Distributed Data Parallel Computing: The Sector Perspective on Big Data

Robert Grossman

July 25, 2010

Laboratory for Advanced Computing
University of Illinois at Chicago

LABORATORY FOR ADVANCED COMPUTING

Open Data Group

open data

Institute for Genomics & Systems Biology
University of Chicago

Institute for Genomics & Systems Biology

Part 1.

Open Cloud Testbed

CENIC

C-Wave

Dragon

MREN

- 9 racks
- 250+ Nodes
- 1000+ Cores
- 10+ Gb/s

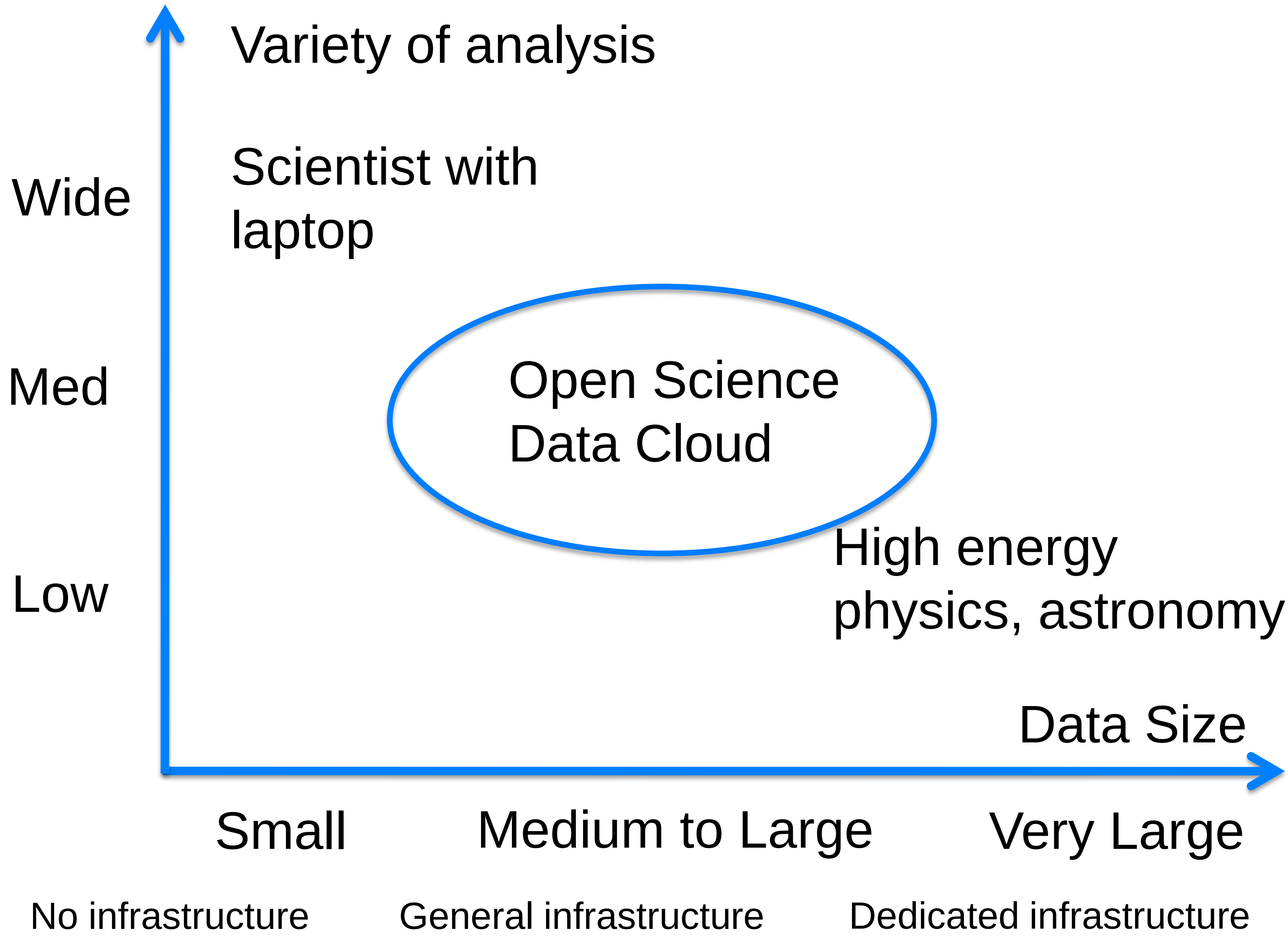
- Hadoop
- Sector/Sphere
- Thrift
- KVM VMs
- Nova
- Eucalyptus VMs

Open Science Data Cloud

NSF OSDC PIRE Project – Working with 5 international partners (all connected with 10 Gbps networks).

sky cloud

Bionimbus (biology & health care)


Variety of analysis
Scientist with laptop
Wide
Med
Low
Open Science Data Cloud
High energy physics, astronomy
Data Size
Small
Medium to Large
Very Large
No infrastructure
General infrastructure
Dedicated infrastructure

Part 2
What's Different About Data Center Computing?

Data center scale computing provides storage and computational resources at the scale and with the reliability of a data center.

The Datacenter as a Computer

An Introduction to the Design of Warehouse-Scale Machines

Luiz André Barroso and Urs Hölzle
Google Inc.

A very nice recent book by Barroso and Holzle

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE # 6

MORGAN & CLAYPOOL PUBLISHERS

Scale is new

Elastic, Usage Based Pricing Is New

costs the same as

1 computer in a rack
for 120 hours

120 computers in three
racks for 1 hour

Simplicity of the Parallel Programming Framework is New

A new programmer can develop a program to proces a container full of data with less than day of training usin; MapReduce.

HPC

Goal: Minimize latency and control heat.

