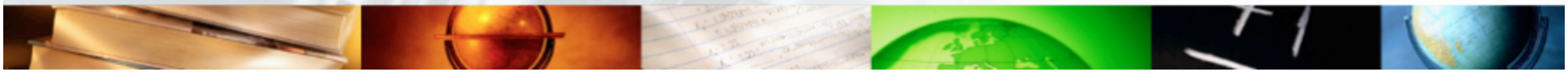
Dryad and DryadLINQ for data-intensive research (and a bit about Windows Azure)

Condor Week 2010, Madison, WI

Christophe Poulain

Microsoft Research



An Introduction

Microsoft External Research

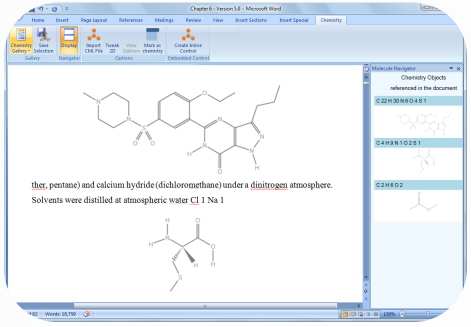
Tony Hey, Corporate VP

Work with the academic and research community to speed research, improve education, foster innovation, and improve lives around the world.

Core Computer Science	Earth, Energy & Environment	Education & Scholarly Communication	Health & Wellbeing

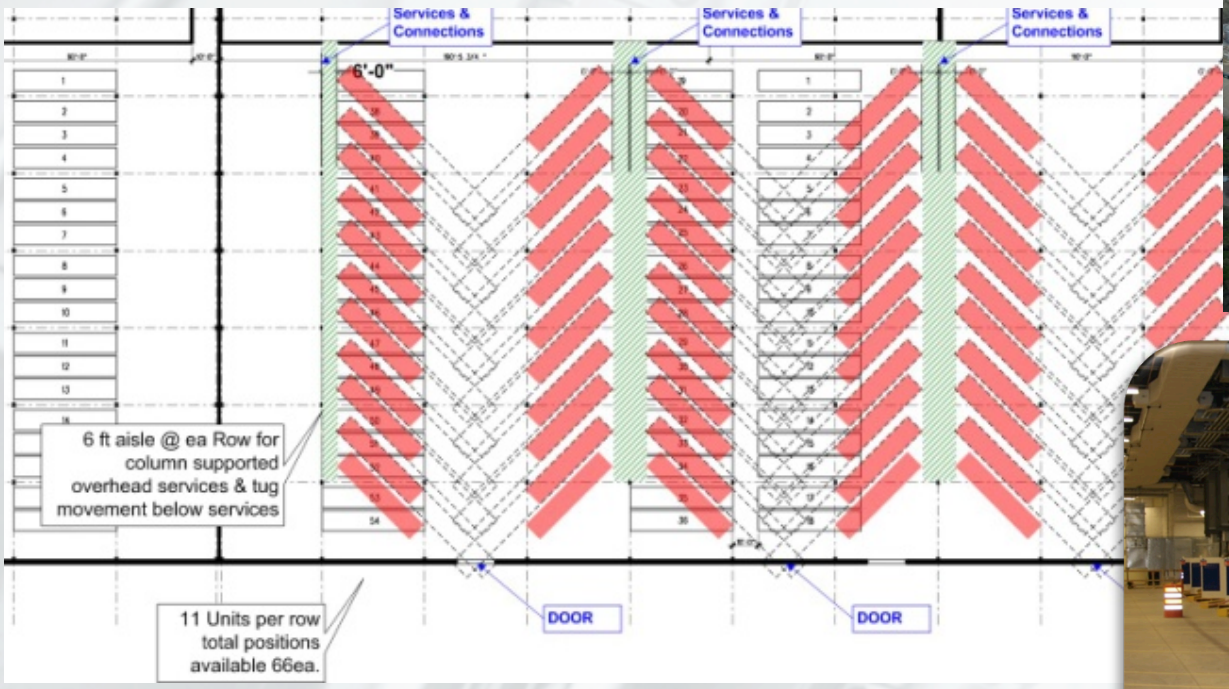
Advanced Research Tools and Services
http://research.microsoft.com/collaboration/tools

One of our aim is to help scientists spend less time on IT issues and more time on discovery by creating open tools and services based on Microsoft platforms and productivity software



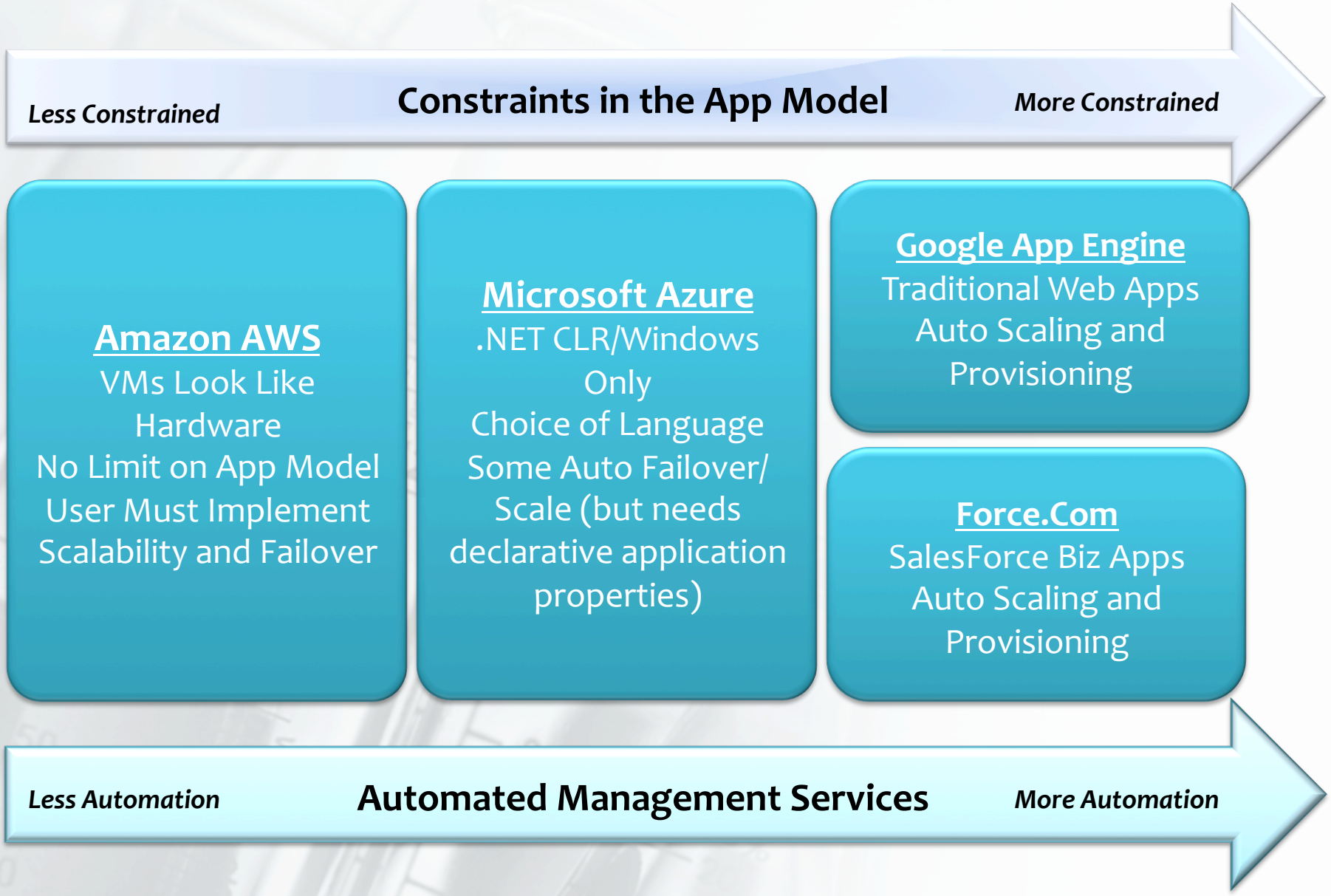
Windows Azure

It is Microsoft's cloud services platform, designed to host web services and applications in Microsoft-owned, Internet-accessible data centers.



112 containers x 2000 servers = 224000 servers

A Spectrum of Application Models



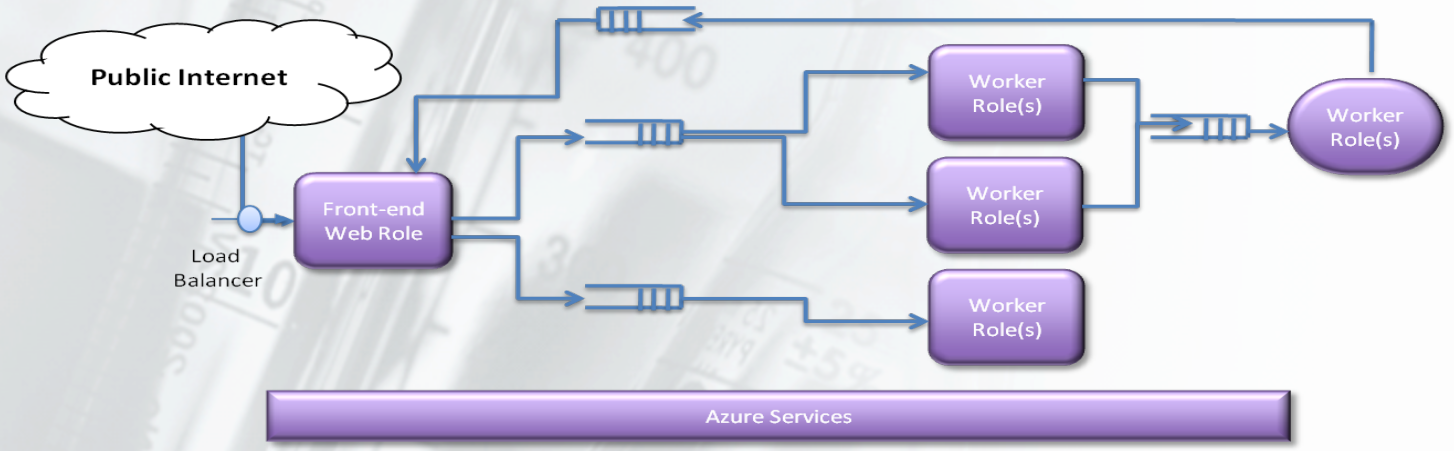
Windows Azure Compute and Storage Services

Compute Services

- **Web Roles** provide web service access to the app by the users. Web roles generate tasks for worker roles
- **Worker Roles** do “heavy lifting” and manage data in tables/blobs
- Communication is through **queues**.
- Roles are replicated as needed to provide massive scalability.

