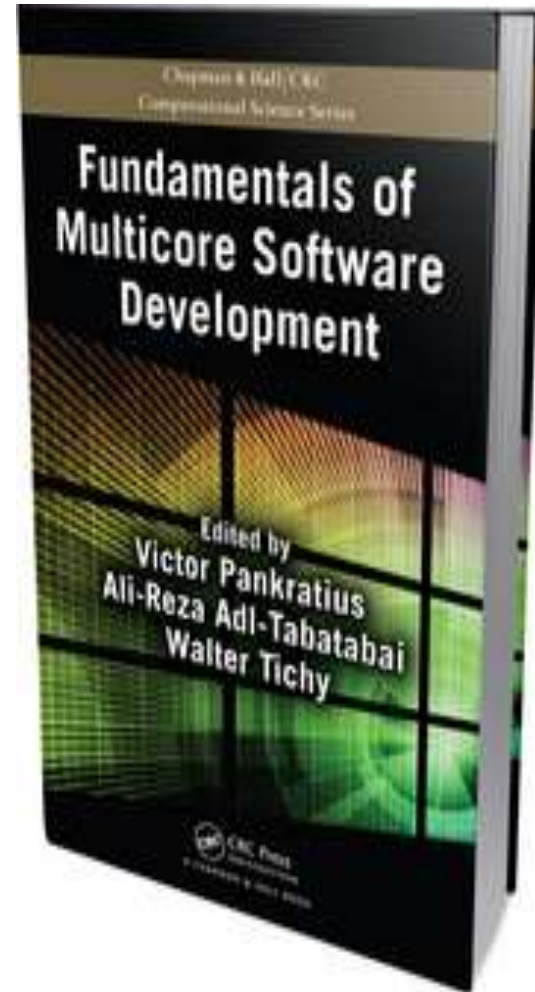
Covered in the new book:

Two examples:

- .NET 4 platform's TPL and PLINQ
- Java 5's concurrency package and proposals for Java 7

Also:

- F#3.0 - the Type Provider mechanism, and a set of built-in type providers for enterprise and web data standards



# Outline

**The Microsoft-Intel UPCRCs**

**Parallel Programming for Morts**

**Parallel Computing and the Cloud**

**Data-Intensive Science**

**The Future?**



# Parallel Computing in the Cloud

- Understanding the architecture of a big cloud data center.
- How is the data center different from a traditional cluster?
- Cloud parallel apps are more about data analysis than simulation.
- MapReduce is well suited to the cloud and is a critical kernel for most data apps.
- Some science examples



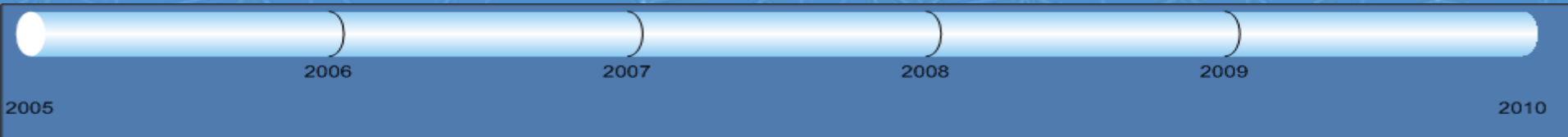


# Cloud Computing: One Definition

For the US National Institute of Standards and Technology (NIST), Cloud Computing means:

- On-demand service
- Broad network access
- Resource pooling
- Flexible resource allocation
- Measured service

# Microsoft's Datacenter Evolution



Datacenter  
Co-Location  
Generation 1



Quincy and San  
Antonio  
Generation 2



Chicago and Dublin  
Generation 3



Modular Datacenter  
Generation 4



Facility PAC

Deployment Scale Unit



Server

*Capacity*



Rack

*Density  
and Deployment*



Containers

*Scalability and  
...Sustainability*



IT PAC

*Time to Market  
Lower TCO*

# Microsoft Cloud built on Data Centers

~100 Globally Distributed Data Centers ranging in size from "edge" facilities to mega-scale centers (100K to 1M servers)



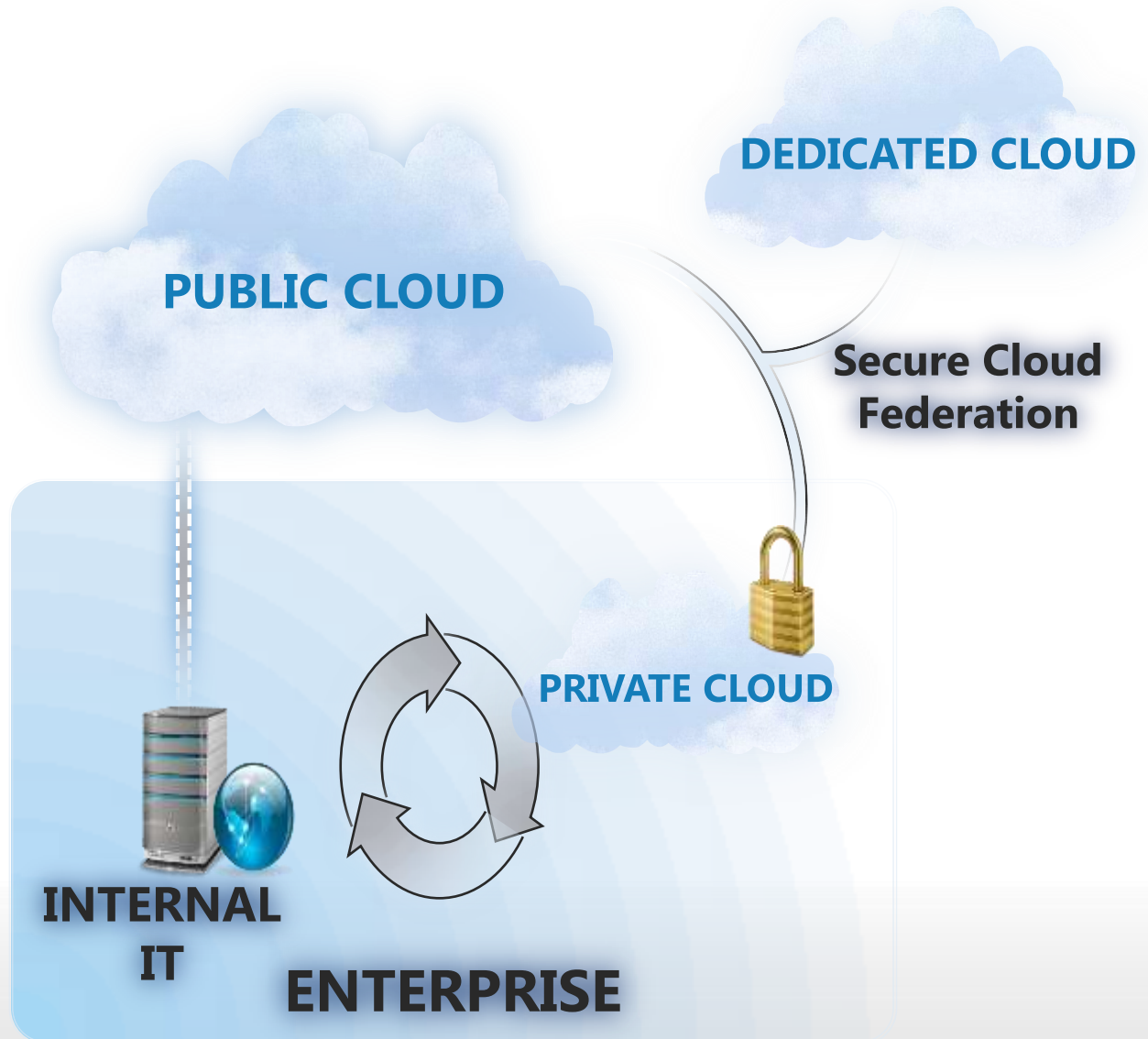
Quincy, WA



Location 4 DCs



# Cloud Options?





# Data Center vs Supercomputers?

## • Network Architecture

Supercomputers:

CLOS “Fat Tree” infiniband  
Low latency – high bandwidth  
protocols

Data Center: IP based

Optimized for Internet Access

## • Data Storage

Supercomputers:

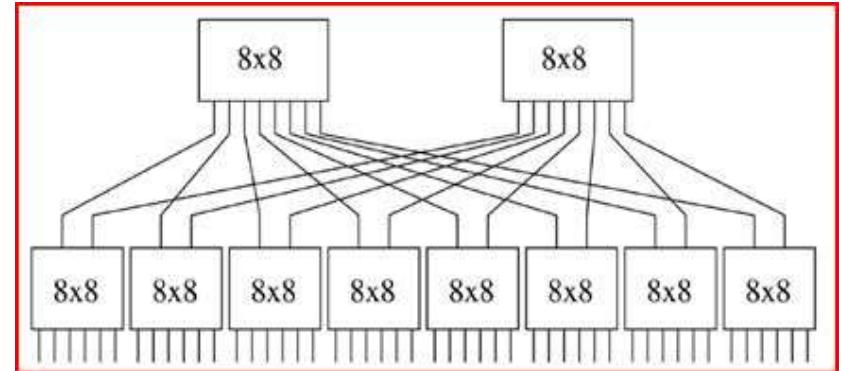
separate data farm  
GPFS or other parallel file  
system