Large Data Clouds

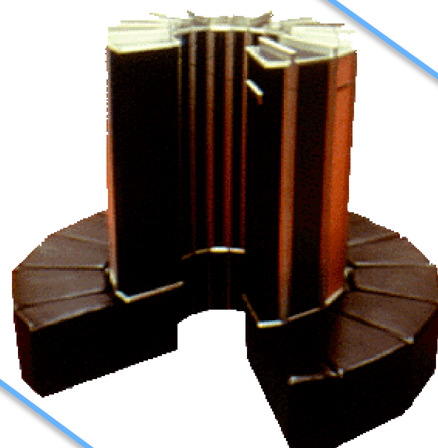
Goal: Maximize data (with matching compute) and control cost.

Elastic Clouds

Goal: Minimize cost of virtualized machines & provide on-demand.

2003
10x-100x

1976
10x-100x



data
science

1670
250x

simulation
science

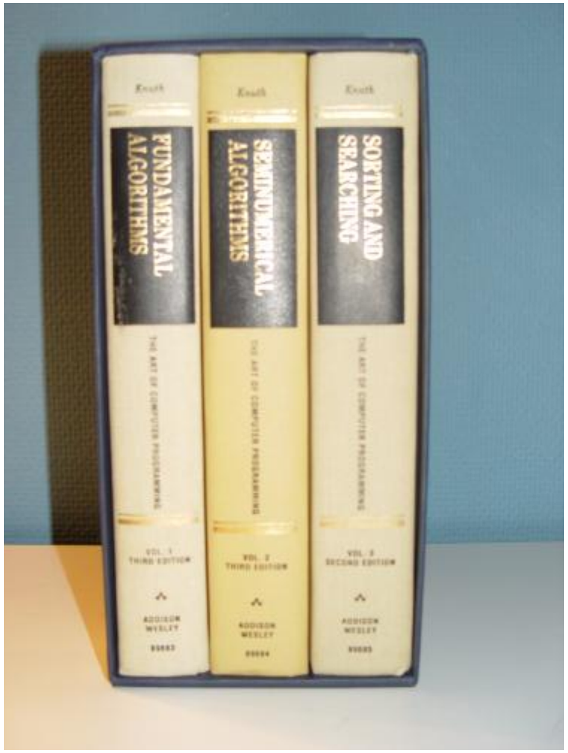
1609
30x

experimental
science

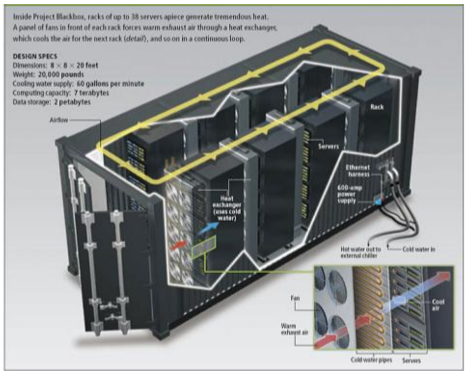
	Databases	**Data Clouds**
Scalability	100's TB	100's PB
Functionality	Full SQL-based queries, including joins	Single keys
Optimized	Databases optimized for safe writes	Data clouds optimized for efficient reads
Consistency model	ACID (Atomicity, Consistency, Isolation & Durability)	Eventual consistency
Parallelism	Difficult because of ACID model; shared nothing is possible	Parallelism over commodity components
Scale	Racks	Data center

	Grids	**Clouds**
Problem	Too few cycles	Too many users & too much data
Infrastructure	Clusters and supercomputers	Data centers
Architecture	Federated Virtual Organization	Hosted Organization
Programming Model	Powerful, but difficult to use	Not as powerful, but easy to use

Part 3
How Do You Program A Data Center?



How Do You Build A Data Center?



- Containers used by Google, Microsoft & others
- Data center consists of 10-60+ containers.

Microsoft Data Center, Northlake, Illinois

What is the Operating System?

. . .

VM 1

VM 5

workstatio
n

. . .