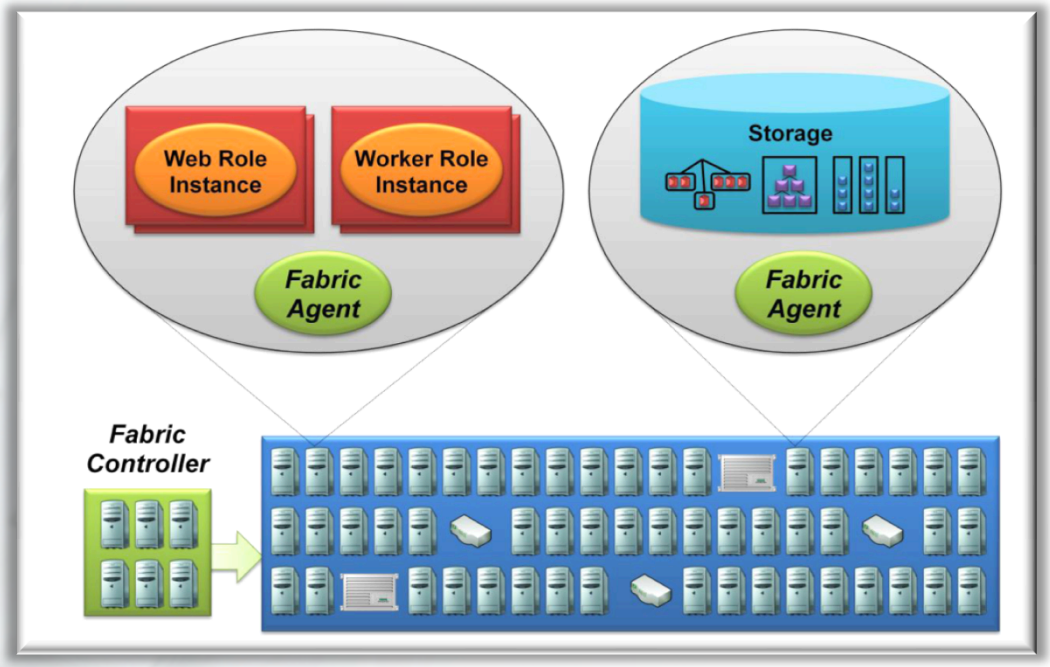
Storage Services

- **Blobs:** large, unstructured data (blobs in containers)
- **Tables:** massive amounts of simply structured data (entity with properties)
- **Queues**: messages or requests, allowing web-roles and worker-roles to interact
- **Drives**: durable NTFS file system volumes sharable across instances
- **SQL Azure** for relational data



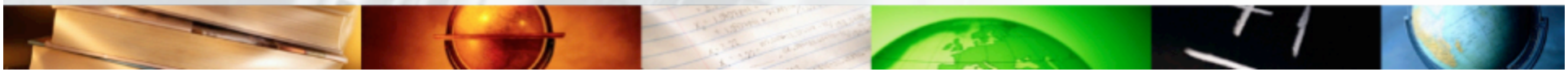
The Azure Fabric

- Consists of a (large) group of machines all managed by software called the **fabric controller**
- The fabric controller is replicated across a group of five to seven machines and owns all of the resources in the fabric
- Because it can communicate with a fabric agent on every computer, the controller is aware of every Windows Azure application running on the fabric
- It manages Windows Azure services and hosted applications (load balancing, backup, replication, failover, scale up/down...)



Characteristics of a Windows Azure Application

- Immediate
  - A credit card vs an e-mail to get started. No waiting.
- Persistent
  - Most applications run continuously
  - Other applications are just fine
- Scalable
  - Meet demands for compute and storage paying as you go
  - Scale according to needs
    - Thousands concurrent users served by one application.
    - One application using thousands of cores.



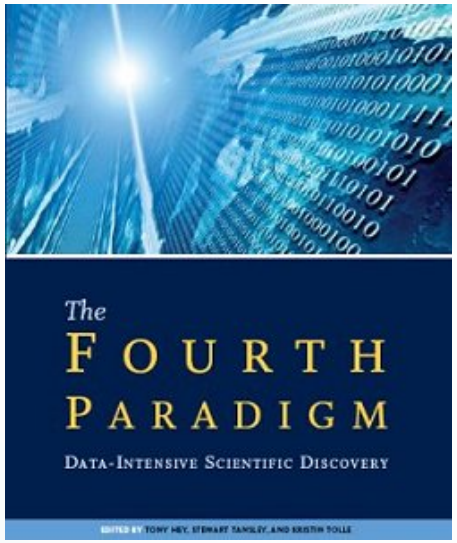
Emergence of a Fourth Research Paradigm

1. Thousand years ago – **Experimental Science**
   - Description of natural phenomena

2. Last few hundred years – **Theoretical Science**
   - Newton's Laws, Maxwell's Equations…

3. Last few decades – **Computational Science**
   - Simulation of complex phenomena

4. Today – **Data-Intensive Science**
   - Scientists overwhelmed with data sets from many different sources
     - Data captured by instruments
     - Data generated by simulations
     - Data generated by sensor networks
   - ➢ eScience is the set of tools and technologies to support data federation and collaboration
     - For analysis and data mining
     - For data visualization and exploration
     - For scholarly communication and dissemination

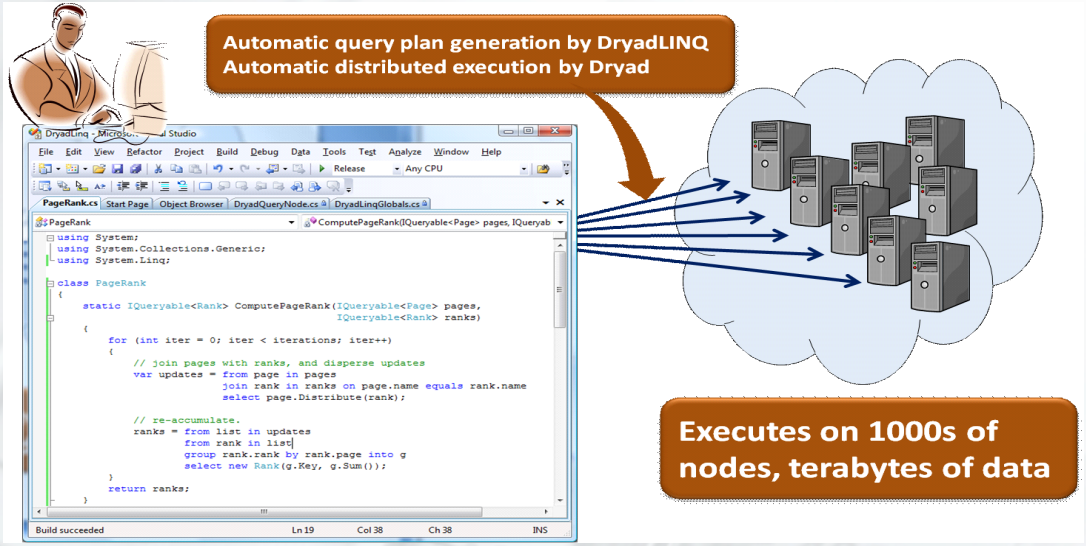


Science must move from data to information to knowledge

With thanks to Tony Hey
With thanks to Jim Gray

Dryad and DryadLINQ

Research programming models for writing distributed data-parallel applications that scale from a small cluster to a large data-center.



Microsoft Research Silicon Valley

A DryadLINQ programmer can use thousands of machines, each of them with multiple processors or cores, without prior knowledge in parallel programming.