Data Centers

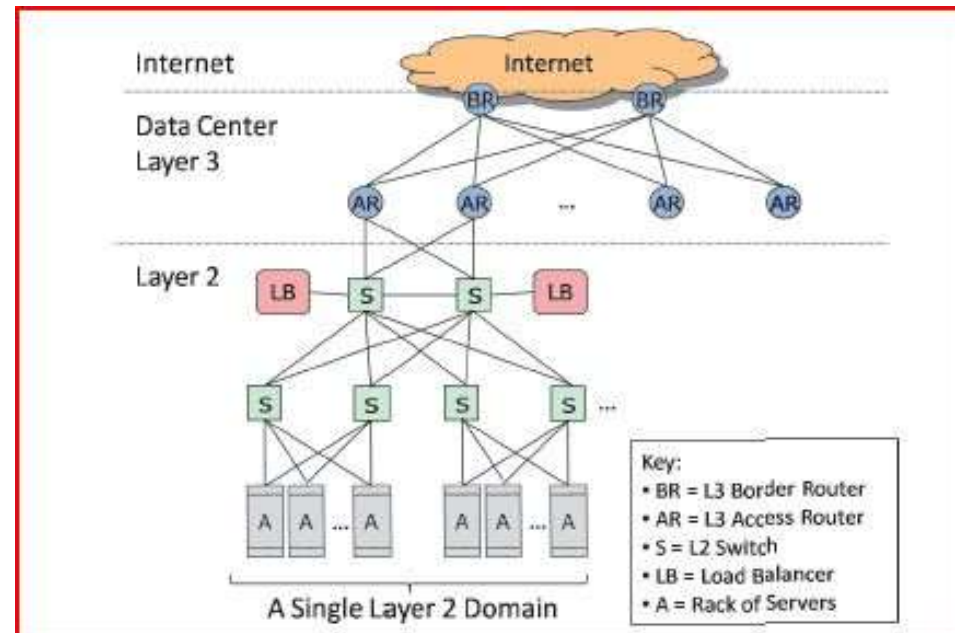
use disk on node +  
memcache + databases

➤ Expect to see future  
convergence!

Fat tree network



Standard Data Center Network



# Types of Cloud Services

## Infrastructure as a Service (IaaS)

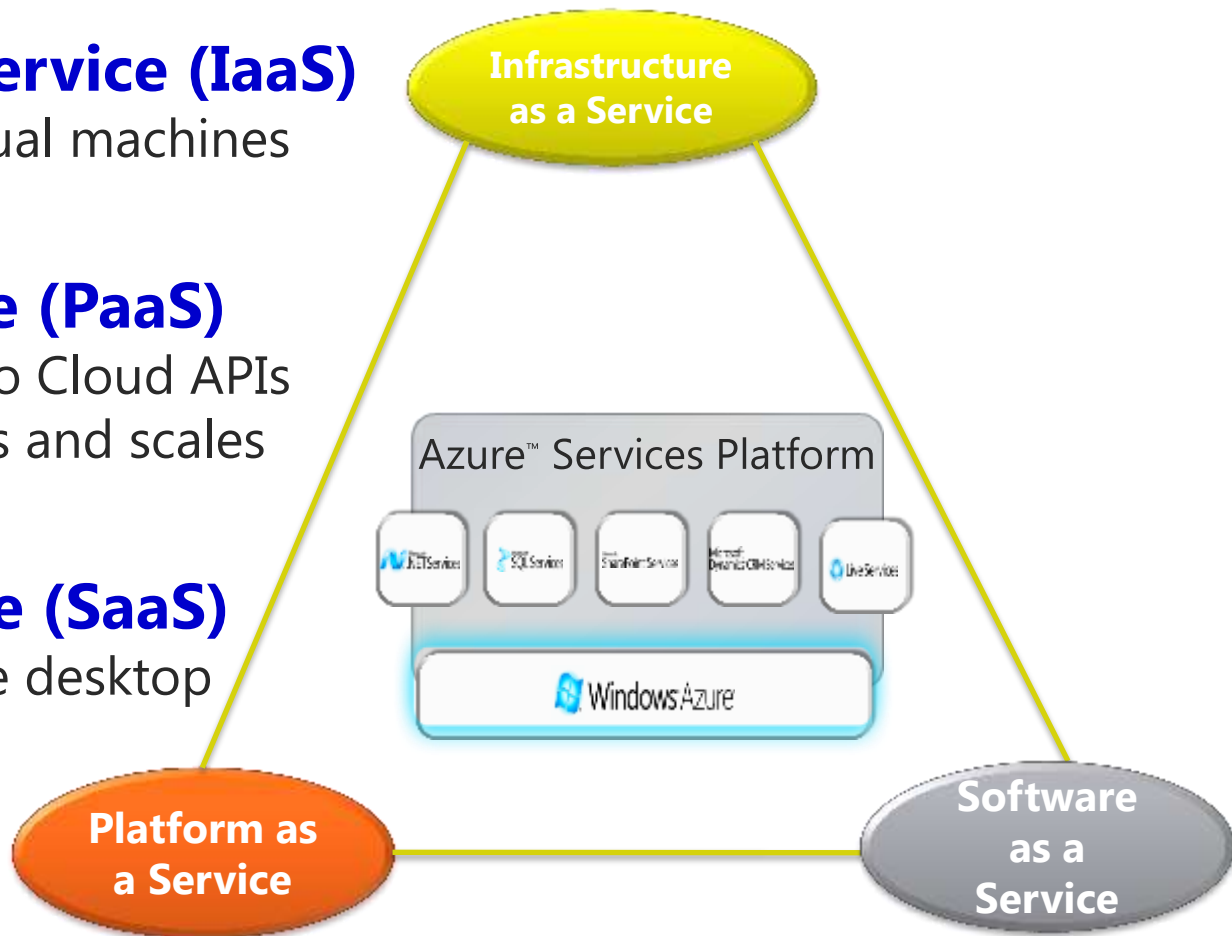
Provide a way to host virtual machines on demand

## Platform as a Service (PaaS)

You write an Application to Cloud APIs and the platform manages and scales it for you.

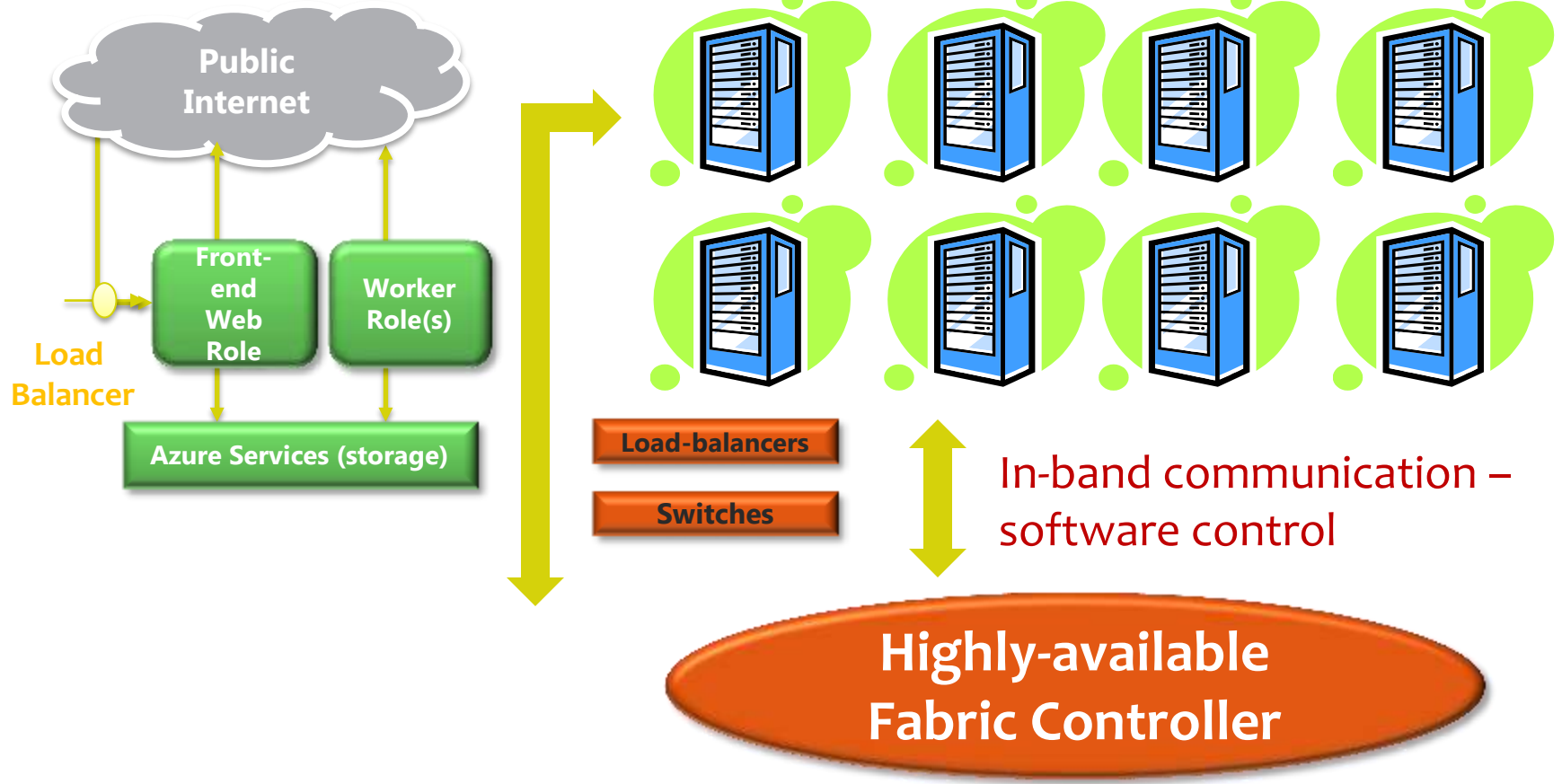
## Software as a Service (SaaS)