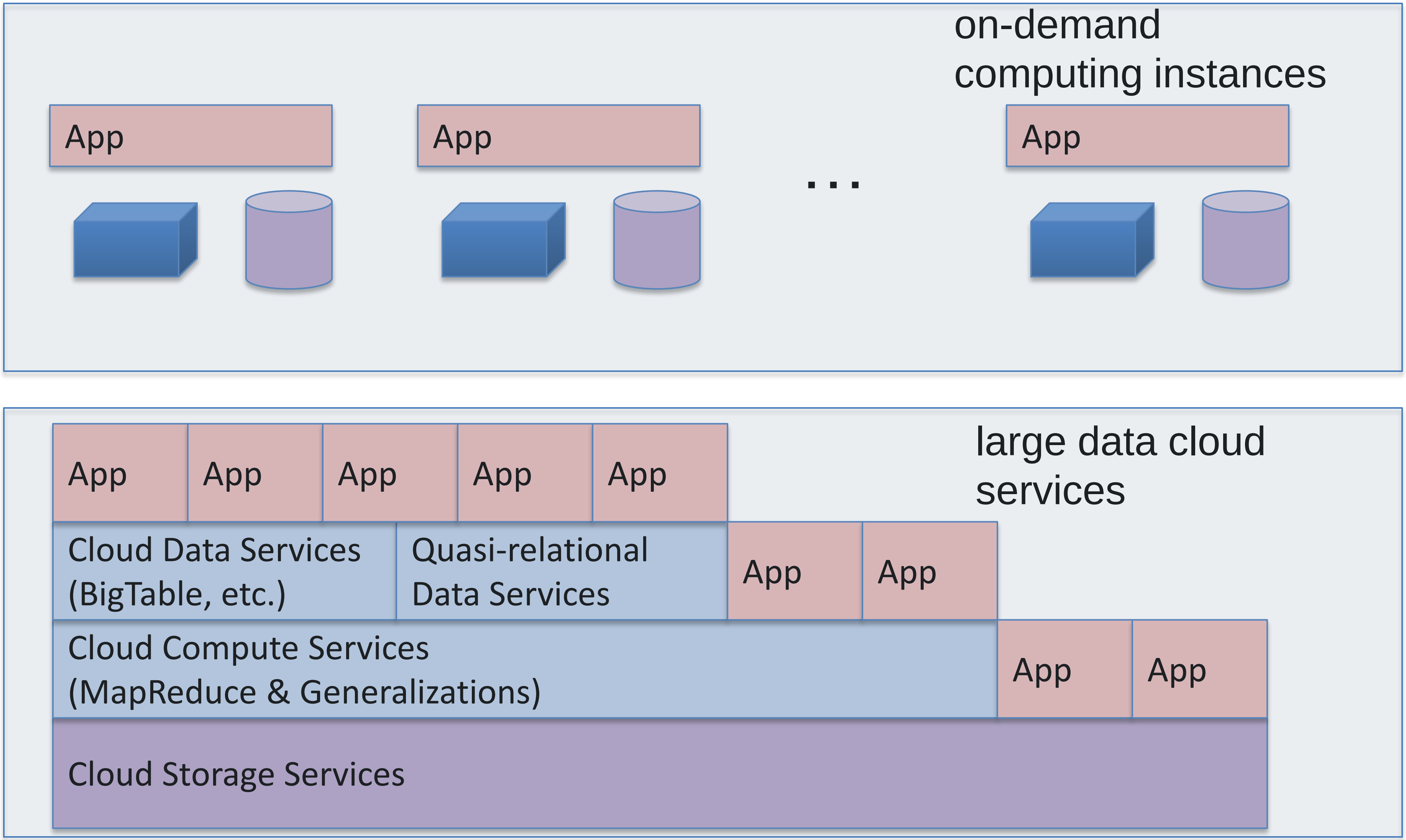
VM 1

VM 50,000

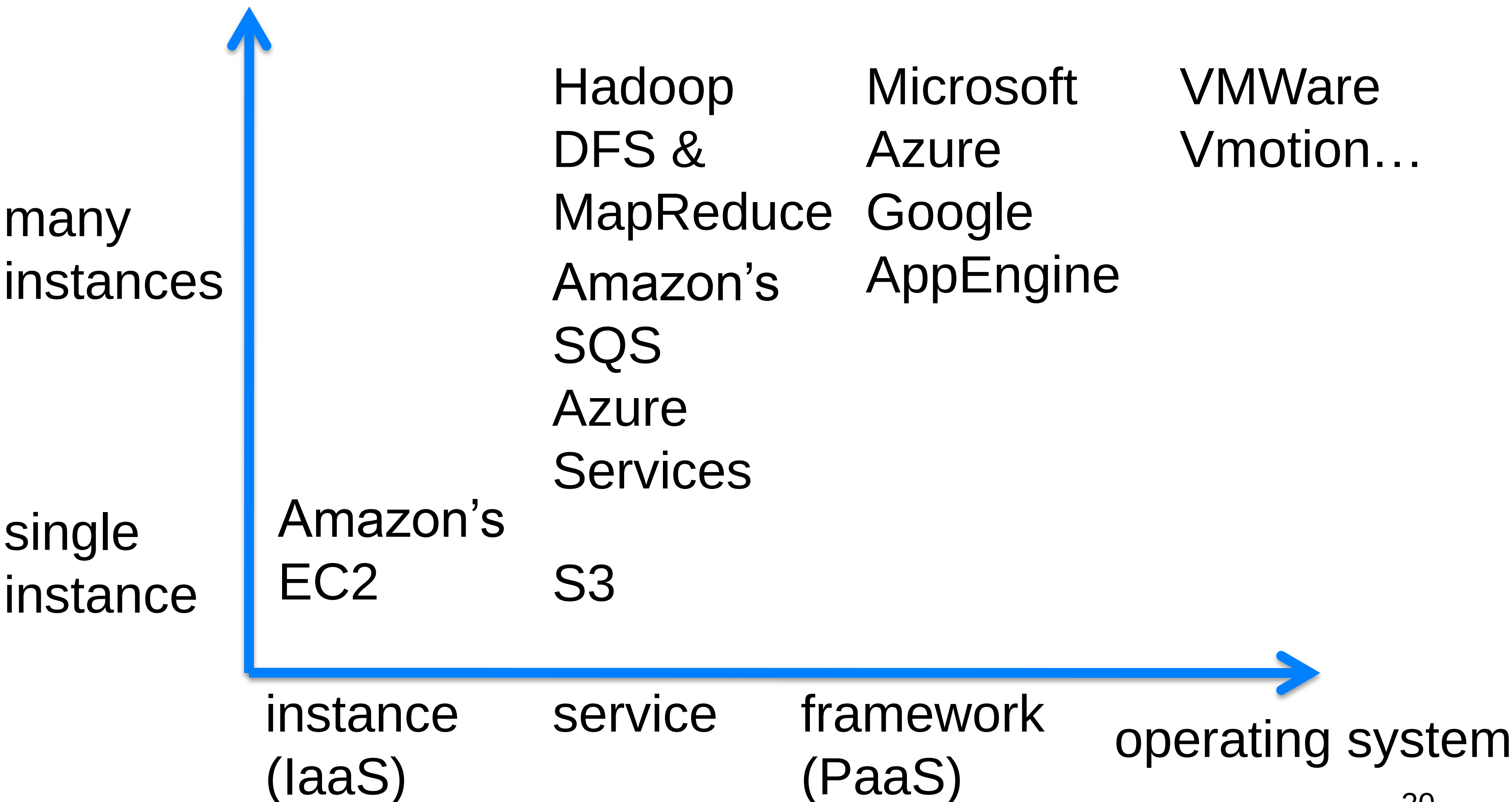
Data Center Operating System

- Data center services include: VM management services, VM fail over and restart, security services, power management services, etc.

Architectural Models: How Do You Fill a Data Center?