Download available at http://connect.microsoft.com/DryadLINQ

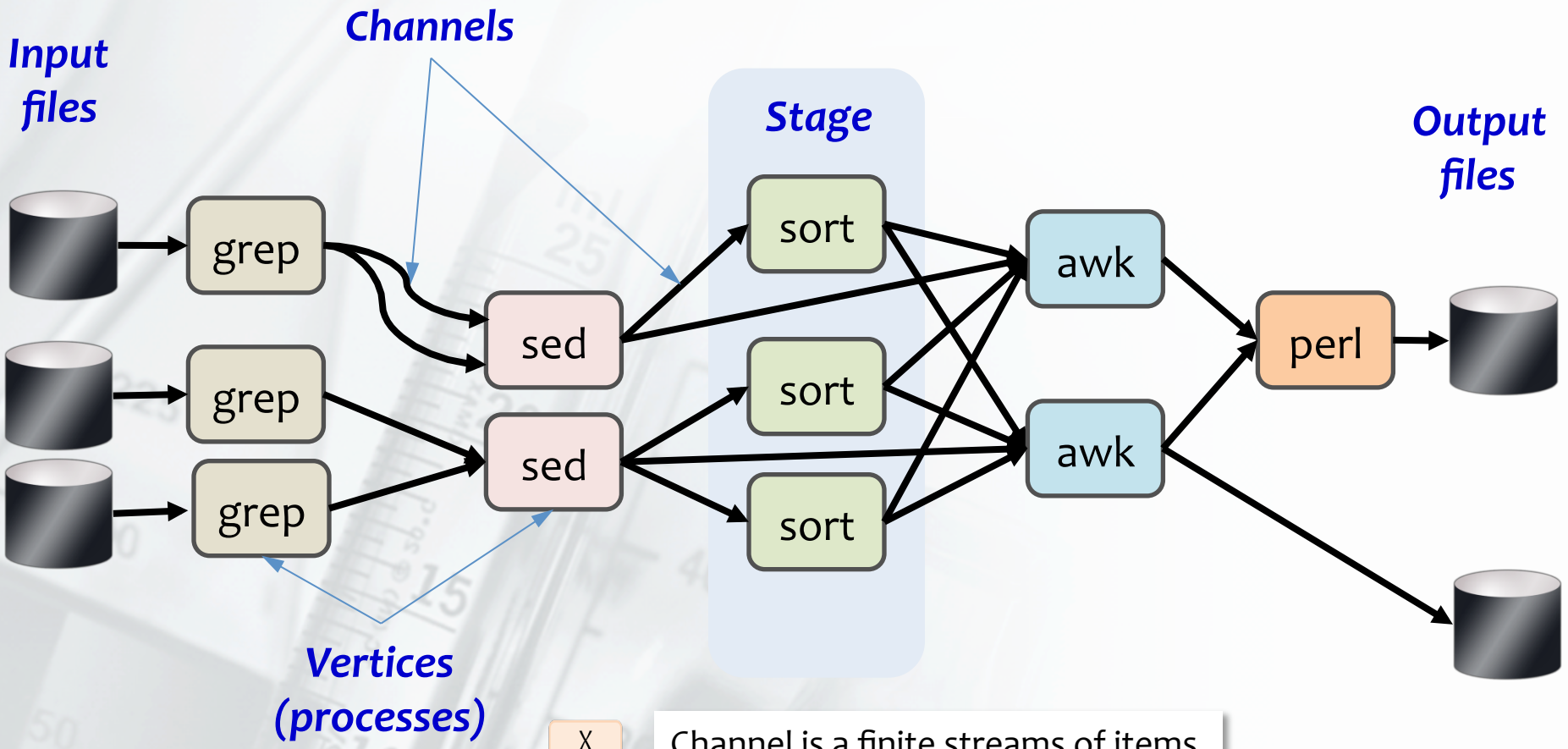
Windows HPC Server 2008

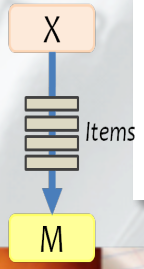
Dryad

- Continuously deployed since 2006
- Running on >> 10^4 machines
- Sifting through > 10Pb data daily
- Runs on clusters > 3000 machines
- Handles jobs with > 10^5 processes each
- Platform for rich software ecosystem
- Used by >> 100 developers

- Written at Microsoft Research, Silicon Valley

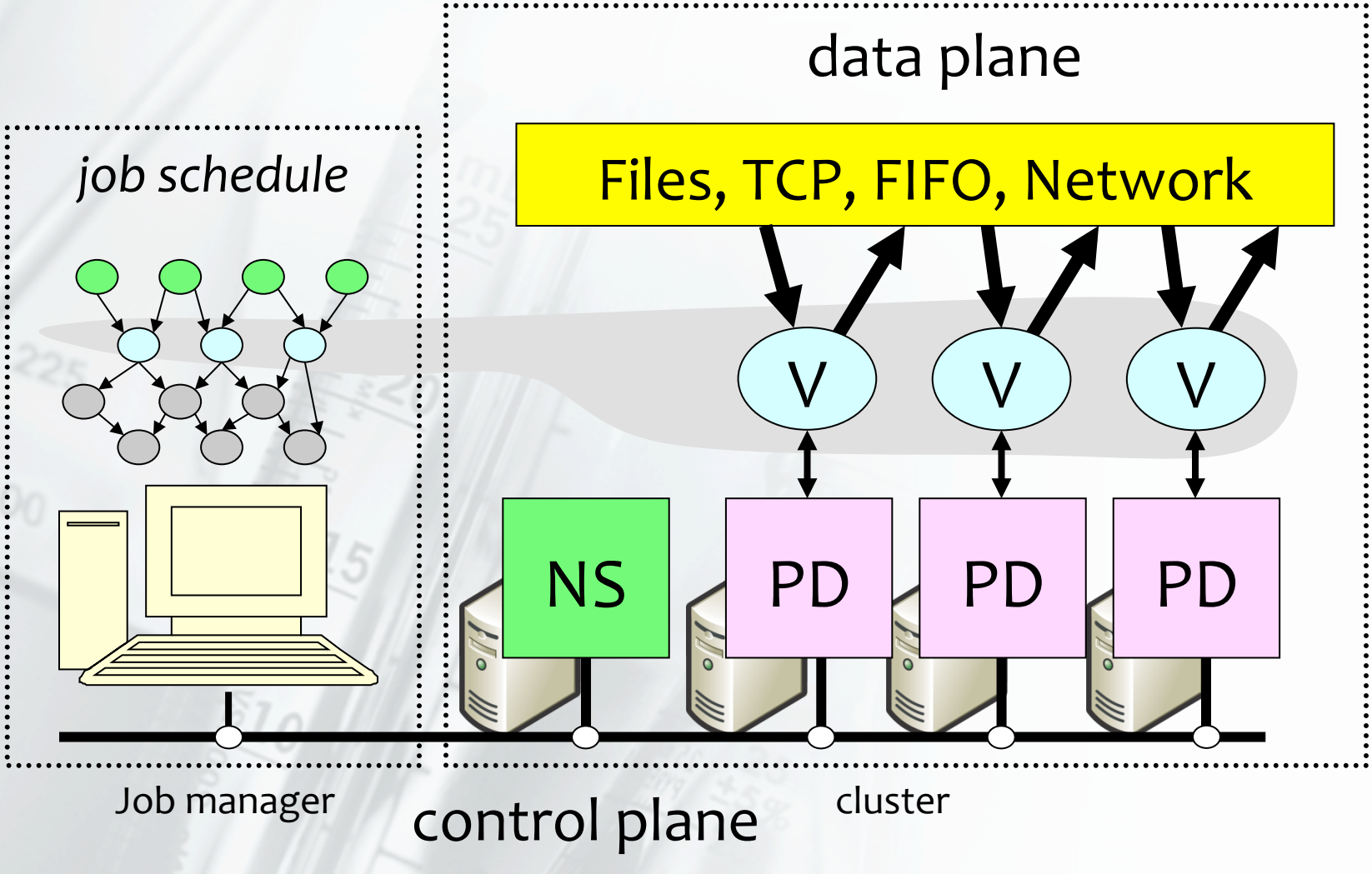
Dryad Job Structure





Channel is a finite streams of items
- NTFS files (temporary)
- TCP pipes (inter-machine)
- Memory FIFOs (intra-machine)

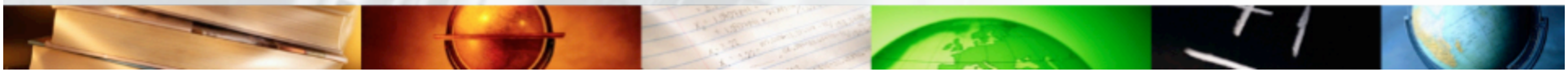
Dryad System Architecture



Dryad Properties

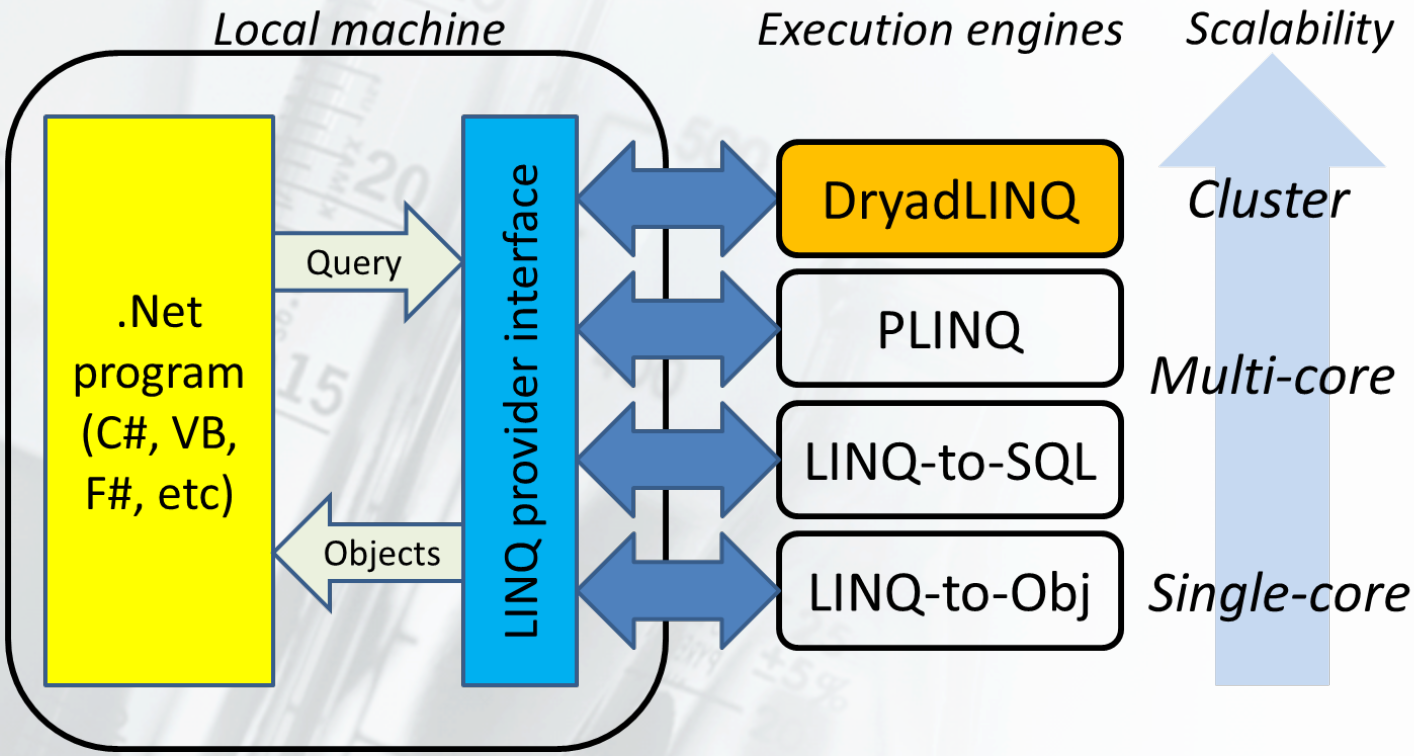
- Provides a general, clean execution layer
  - Dataflow graph as the computation model
  - Higher language layer supplies graph, vertex code, channel types, hints for data locality, …
- Automatically handles execution
  - Distributes code, routes data
  - Schedules processes on machines near data
  - Masks failures in cluster and network