Delivery of software to the desktop from the Cloud



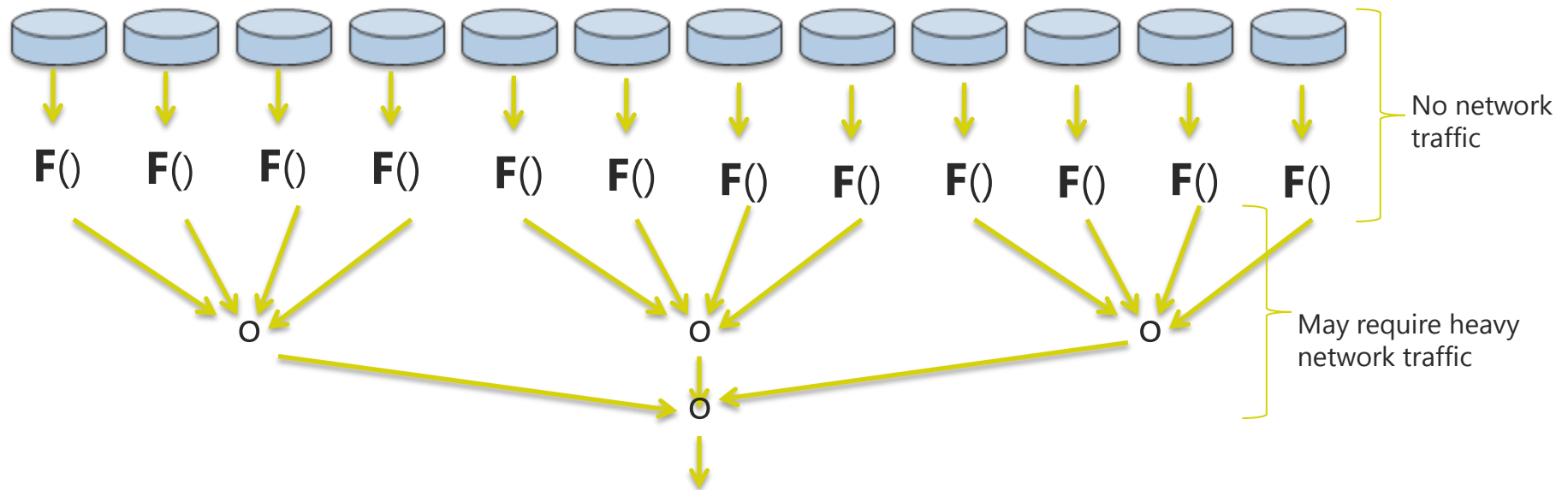
# Azure Cloud Programming Model

## Abstract Programming Model:



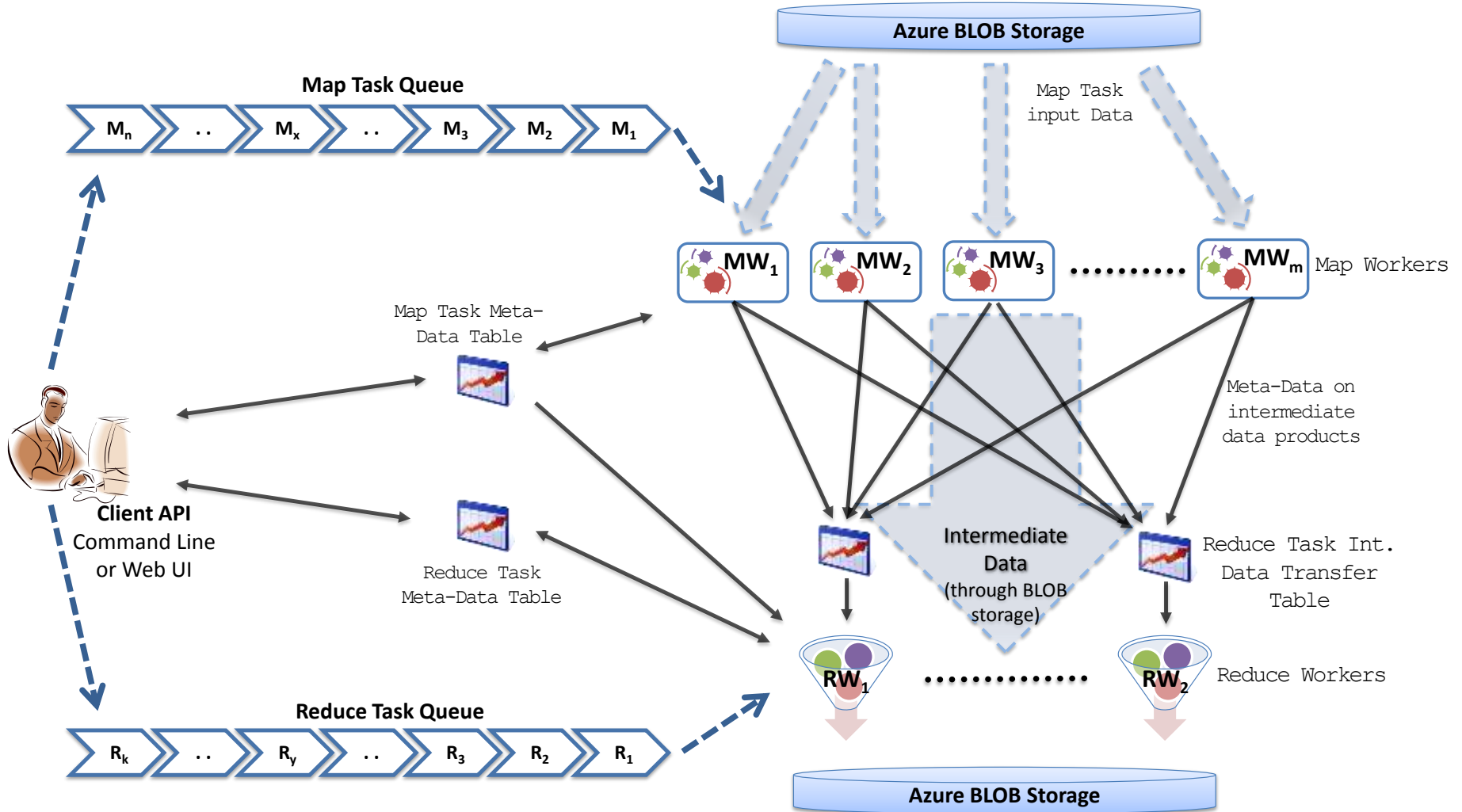
# “Traditional” Cloud Parallel Apps

- Based on ensemble “parameter sweep” or MapReduce paradigm
- Start with a distributed data collection and apply function  $F()$  in parallel and reduce with an associate operator  $O$ .