Instances, Services & Frameworks



Some Programming Models for Data Centers

- Operations over data center of disks
  - MapReduce (“string-based”)
  - Iterate MapReduce (Twister)
  - DryadLINQ
  - User-Defined Functions (UDFs) over data center
  - SQL and Quasi-SQL over data center
  - Data analysis / statistics functions over data center

More Programming Models

- Operations over data center of memory
  - Memcached (distributed in-memory key-value store)
  - Grep over distributed memory
  - UDFs over distributed memory
  - SQL and Quasi-SQL over distributed memory
  - Data analysis / statistics over distributed memory

Part 4. Stacks for Big Data

hadoop

HBASE

noSQL

Sector/Sphere

Cassandra

Thrift

Apache CouchDB

CLOUDSTORE

Project Voldemort
A distributed database.

The Google Data Stack

- The Google File System (2003)
- MapReduce: Simplified Data Processing… (2004)
- BigTable: A Distributed Storage System… (2006)

Map-Reduce Example

- Input is file with one document per record
- User specifies map function
  - key = document URL
  - Value = terms that document contains

("doc cdickens",
"it was the best of times")

map

"it", 1
"was", 1
"the", 1
"best", 1

Example (cont'd)

- MapReduce library gathers together all pairs with the same key value (shuffle/sort phase)
- The user-defined reduce function combines all the values associated with the same key

key = "it"
values = 1, 1

key = "was"
values = 1, 1

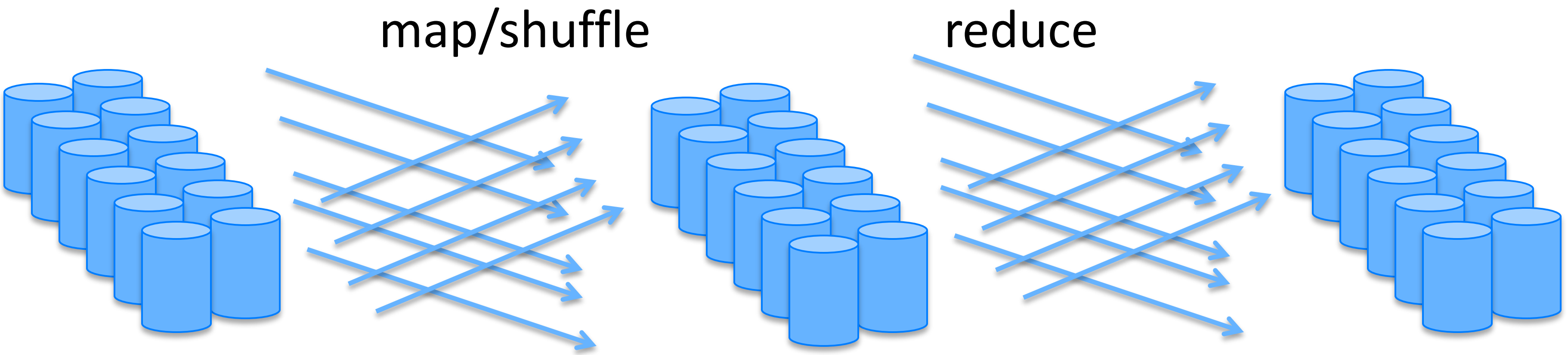
key = "best"
values = 1

key = "worst"
values = 1

reduce

"it", 2
"was", 2
"best", 1
"worst", 1

Applying MapReduce to the Data in Storage Cloud



Google's Large Data Cloud

Applications

Compute Services — Google's MapReduce

Data Services — Google's BigTable

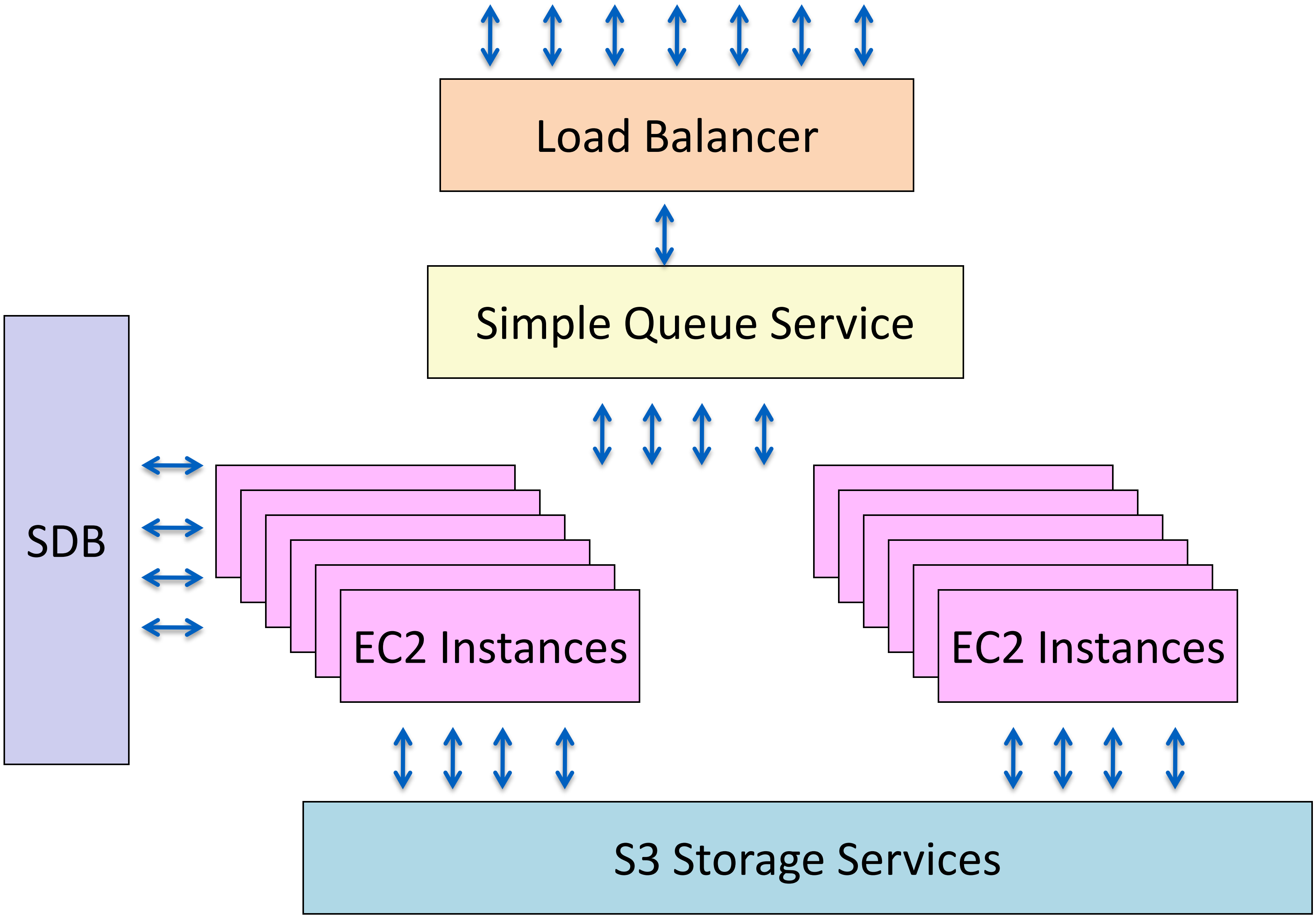
Storage Services — Google File System (GFS)