But programming Dryad is not easy



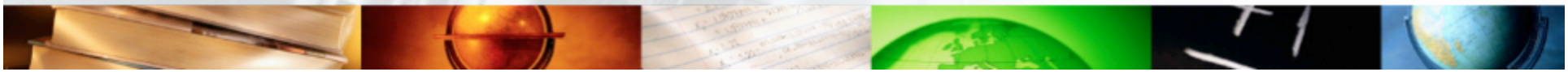
DryadLINQ Experience

- Sequential, single machine programming abstraction
- Same program runs on single-core, multi-core, or cluster
- Familiar programming languages
- Familiar development environment



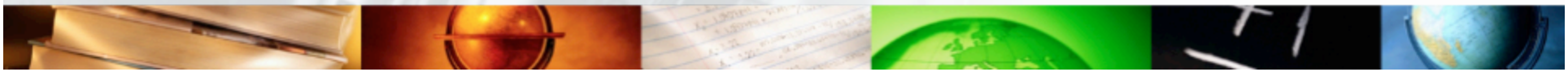
LINQ

- Microsoft's Language INtegrated Query
  - Released with .NET Framework 3.5, Visual Studio optional
- A set of operators to manipulate datasets in .NET
  - Support traditional relational operators
    - Select, Join, GroupBy, Aggregate, etc.
  - Integrated into .NET programming languages
    - Programs can call operators
    - Operators can invoke arbitrary .NET functions
- Data model
  - Data elements are strongly typed .NET objects
  - Much more expressive than SQL tables
- Extremely extensible
  - Add new custom operators
  - Add new execution providers

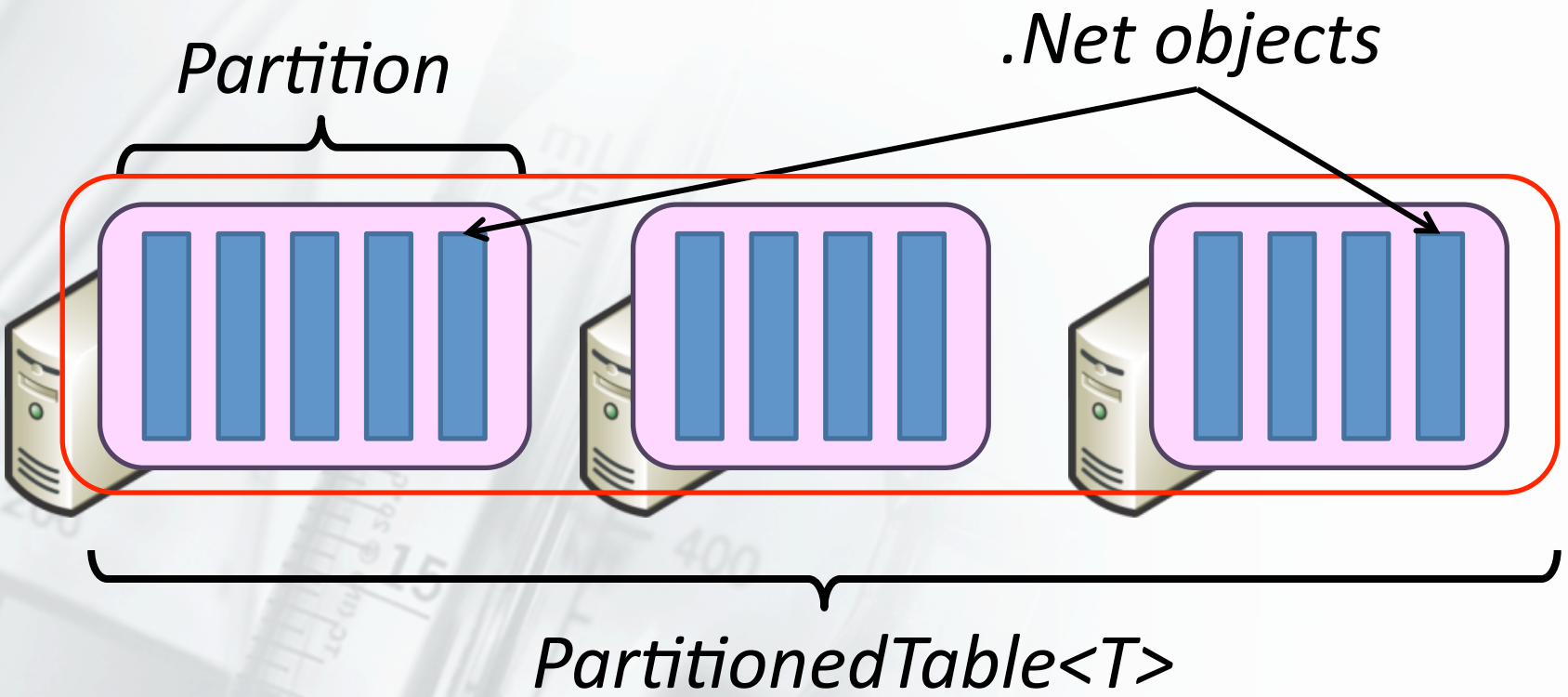


Example of LINQ Query

```
IEnumerable<string> logs = GetLogLines();
var logentries =
      from line in logs
      where !line.StartsWith("#")
      select new LogEntry(line);
var user =
      from access in logentries
      where access.user.EndsWith(@"\ulfar")
      select access;
var accesses =
      from access in user
      group access by access.page into pages
      select new UserPageCount("ulfar", pages.Key, pages.Count());
var htmAccesses =
      from access in accesses
      where access.page.EndsWith(".htm")
      orderby access.count descending
      select access;
```

DryadLINQ Data Model

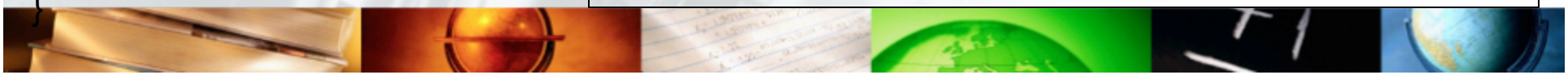


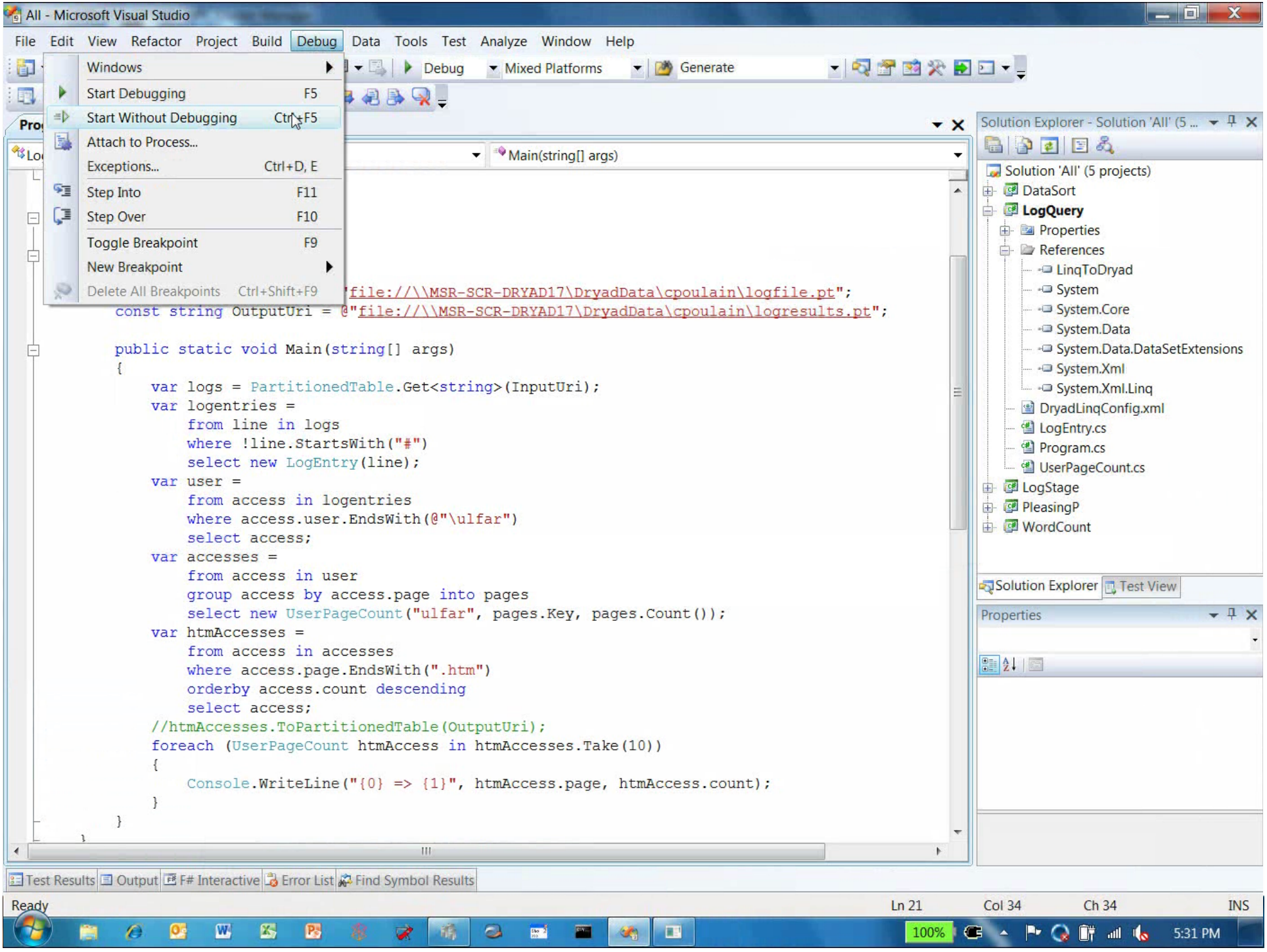
PartitionedTable<T> implements IQueryable<T> and IEnumerable<T>