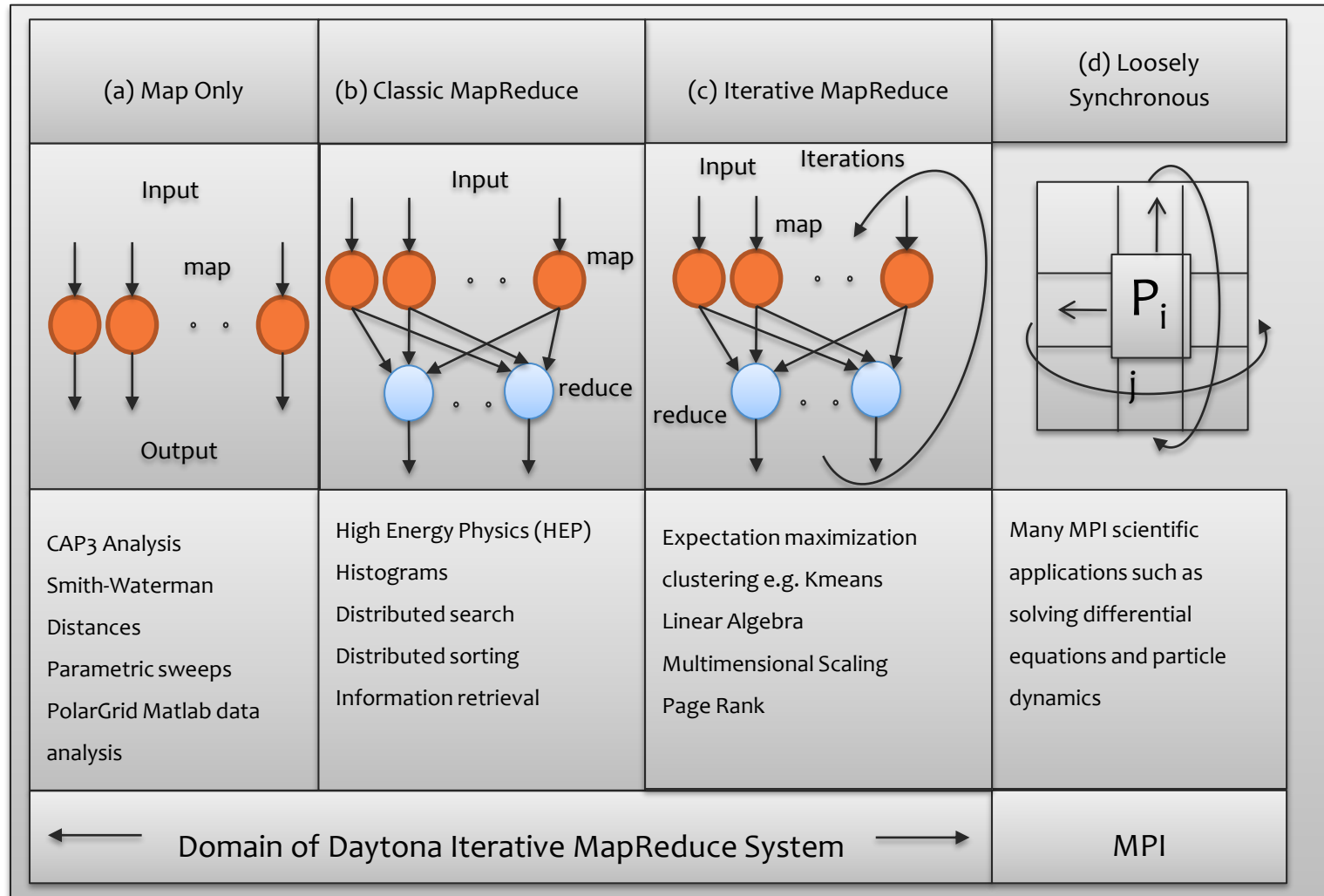
# Twister: Iterative Map-Reduce



Thanks to Geoffrey Fox

# 'Daytona' Iterative MapReduce

Microsoft Research now released optimized version of Fox's Twister algorithm on Azure



# Microsoft Azure Research Projects

90 projects world wide



# Sample Projects on Azure Cloud

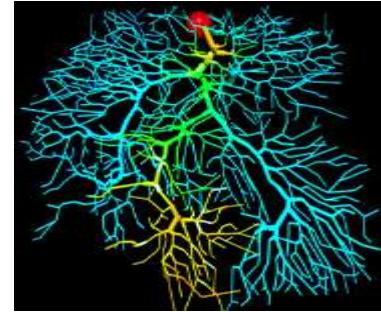
- **Protein Folding**

- The University of Washington is studying the ways proteins from salmonella virus inject DNA into cells. Used 2000 concurrent cores. PI: Nikolas Sgourakis, Baker Lab.



- **Joint Genetic and Neuroimaging Analysis**

- France's premier research institute INRIA is using 1000 cores of Azure to study large cohorts of subjects to understand links between genetic patterns and brain anomalies. Pis: Radu Marius Tudoran, Gabriel Antoniu IRISA INRIA France.



- **Fire Risk**

- This app from the University of Aegean estimates the fire risk probability using meteo and geo-data sources and calculates the so-called fire risk index. A client application for the fire and forest services as well as cloud services that allow access to real-time data from sensors and on-the-ground reporting. This service has been tested and validated with fire-fighting crews in both Mytilene and Thessaloniki, Greece PI: Kostas Kalabokidis



# More Examples

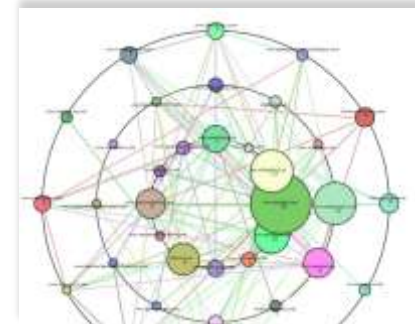
- **Drug Discovery**

- Researchers at Newcastle University in the U.K. are using Azure to model the properties (toxicity, solubility, biological activity) of molecules for potential use as drugs. This cloud solution is primarily aimed at domain scientists who do not have advanced IT skills. PI: Paul Watson



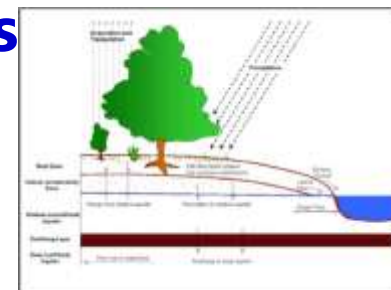
- **Predicate-Argument Structure Analysis**

- University of Kyoto team applied a predicate-argument structure analysis to a huge Japanese corpora consisting of about 20 billion web sentences, to improve the open-search engine infrastructure TSUBAKI, which is based on deep natural language processing. To achieve this goal 10,000 core on Windows Azure were used in a massively parallel computation that took about a week.



- **Model and Manage Large Watershed Systems**

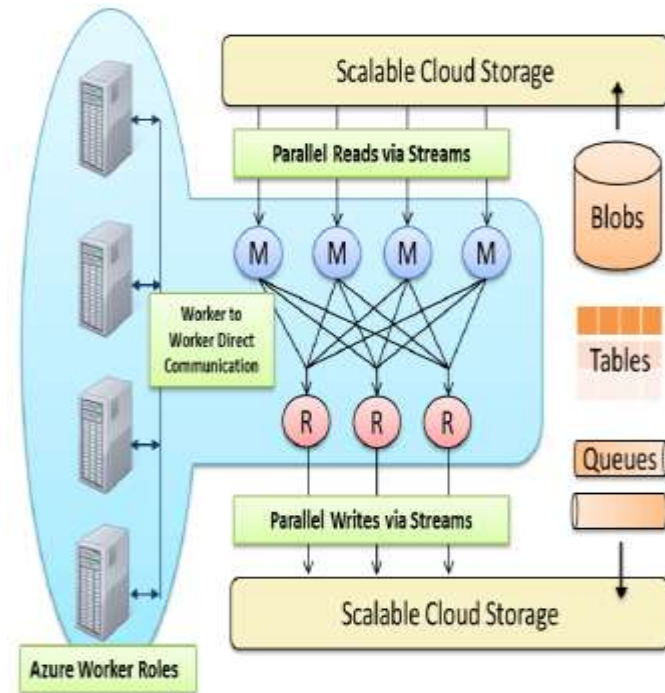
- Predict the impact of land use change and climate change on water resources. Going beyond modeling to include entire workflow from data collection to decision making. University of Virginia and South Carolina. Using HPC Scheduler on Azure to launch thousands of analysis jobs.