Google's Stack

Hadoop's Large Data Cloud

Hadoop's Stack	
Applications	
Compute Services	Hadoop's MapReduce
Data Services	NoSQL Databases
Storage Services	Hadoop Distributed File System (HDFS)

Amazon Style Data Cloud



Evolution of NoSQL Databases

- Standard architecture for simple web apps:
  - Front end load balanced web servers
  - Business logic layer in the middle
  - Backend database
- Databases do not scale well with very large numbers of users or very large amounts of data
- Alternatives include
  - Sharded (partitioned) databases
  - master-slave databases
  - memcached

NoSQL Systems

- Suggests No SQL support, also Not Only SQL
- One or more of the ACID properties not supported
- Joins generally not supported
- Usually flexible schemas
- Some well known examples: Google's BigTable, Amazon's S3 & Facebook's Cassandra
- Several recent open source systems

Different Types of NoSQL Systems

- Distributed Key-Value Systems
  - Amazon’s S3 Key-Value Store (Dynamo)
  - Voldemort
- Column-based Systems
  - BigTable
  - HBase
  - Cassandra
- Document-based systems
  - CouchDB

Cassandra vs MySQL Comparison

- MySQL > 50 GB Data
  Writes Average : ~300 ms
  Reads Average : ~350 ms
- Cassandra > 50 GB Data
  Writes Average : 0.12 ms
  Reads Average : 15 ms

Source: Avinash Lakshman, Prashant Malik, Cassandra Structured Storage System over a P2P Network, static.last.fm/johan/nosql-20090611/cassandra_nosql.pdf

CAP Theorem

- Proposed by Eric Brewer, 2000
- Three properties of a system: consistency, availability and partitions
- You can have at most two of these three properties for any shared-data system
- Scale out requires partitions
- Most large web-based systems choose availability over consistency

Reference: Brewer, PODC 2000; Gilbert/Lynch, SIGACT News 2002

Eventual Consistency

- All updates eventually propagate through the system and all nodes will eventually be consistent (assuming no more updates)
- Eventually, a node is either updated or removed from service.
- Can be implemented with Gossip protocol
- Amazon’s Dynamo popularized this approach
- Sometimes this is called BASE (**B**asically **A**vailable, **S**oft state, **E**ventual consistency), as opposed to ACID

Part 5. Sector Architecture

Design Objectives

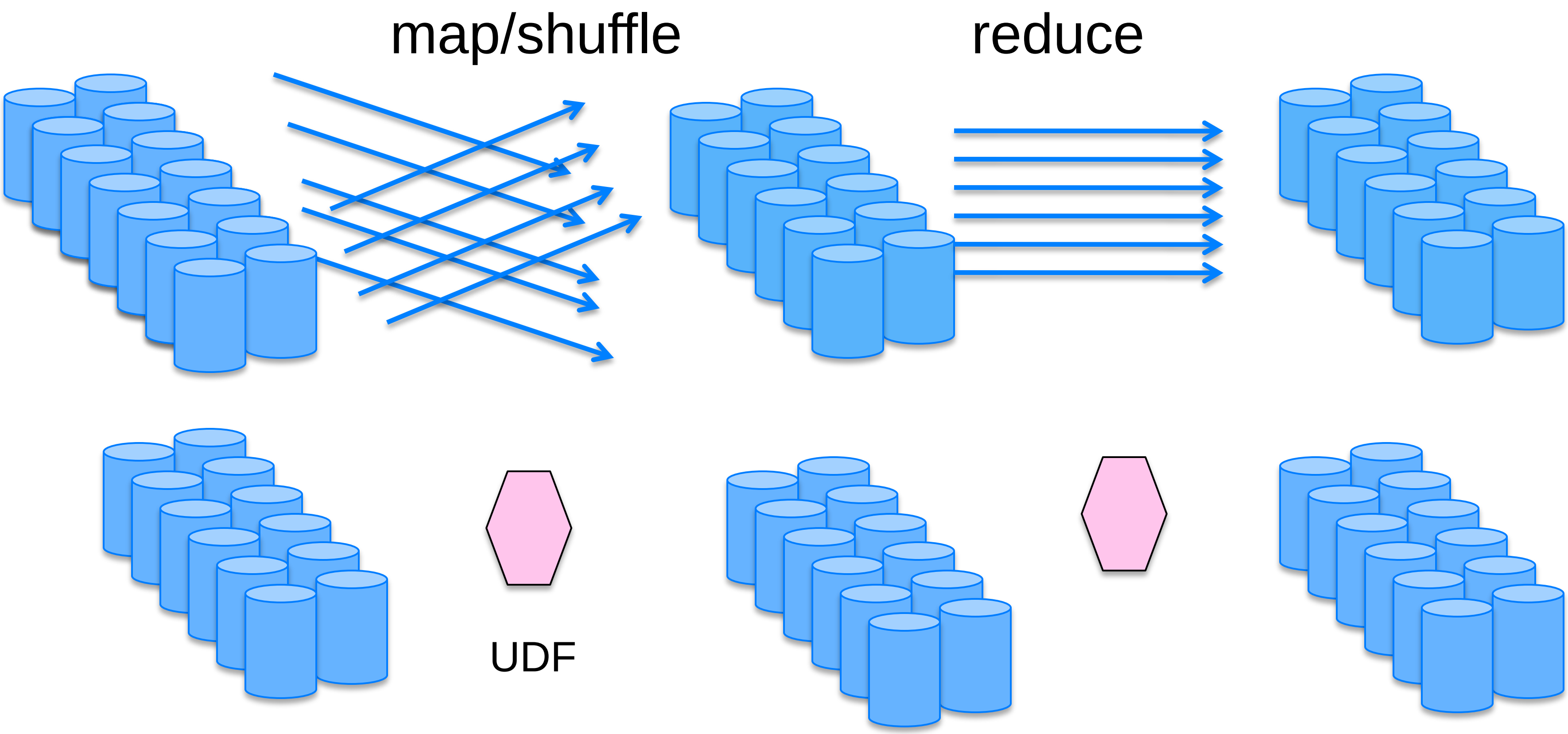
1. Provide Internet scale data storage for large data
   - Support multiple data centers connected by high speed wide networks
2. Simplify data intensive computing for a larger class of problems than covered by MapReduce
   - Support applying User Defined Functions to the data managed by a storage cloud, with transparent load balancing and fault tolerance