A complete DryadLINQ program

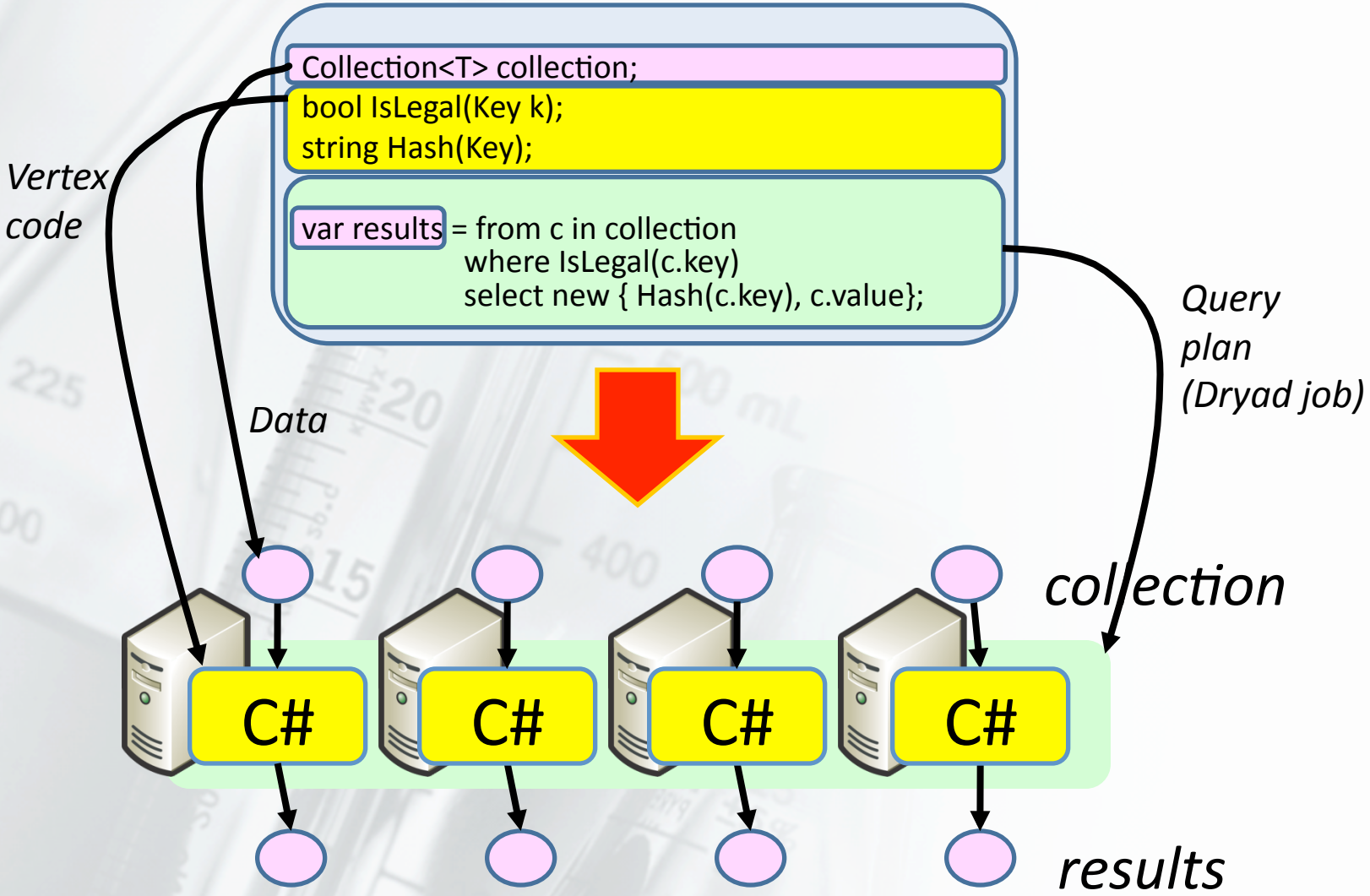
```
public class LogEntry {
   public string user;
   public string ip;
   public string page;
   public LogEntry(string line) {
      string[] fields = line.Split(' ');
      this.user = fields[8];
      this.ip = fields[9];
      this.page = fields[5];
   }
}

public class UserPageCount{
   public string user;
   public string page;
   public int count;
   public UserPageCount(
      string user, string page, int
count) {
      this.user = user;
      this.page = page;
      this.count = count;
   }
}
```

```
PartitionedTable<string> logs = PartitionedTable.Get<string>(
   @"file:\\MSR-SCR-DRYAD01\DryadData\cpoulain\logfile.pt"
);
var logentries =
   from line in logs
   where !line.StartsWith("#")
   select new LogEntry(line);
var user =
   from access in logentries
   where access.user.EndsWith(@"\ulfar")
   select access;
var accesses =
   from access in user
   group access by access.page into pages
   select new UserPageCount("ulfar", pages.Key, pages.Count());
var htmAccesses =
   from access in accesses
   where access.page.EndsWith(".htm")
   orderby access.count descending
   select access;
htmAccesses.ToPartitionedTable(
   @"file:\\MSR-SCR-DRYAD01\DryadData\cpoulain\results.pt"
);
```




All - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Analyze Window Help
Windows
Start Debugging F5
Start Without Debugging Ctrl+F5
Attach to Process...
Exceptions... Ctrl+D, E
Step Into F11
Step Over F10
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Debug
Mixed Platforms
Generate
Main(string[] args)
const string InputUri = @"file://\\MSR-SCR-DRYAD17\DryadData\cpoulain\logfile.pt";
const string OutputUri = @"file://\\MSR-SCR-DRYAD17\DryadData\cpoulain\logresults.pt";
public static void Main(string[] args)
{
var logs = PartitionedTable.Get<string>(InputUri);
var logentries =
from line in logs
where !line.StartsWith("#")
select new LogEntry(line);
var user =
from access in logentries
where access.user.EndsWith(@"\ulfar")
select access;
var accesses =
from access in user
group access by access.page into pages
select new UserPageCount("ulfar", pages.Key, pages.Count());
var htmAccesses =
from access in accesses
where access.page.EndsWith(".htm")
orderby access.count descending
select access;
//htmAccesses.ToPartitionedTable(OutputUri);
foreach (UserPageCount htmAccess in htmAccesses.Take(10))
{
Console.WriteLine("{0} => {1}", htmAccess.page, htmAccess.count);
}
}
Solution Explorer - Solution 'All' (5 ...
Solution 'All' (5 projects)
DataSort
LogQuery
Properties
References
LinqToDryad
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Xml
System.Xml.Linq
DryadLinqConfig.xml
LogEntry.cs
Program.cs
UserPageCount.cs
LogStage
PleasingP
WordCount
Solution Explorer
Test View
Properties
Test Results
Output
F# Interactive
Error List
Find Symbol Results
Ready
Ln 21
Col 34
Ch 34
INS
100%
5:31 PM