# What have we learned?

1. Traditional communication-intensive MPI apps belong on supercomputers.
2. The cloud advantage
  - Applications that require sharing and web access
  - Massive “Map Reduce” data analytics on cloud resident data
  - Massive ensemble computations
3. The scale-out as needed model works.
  - Users prefer spending on pay-as-you-go cloud to buying and cluster hardware. (use 500 cores 2 days a week vs. 120 cores purchased)
  - Most researcher prefer to avoid maintaining cluster and data storage facilities.
  - Most users do not have access to supercomputers



# Outline

**Moore's Law and the Multicore Revolution**

**Parallel Programming for Morts**

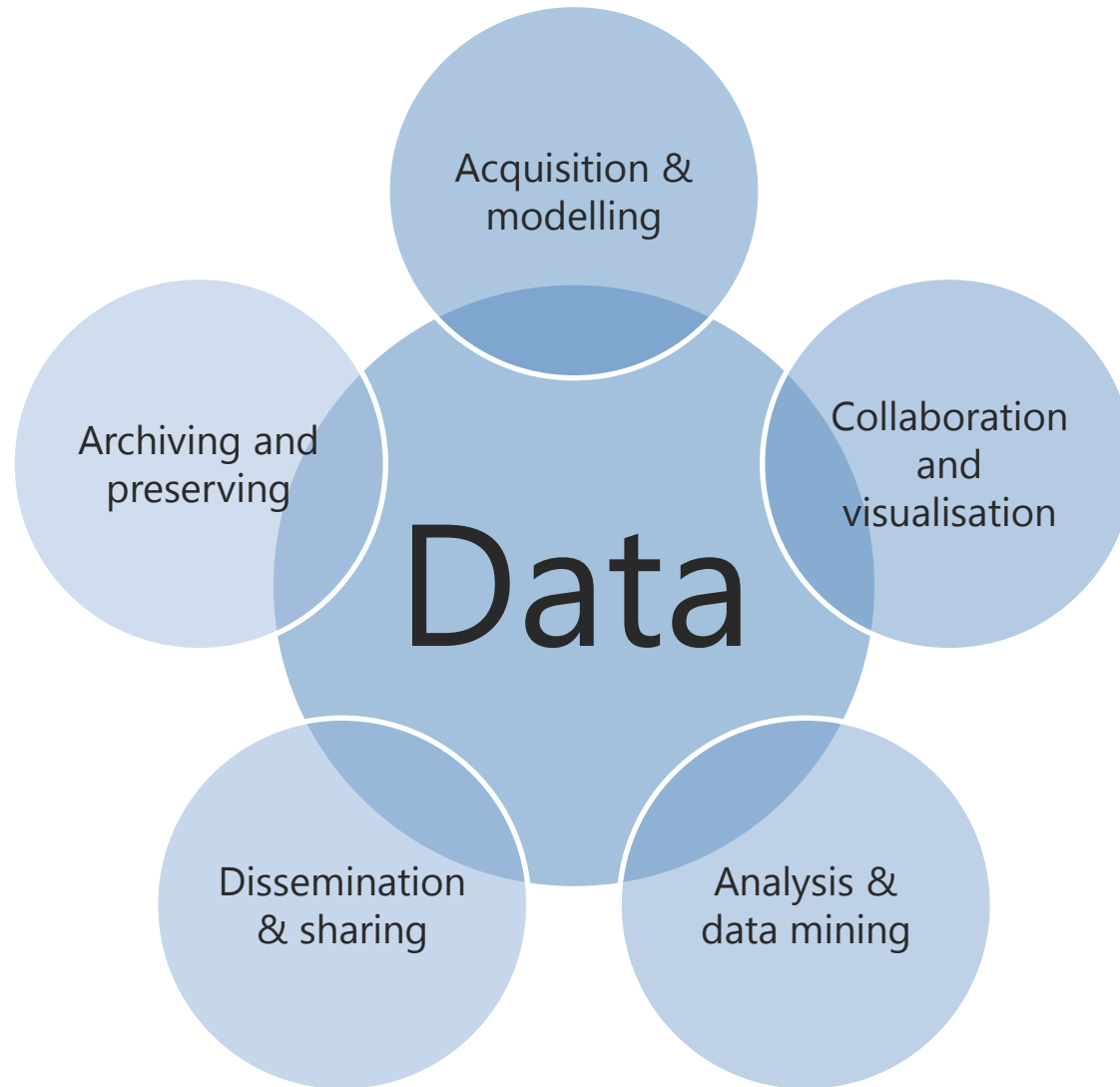
**Big Data and the Cloud**

**Data-Intensive Science**

**The Future?**



# The Fourth Paradigm: Data-intensive Science





# Need for more powerful tools ...

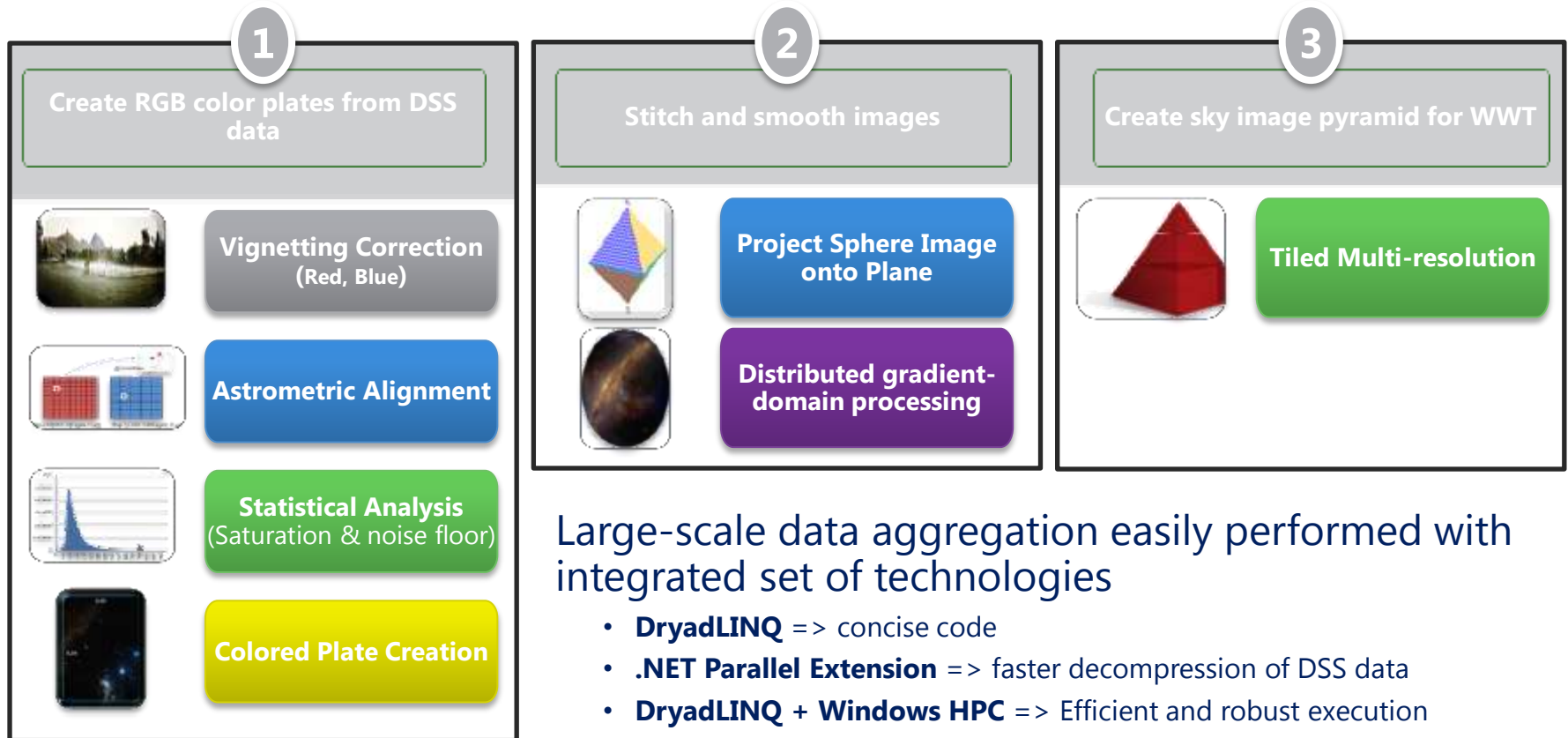
- Increasing scale and heterogeneity of data makes it more difficult to assemble, process and reduce data to derive science results
- Three significant barriers:
  - **Resources**
    - Many research groups do not have access to sufficient computational and data resources
  - **Complexity**
    - Coordination to manage data, schedule jobs, support fault tolerance becomes increasingly complex
  - **Tedium**
    - Copying 10 files OK; copying more than 1000 problematic

# Two Examples

- Terapixel Image Project
  - Use MapReduce-like DryadLINQ technology
  - Use .NET parallel extensions for code running in parallel on multicore nodes of HPC Cluster
  - Use Trident Scientific Workflow technology from Microsoft Research to manage overall process
- MODIS Azure Project
  - Computing Evapotranspiration from MODIS satellite data using Azure Cloud resources

# Generating a TeraPixel Image

## Computational and Data Intensive Application



Large-scale data aggregation easily performed with integrated set of technologies