Sector's Large Data Cloud

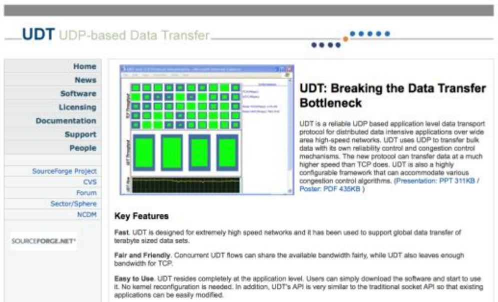
Sector's Stack	
Applications	
Compute Services	Sphere's UDFs
Data Services	
Storage Services	Sector's Distributed File System (SDFS)
Routing & Transport Services	UDP-based Data Transport Protocol (UDT)

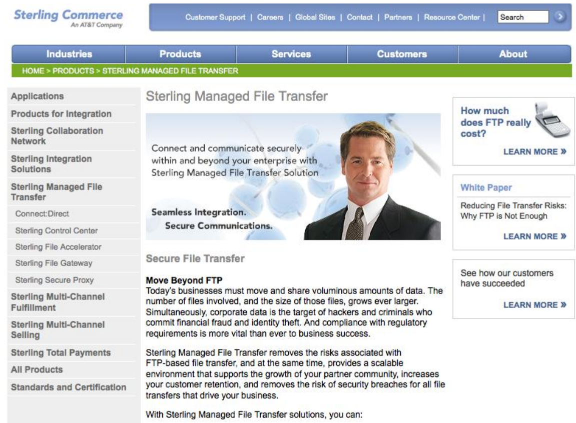
Sector's Stack

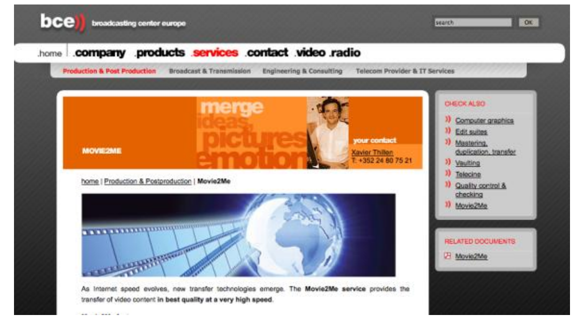
Apply User Defined Functions (UDF) to Files in Storage Cloud