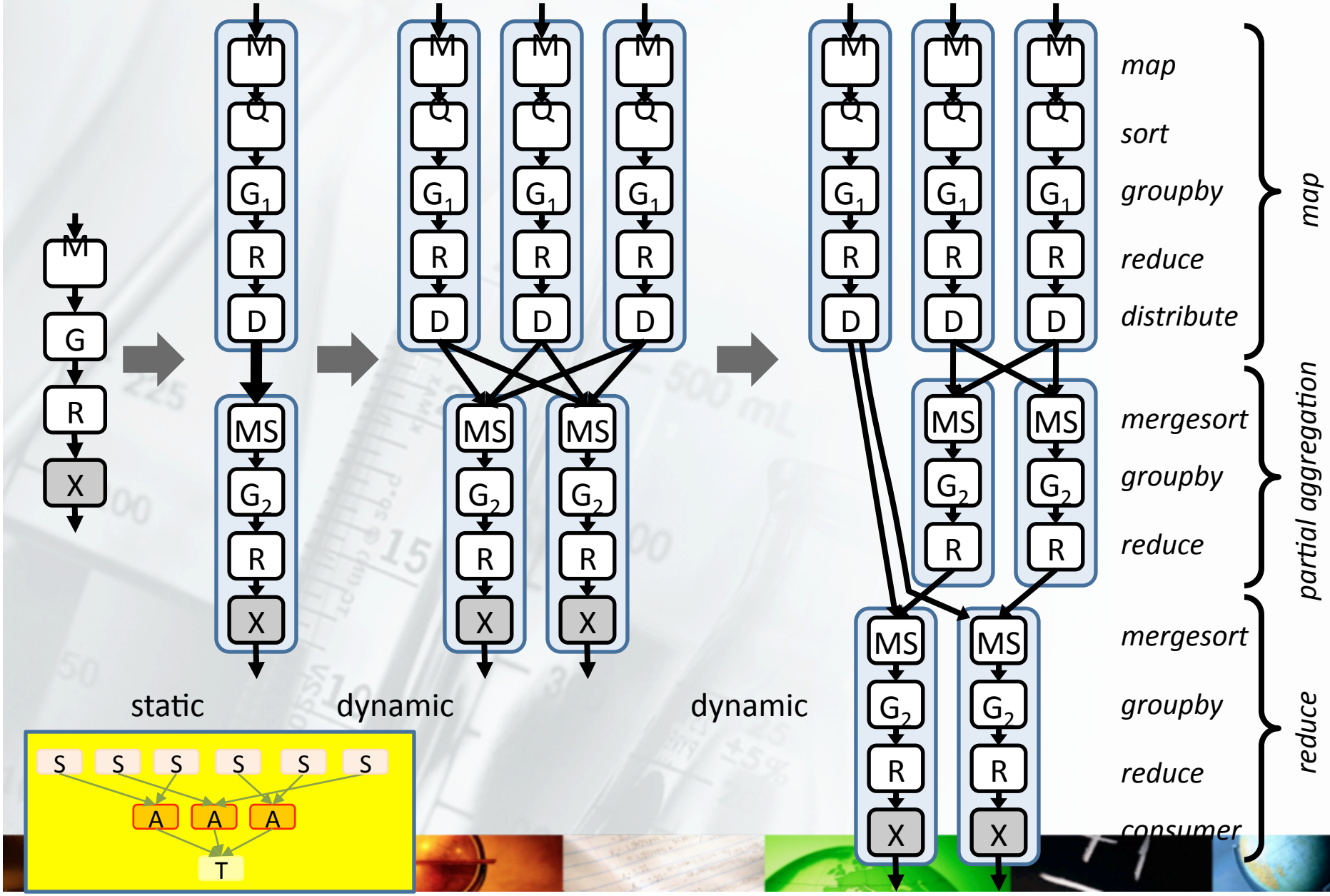
DryadLINQ Provider



Map-Reduce in DryadLINQ

```
public static IQueryable<S> MapReduce<T,M,K,S>(
    this IQueryable<T> input,
            Func<T, IEnumerable<M>> mapper,
            Func<M,K> keySelector,
            Func<IGrouping<K,M>,S> reducer)
{
    var map = input.SelectMany(mapper);
    var group = map.GroupBy(keySelector);
    var result = group.Select(reducer);
    return result;
}
```

Map-Reduce Plan

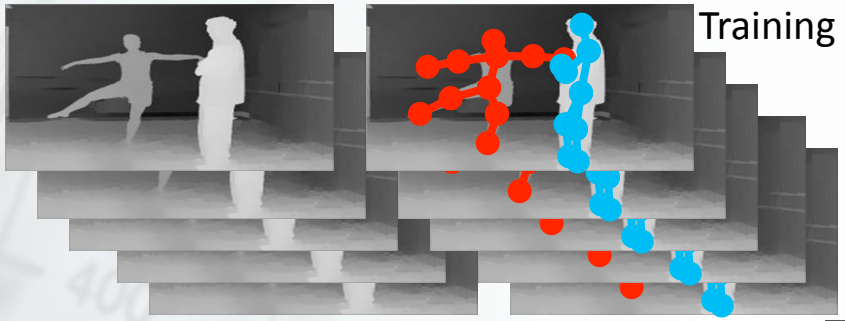


Project Natal: Learning From Data

Project Natal is using DryadLINQ to train enormous decision trees from millions of images across hundreds of cores.

Motion Capture
(ground truth)

Rasterize



Training examples

Machine learning

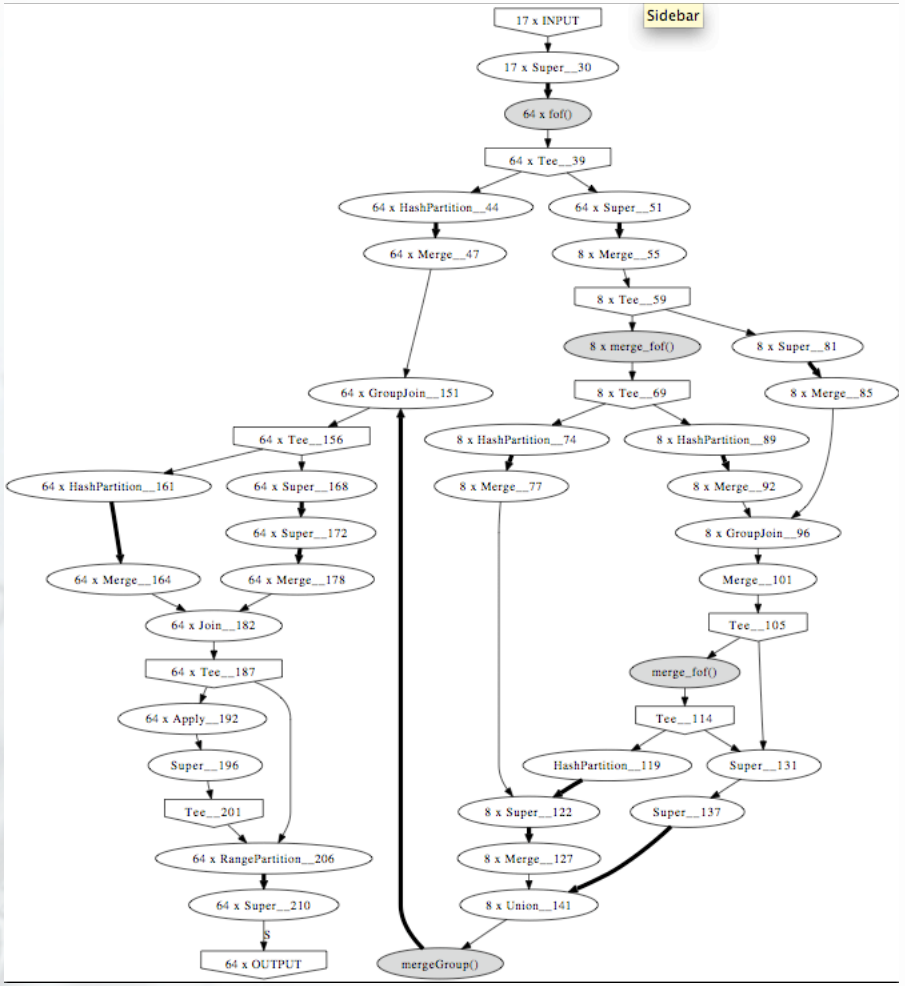
Classifier

Recognize players from depth map at frame rate using fraction of Xbox CPU.

Scalable clustering algorithm for N-body simulations in a shared-nothing cluster

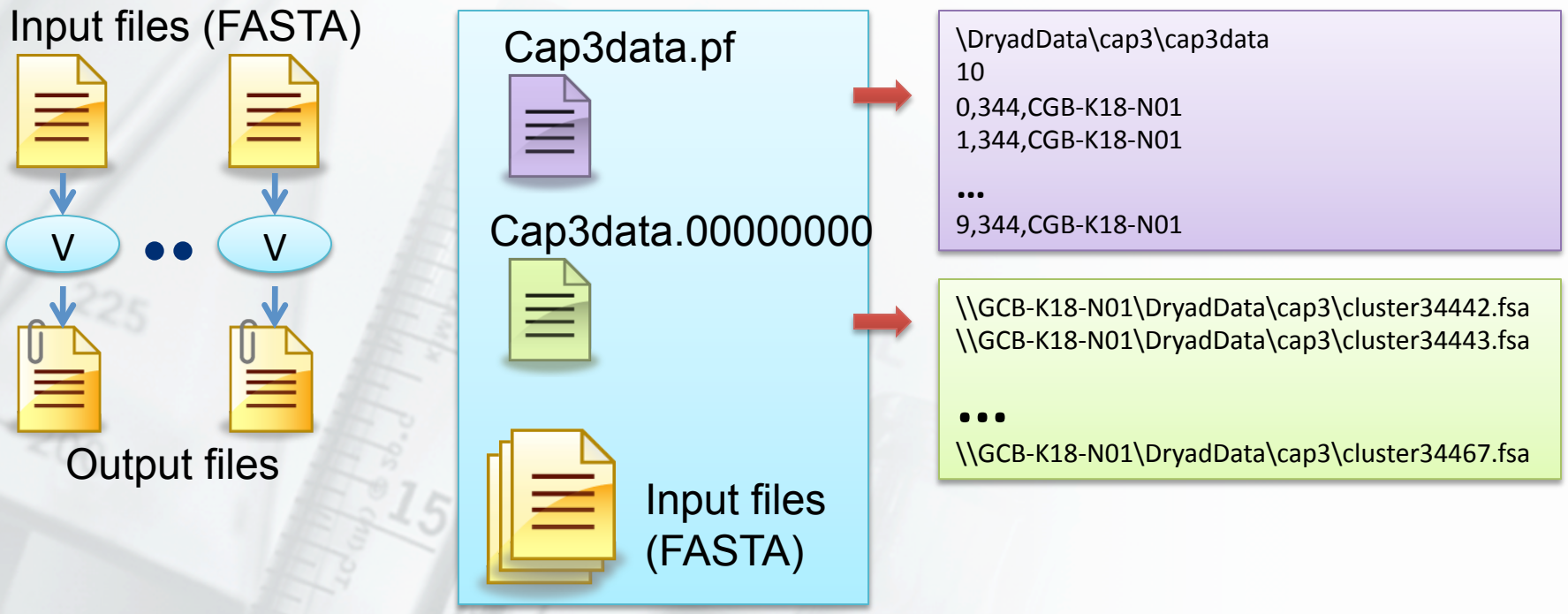
YongChul Kwon, Dylan Nunley, Jeffrey P. Gardner, Magdalena Balazinska, Bill Howe, and Sarah Loebman. UW Tech Report. UW-CSE-09-06-01. June 2009.