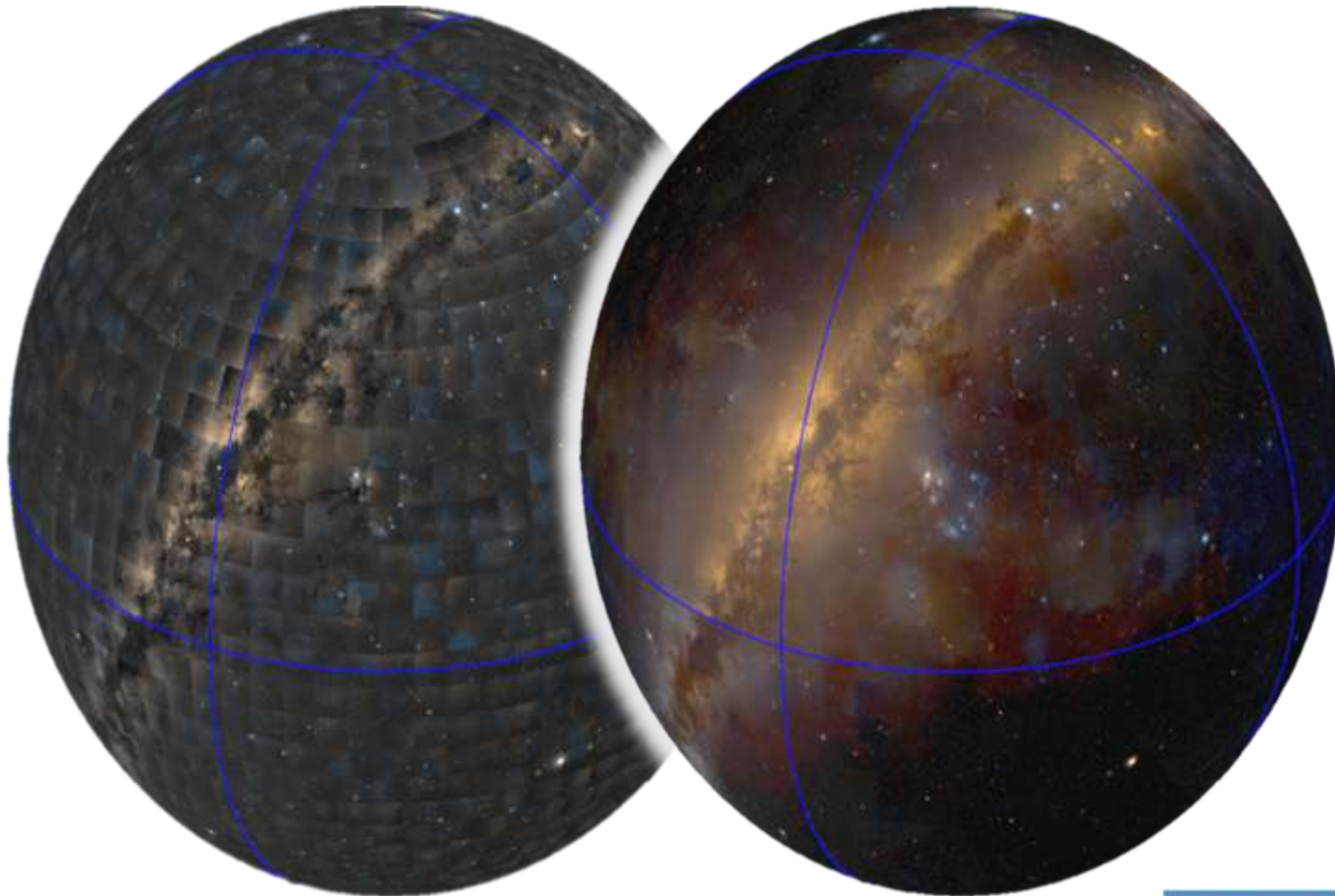
- **DryadLINQ** => concise code
- **.NET Parallel Extension** => faster decompression of DSS data
- **DryadLINQ + Windows HPC** => Efficient and robust execution

Managed and Coordinated by **Project Trident: A** Scientific Workflow Workbench

**SDSS Data: 3,120,100 files, 417 GB**

# TeraPixel Image for WorldWide Telescope

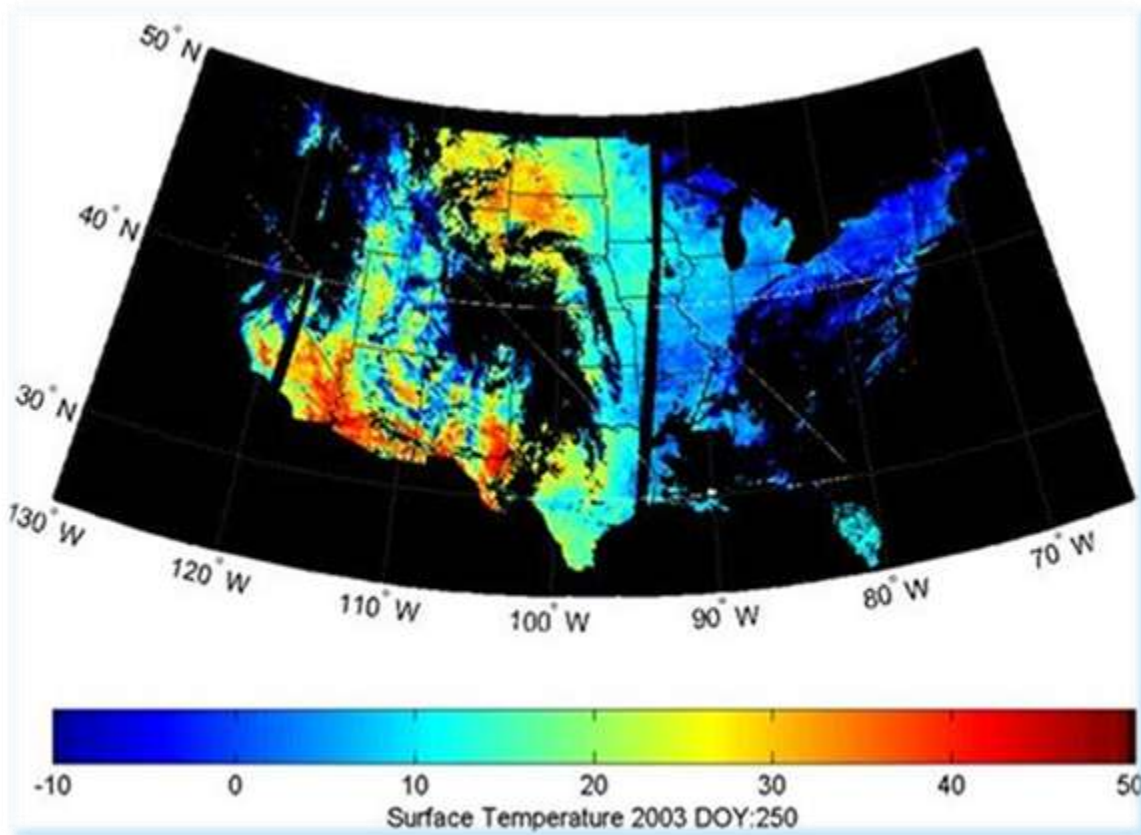
**Result:** *Largest, clearest, and smoothest sky image in the world*



## ***Special Thanks to***

- Brian McLean (Space Telescope Science Institute),
- Misha Kazhdan (Johns Hopkins University), Hugues Hoppe (MSR), and Dinoj Surendran (MSR)
- Dean Guo (MSR), Christophe Poulain (MSR)
- Aditi Team