UDT



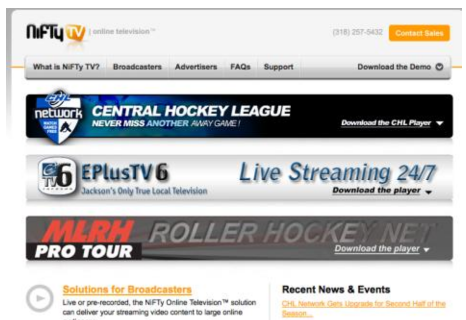




udt.sourceforge.net

Sterling Commerce

Movie2Me



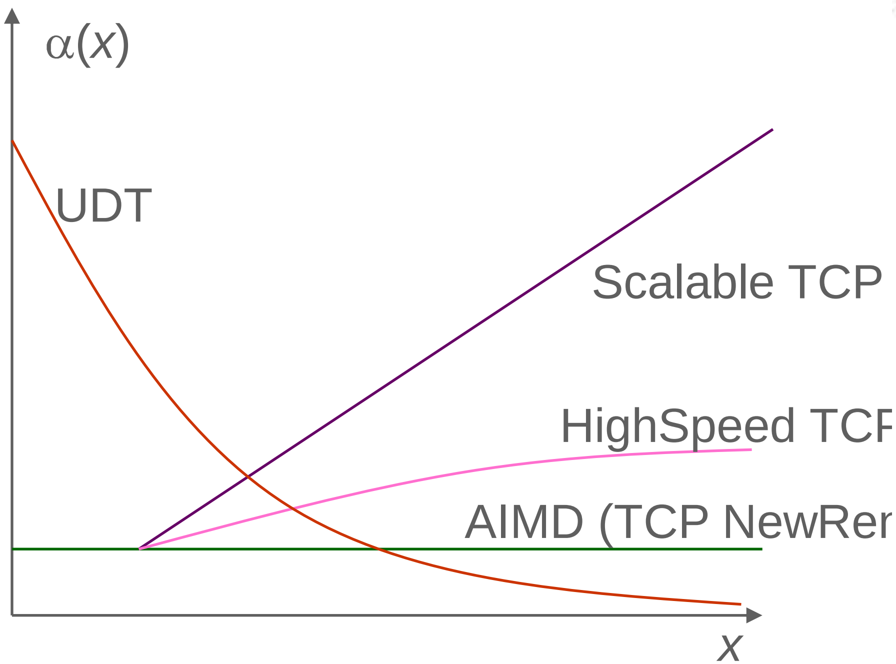
Globus

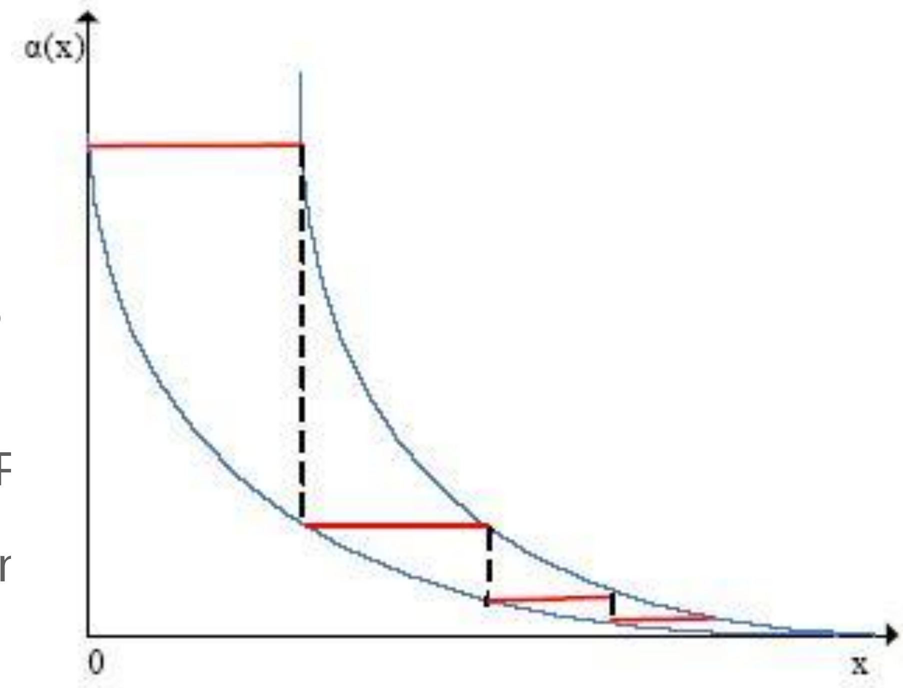
Nifty TV

Power Folder

UDT has been downloaded 25,000+ times

Alternatives to TCP – Decreasing Increases AIMD Protocols

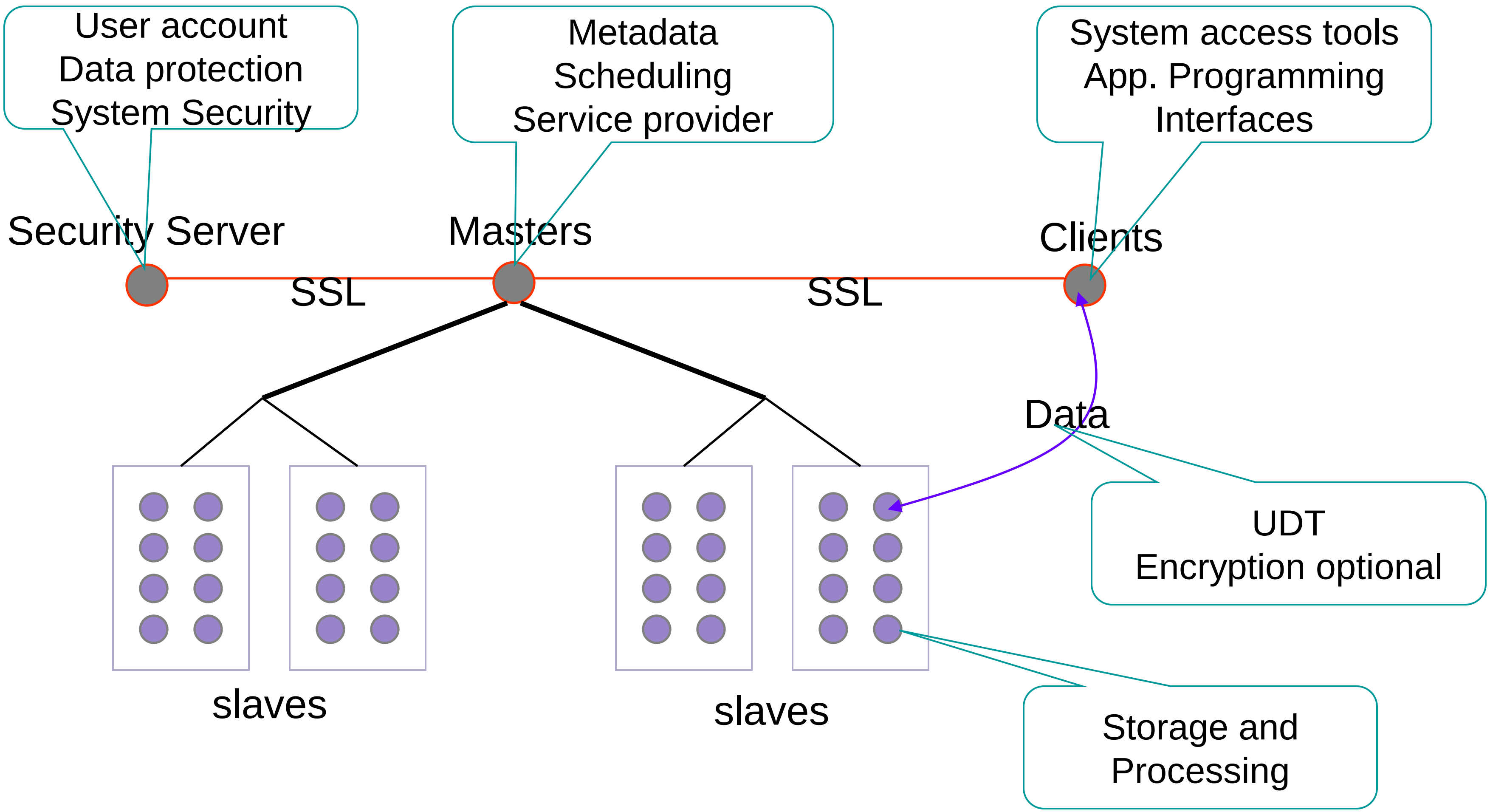




$$x \leftarrow x + \alpha(x)$$ increase of packet sending rate x

$$x \leftarrow (1-\beta)\,x$$ decrease factor

System Architecture



	Hadoop DFS	Sector DFS
Storage Cloud	Block-based file system	File-based
Programming Model	MapReduce	UDF & MapReduce
Protocol	TCP	UDP-based protocol (UDT)
Replication	At write	At write or period.
Security	Not yet	HIPAA capable