- Large-scale spatial clustering
  - 916M particles in 3D clustered under 70 minutes with 8 nodes.
- Re-implemented using DryadLINQ
  - Partition > Local cluster > Merge cluster > Relabel
  - Faster development and good scalability



CAP3 - DNA Sequence Assembly Program [1]

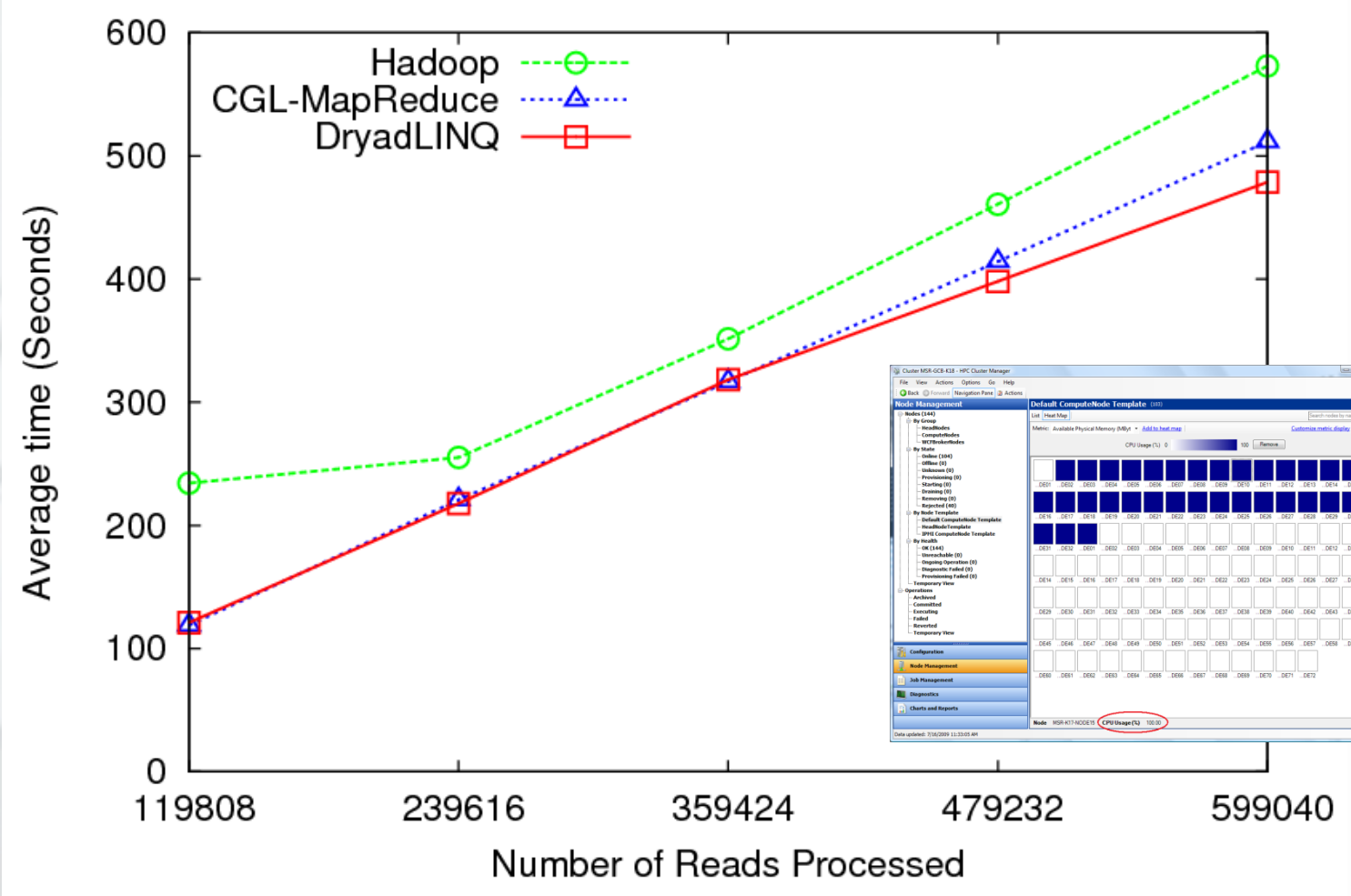
EST (Expressed Sequence Tag) corresponds to messenger RNAs (mRNAs) transcribed from the genes residing on chromosomes. Each individual EST sequence represents a fragment of mRNA, and the EST assembly aims to re-construct full-length mRNA sequences for each expressed gene.




```
IQueryable<LineRecord> inputFiles=PartitionedTable.Get<LineRecord>(uri);
IQueryable<OutputInfo> = inputFiles.Select(x=>ExecuteCAP3(x.line));
```

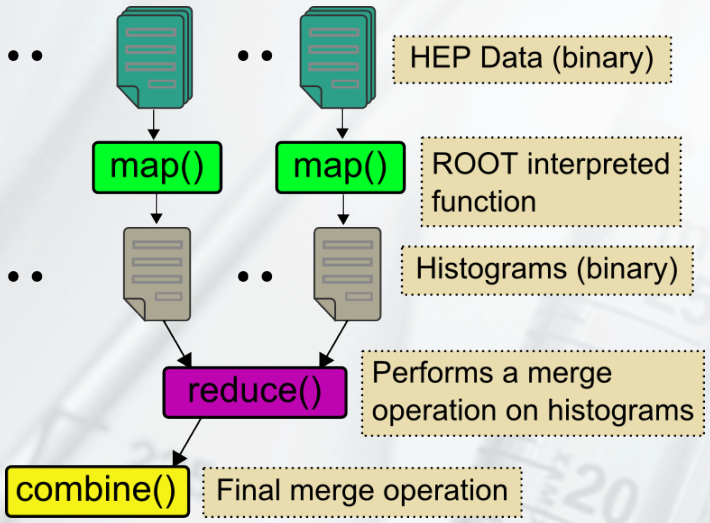
[1] X. Huang, A. Madan, “CAP3: A DNA Sequence Assembly Program,” Genome Research, vol. 9, no. 9, 1999.

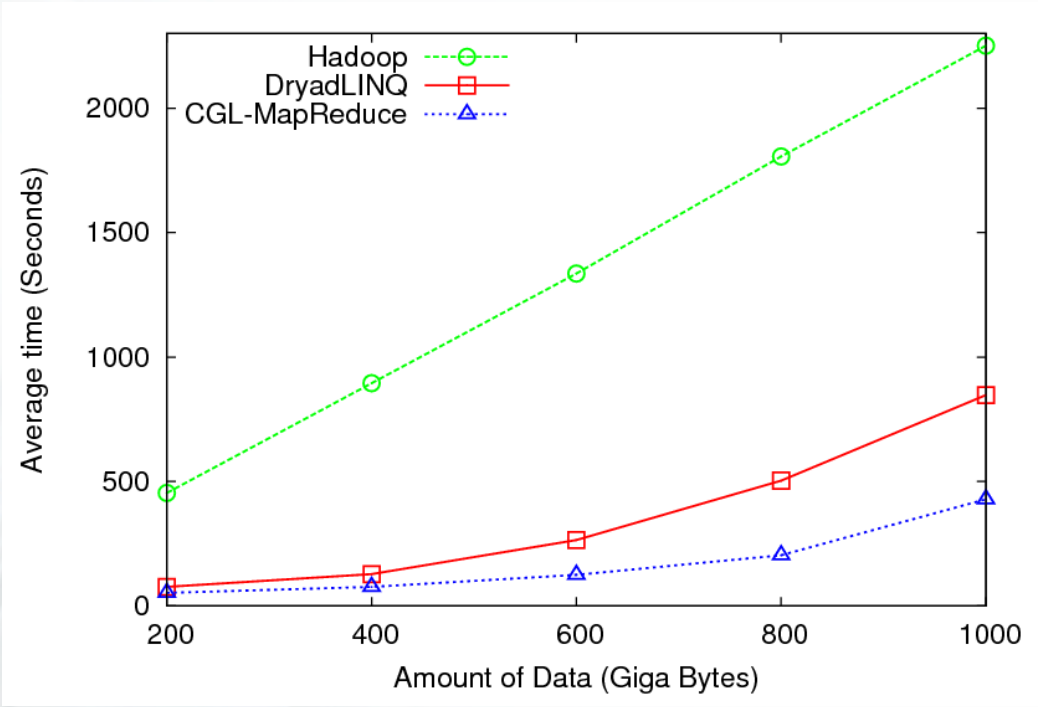
CAP3 - Performance



"DryadLINQ for Scientific Analyses", Jaliya Ekanayake, Thilina Gunarathnea, Geoffrey Fox, Atilla Soner Balkir, Christophe Poulain, Nelson Araujo, Roger Barga (IEEE eScience '09)

High Energy Physics Data Analysis



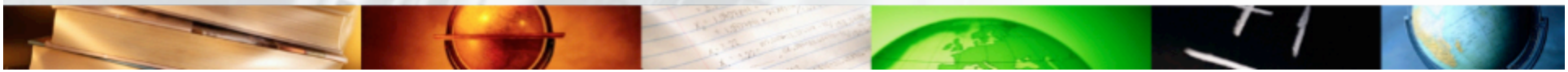


- Histogramming of events from a large (up to 1TB) data set
- Data analysis requires ROOT framework (ROOT Interpreted Scripts)
- Performance depends on disk access speeds
- Hadoop implementation uses a shared parallel file system (Lustre)
  - ROOT scripts cannot access data from HDFS
  - On demand data movement has significant overhead
- Dryad stores data in local disks giving better performance over Hadoop

Parting Thought #1

- Windows Azure is Microsoft's cloud platform
  http://www.microsoft.com/windowsazure/

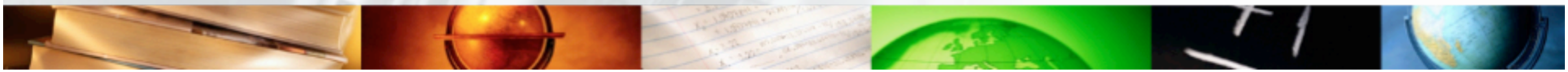
- To Condor, this looks like one more resource to harness



Parting Thought #2

DryadLINQ provides a powerful, elegant programming environment for large-scale data-parallel computing.

Is this of interest to the Condor community?