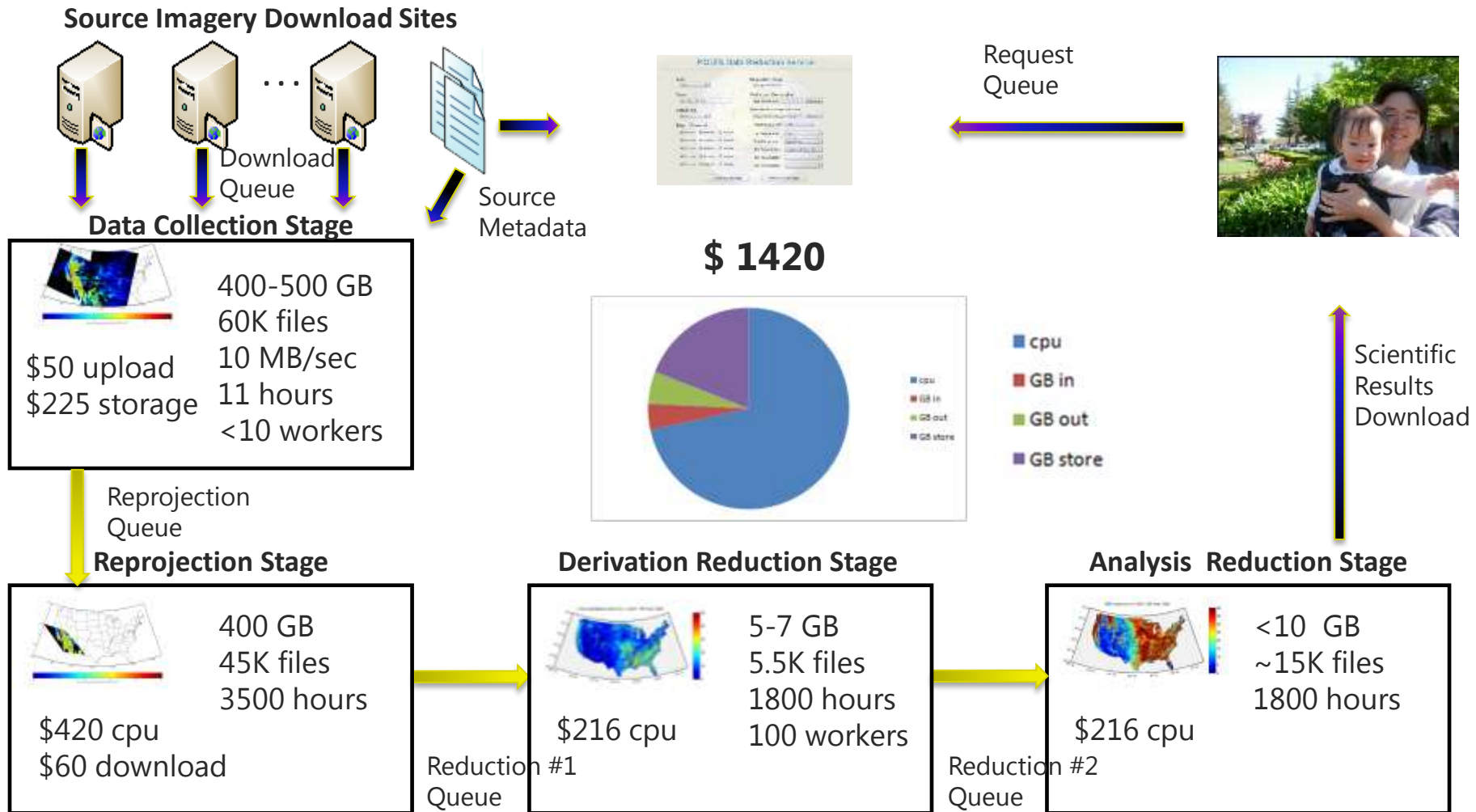
# MODIS Azure: *Computing Evapotranspiration (ET) in the Cloud*



A pipeline for  
download,  
processing, and  
reduction of diverse  
NASA MODIS  
satellite imagery

*Contributors: Catharine van Ingen (MSR), Youngryel Ryu (UC Berkeley), Jie Li (Univ. of Virginia)*

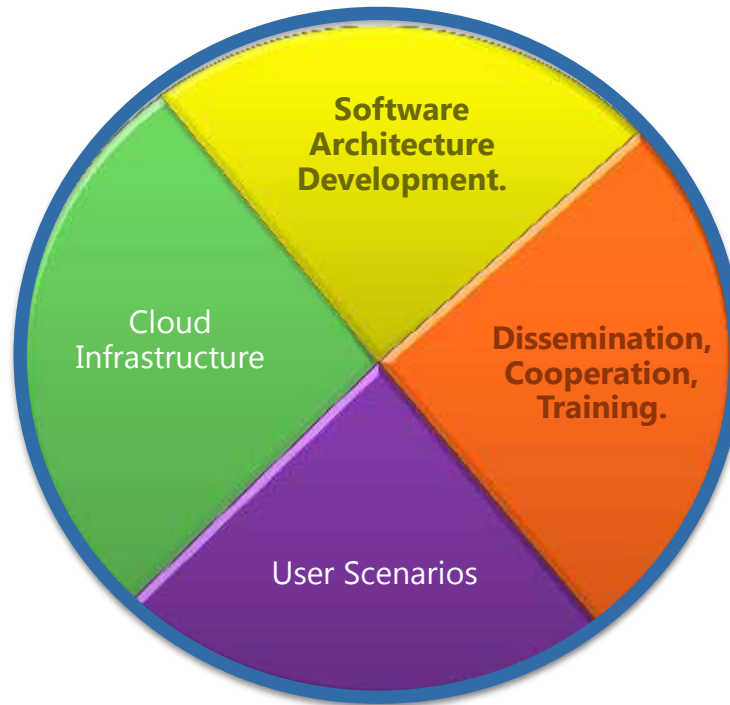
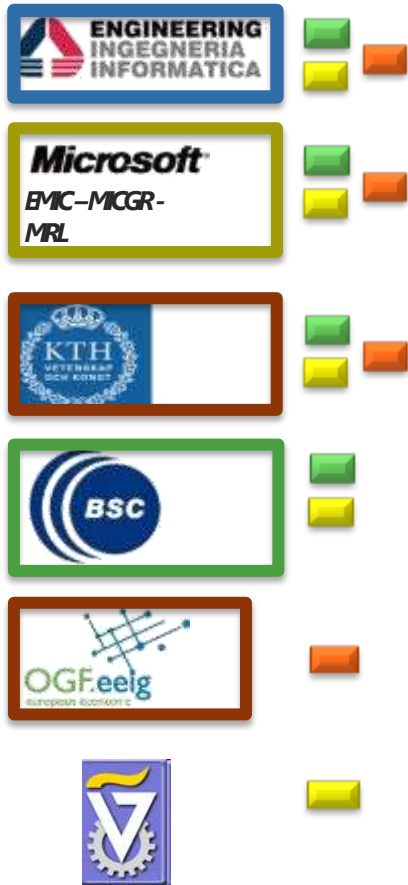
# Cost of Computing ET for 1 US Year





# EU VENUS-C Project:

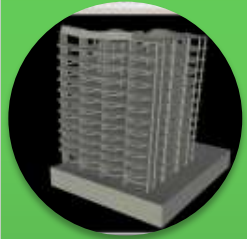
*Virtual multidisciplinary **EnviroN**ments  
USing **C**loud infrastructures*



**Project led by Fabrizio Gagliardi**



# Seven Pilot Scenarios



T5.1

Structural  
Analysis for  
Civil  
Engineering



T5.2

Building  
Information  
Management



T5.3

Data for  
Science -  
AquaMaps



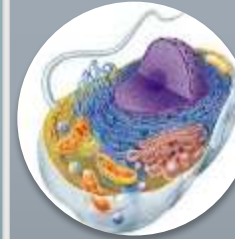
T5.4

Civil  
Protection  
and  
Emergencies



T5.5

Bioinformatics



T5.6

System  
Biology



T5.7

Drug  
Discovery



# Civil Protection Application: 'Greek Fire'



# Outline

**Moore's Law and the Multicore Revolution**

**Parallel Programming for Morts**

**Big Data and the Cloud**

**Data-Intensive Science**

**The Future?**







## Search Blog

# Introducing Schema.org: Bing, Google and Yahoo Unite to Build the Web of Objects

The Bing Team 6/2/2011 10:01 AM Comments (3)



We've been talking for a while about the need to rethink the search experience to better reflect both the changing web and advancing user habits.

One of the biggest challenges and opportunities we see is to literally create a high-definition proxy of the physical world inside of Bing. In other words, we want to be able to model the world in which we all live to the level that search can actually help you make decisions and get things done in real life by understanding all the options the world presents.

We've made great progress on the technical front to begin to model the real world from the messy bits of data scattered across the web. Things like movies have benefitted from this work. We're now able to understand "Casablanca" is a movie and literally mine the web to re-assemble information about that movie from millions of sites.

But we think we can do better. We want to enable publishers to give us hints about what things they are describing on their sites. Rather than rely solely on machine learning and other AI techniques, we asked "what if we could enable publishers to have a single schema they could use to describe their sites that all search engines could understand?"

## Getting started with schema.org

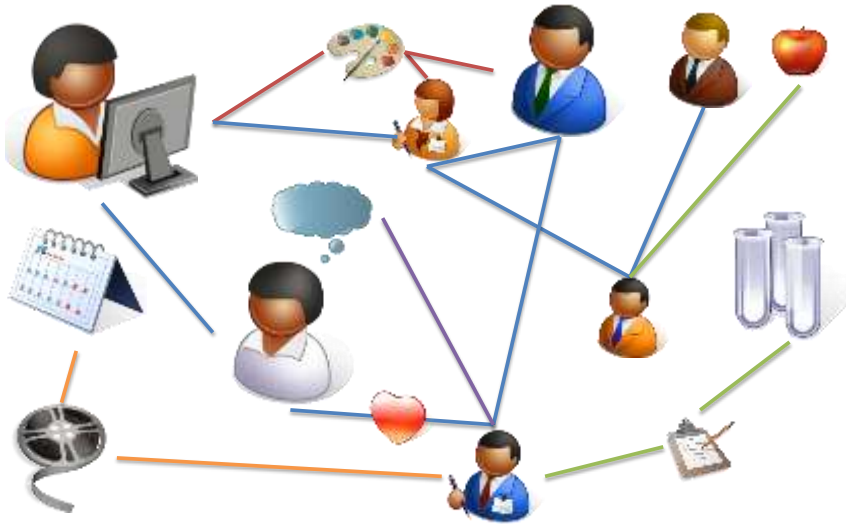
Most webmasters are familiar with HTML tags on their pages. Usually, HTML tags tell the browser how to display the information included in the tag. For example, `<h1>Avatar</h1>` tells the browser to display the text string "Avatar" in a heading 1 format. However, the HTML tag doesn't give any information about what that text string means—"Avatar" could refer to the a hugely successful 3D movie, or it could refer to a type of profile picture—and this can make it more difficult for search engines to intelligently display relevant content to a user.

Schema.org provides a collection of shared vocabularies webmasters can use to mark up their pages in ways that can be understood by the major search engines: Google, Microsoft, and Yahoo!

You use the [schema.org](#) vocabulary, along with the [microdata format](#), to add information to your HTML content. While the long term goal is to support a wider range of formats, the initial focus is on Microdata. This guide will help get you up to speed with microdata and schema.org, so that you can start adding markup to your web pages.

1. [How to mark up your content using microdata](#)
  - a. [Why use microdata?](#)
  - b. [itemscope and itemtype](#)
  - c. [itemprop](#)
  - d. [Embedded items](#)
2. [Using the schema.org vocabulary](#)
  - a. [schema.org types and properties](#)
  - b. [Expected types, text, and URLs](#)
  - c. [Testing your markup](#)
3. [Advanced topic: Machine-understandable versions of information](#)
  - a. [Dates, times, and durations](#)
  - b. [Enumerations and canonical references](#)
  - c. [Missing/implicit information](#)
  - d. [Extending schema.org](#)

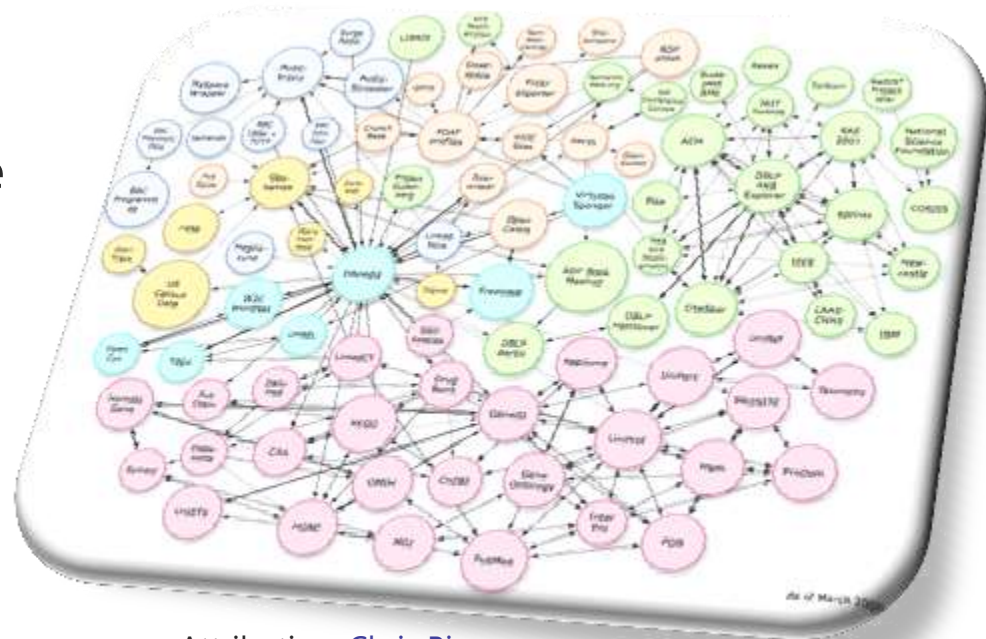
# Semantically linked data



- Data/information is interconnected through machine-interpretable information (e.g. **paper X is about star Y**)
- Social networks are a special case of 'data meshes'

## A knowledge ecosystem:

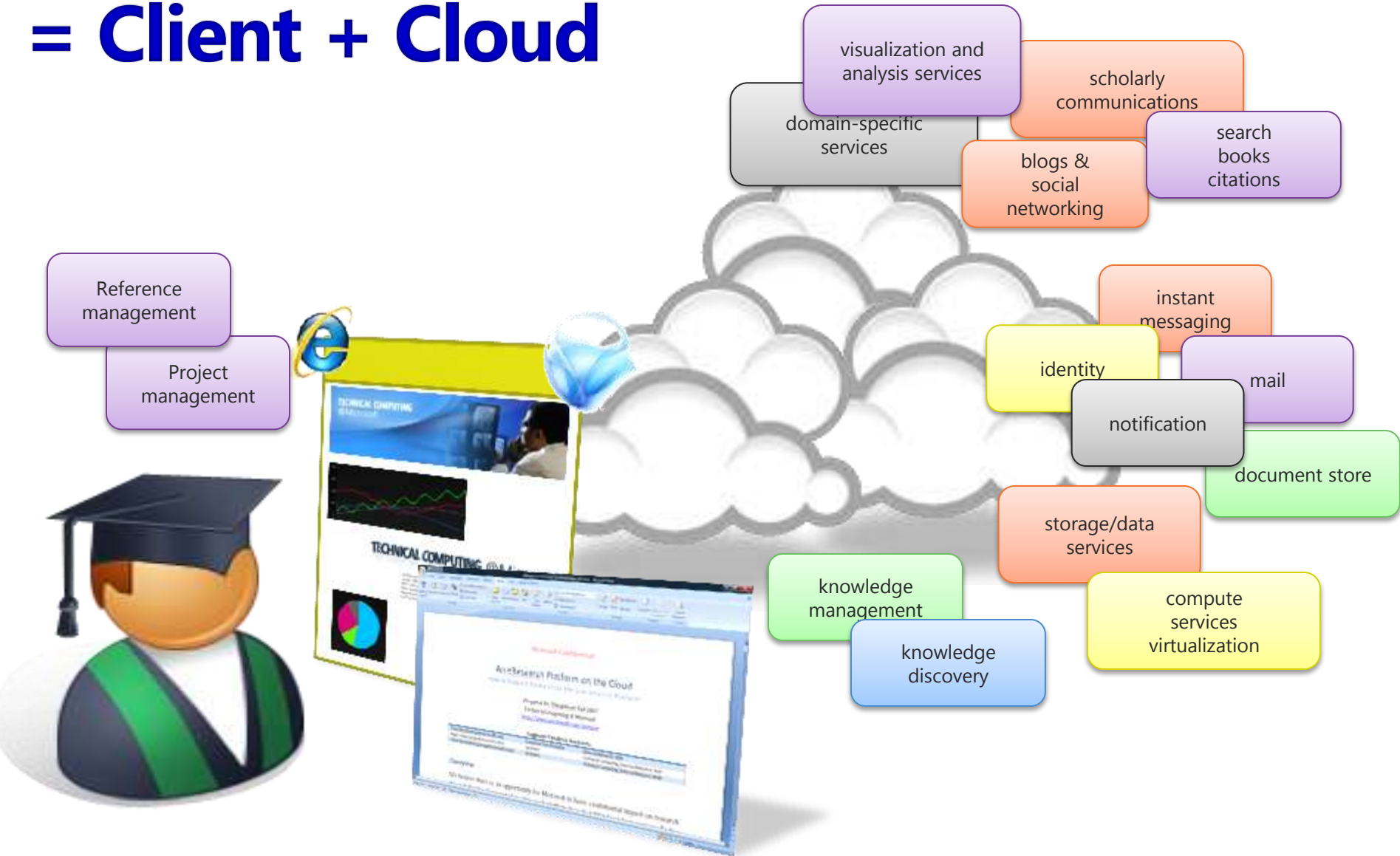
- A richer authoring experience
- An ecosystem of services
- Semantic storage
- Open, Collaborative, Interoperable, and Automatic



Attribution: [Chris Bizer](#)



# Future Research Infrastructure = Client + Cloud





# Acknowledgements

- Thanks to Roger Barga, Judith Bishop, Jim Larus, Ade Miller, Kenji Takeda, and Juan Vargas for their help in preparing this talk.
- They are not responsible for any misunderstandings or opinions!

# Resources

Microsoft Research

<http://research.microsoft.com>

Microsoft Research Downloads

<http://research.microsoft.com/accelerators>

Microsoft Research Connections

<http://research.microsoft.com/connections>

Science at Microsoft

<http://www.microsoft.com/science>

Python Tools for Visual Studio

<http://pytools.codeplex.com>

Outercurve Foundation

<http://www.outercurve.org>



***Microsoft***<sup>®</sup>

*Your potential. Our passion.*<sup>™</sup>