Acknowledgements

MSR Silicon Valley Dryad & DryadLINQ teams

Andrew Birrell, Mihai Budiu, Jon Currey, Ulfar Erlingsson, Dennis Fetterly, Michael Isard, Pradeep Kunda, Mark Manasse, Chandu Thekkath and Yuan Yu .

http://research.microsoft.com/en-us/projects/dryad
http://research.microsoft.com/en-us/projects/dryadlinq

MSR External Research

Advanced Research Tools and Services Team
http://research.microsoft.com/en-us/collaboration/tools/dryad.aspx

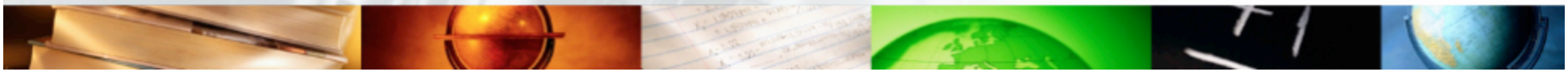
MS Product Groups: HPC, Parallel Computing Platform.

Academic Collaborators

Jaliya Ekanayake, Geoffrey Fox, Thilina Gunarathne, Scott Beason, Xiaohong Qiu (Indiana University Bloomington).

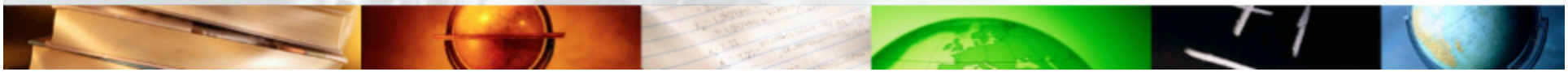
YongChul Kwon, Magdalena Balazinska (University of Washington).

Atilla Soner Balkir, Ian Foster (University of Chicago).



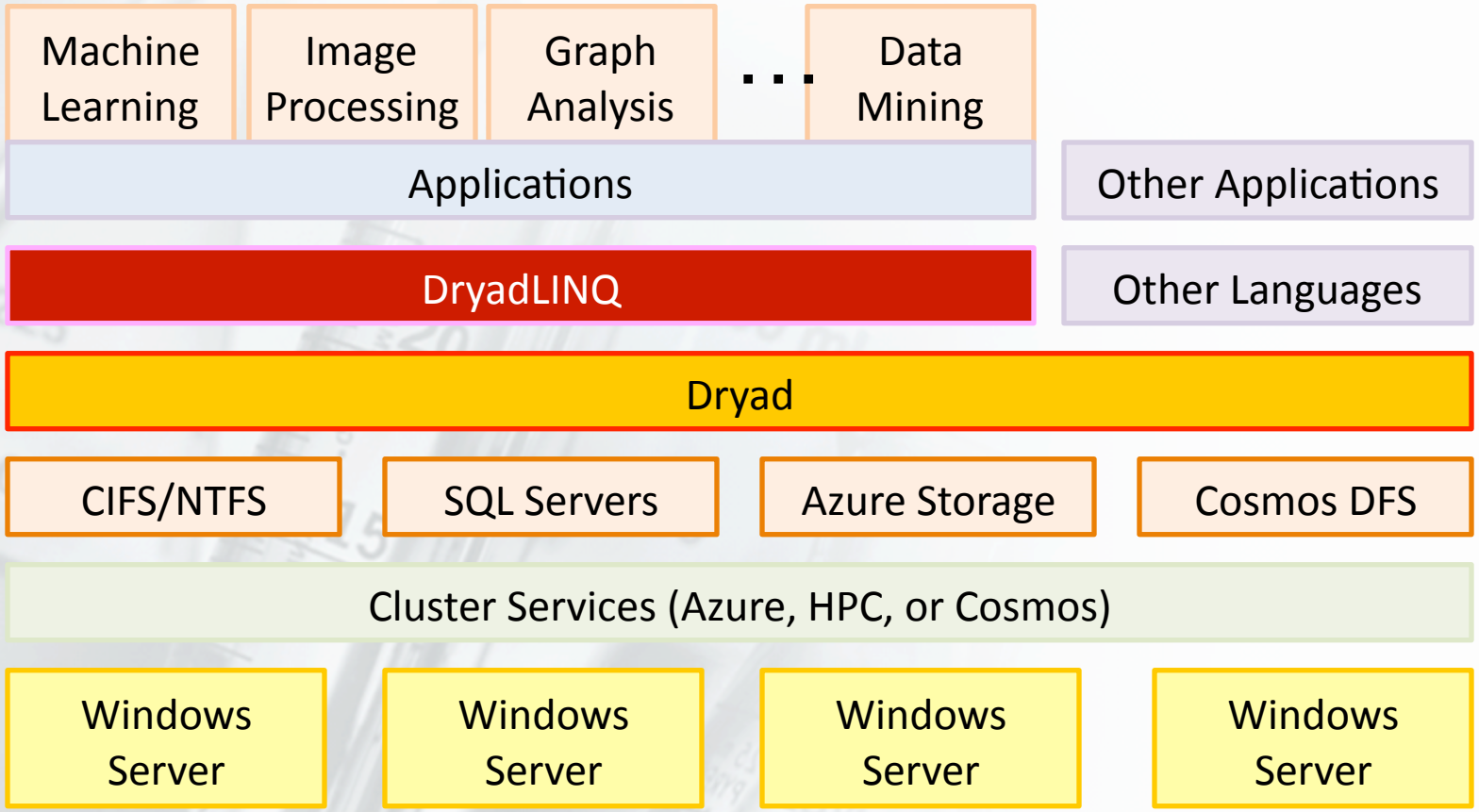
Dryad/DryadLINQ Papers

1. Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks (EuroSys'07)
2. DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language (OSDI'08)
3. Hunting for prolems with Artemis (Usenix WASL, San Diego 08)
4. Distributed Data-Parallel Computing Using a High-Level Programming Language (SIGMOD'09)
5. Quincy: Fair scheduling for distributed computing clusters (SOSP'09)
6. Distributed Aggregation for Data-Parallel Computing: Interfaces and Implementations (SOSP'09)
7. DryadInc: Reusing work in large scale computation (HotCloud 09).

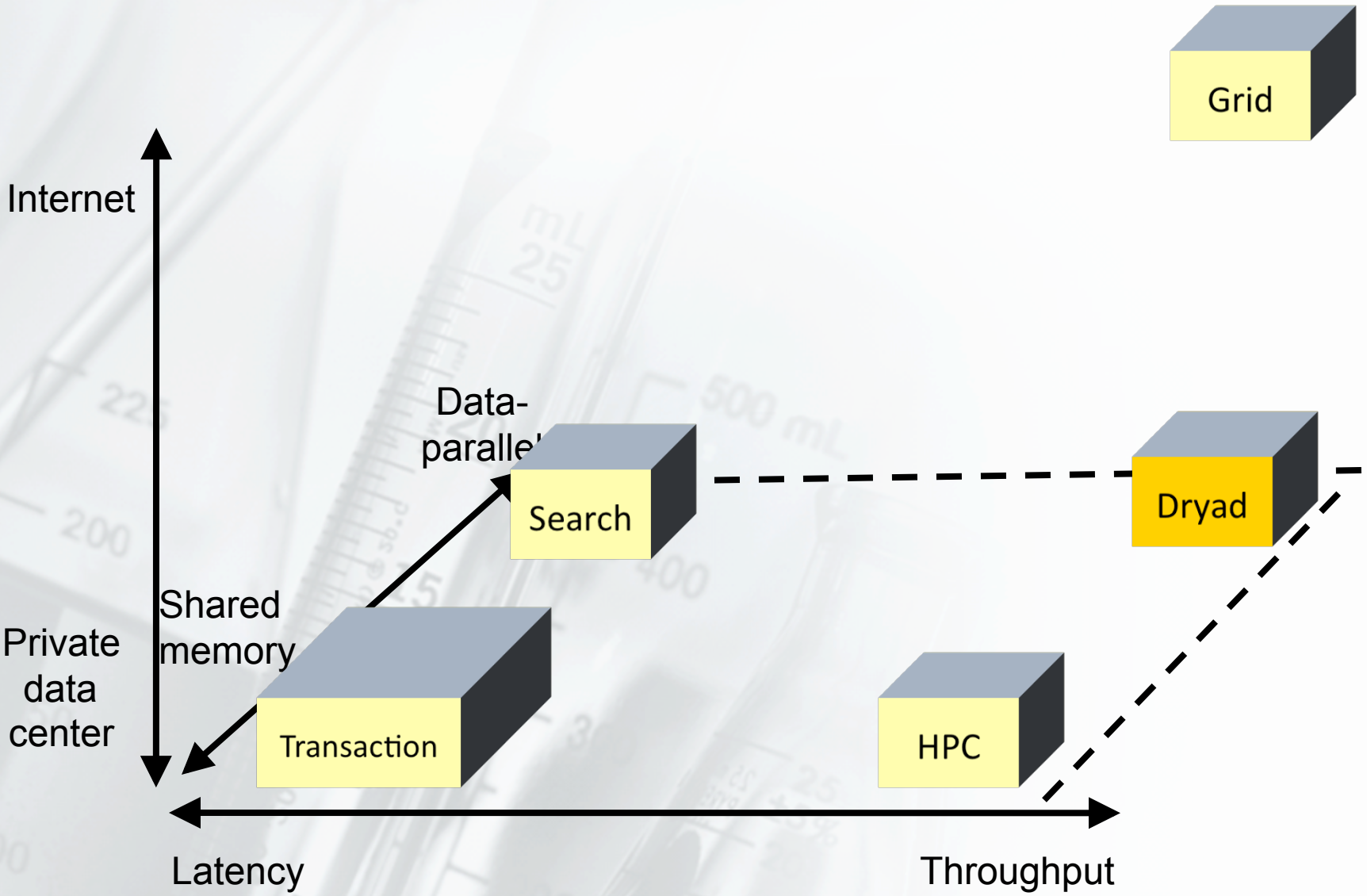


Extra Slides

Software Stack



Design Space



Debugging and Diagnostics