Language	Java	C++

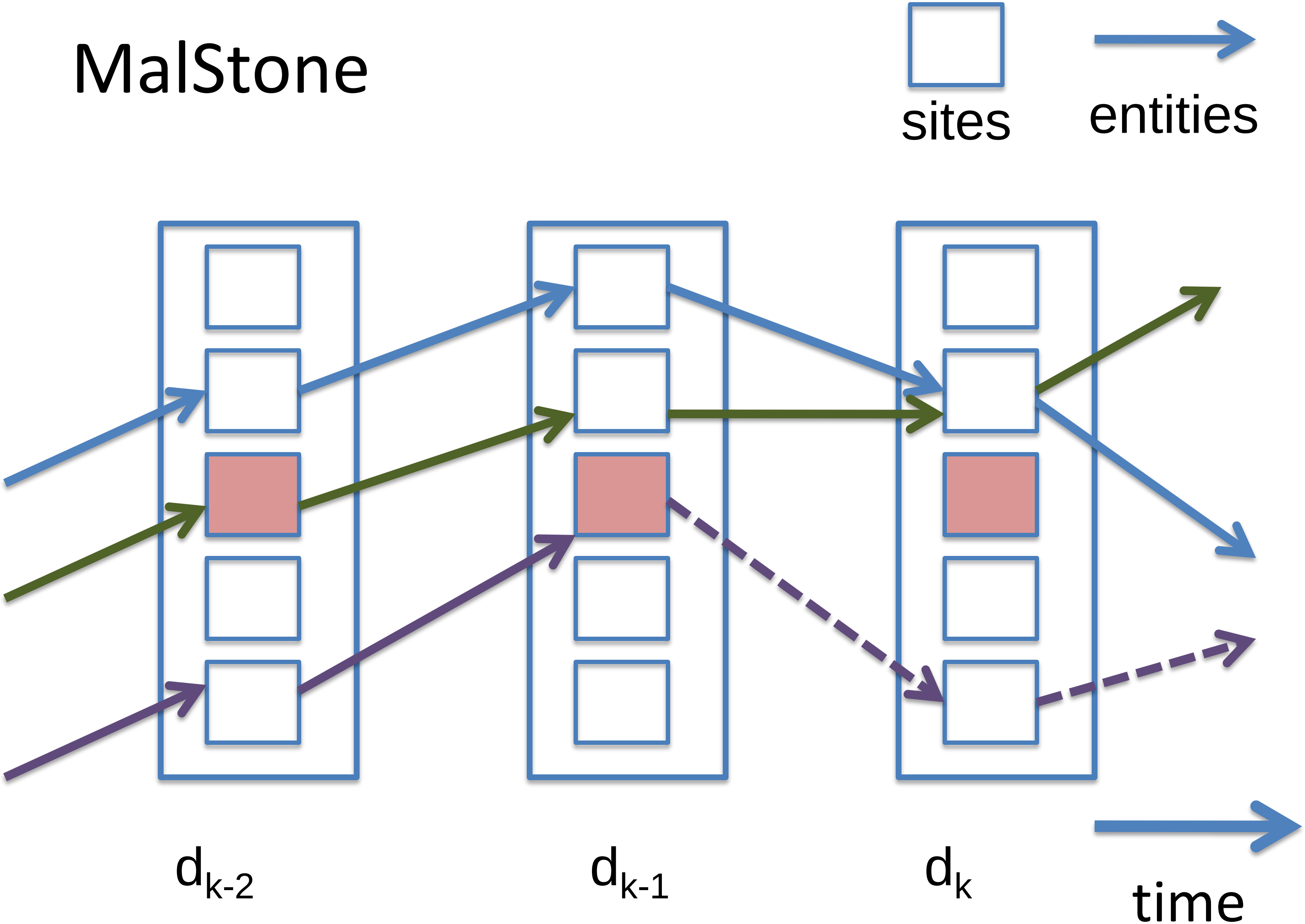
	MapReduce	Sphere
Storage	Disk data	Disk & in-memory
Processing	Map followed by Reduce	Arbitrary user defined functions
Data exchanging	Reducers pull results from mappers	UDF's push results to bucket files
Input data locality	Input data is assigned to nearest mapper	Input data is assigned to nearest UDF
Output data locality	NA	Can be specified

Terasort Benchmark

	1 Rack	**2 Racks**	**3 Racks**	**4 Racks**
Nodes	32	64	96	128
Cores	128	256	384	512
Hadoop	85m 49s	37m 0s	25m 14s	17m 45s
Sector	28m 25s	15m 20s	10m 19s	7m 56s
Speed up	3.0	2.4	2.4	2.2

Sector/Sphere 1.24a, Hadoop 0.20.1 with no replication on Phase 2 of Open Cloud Testbed with co-located racks.

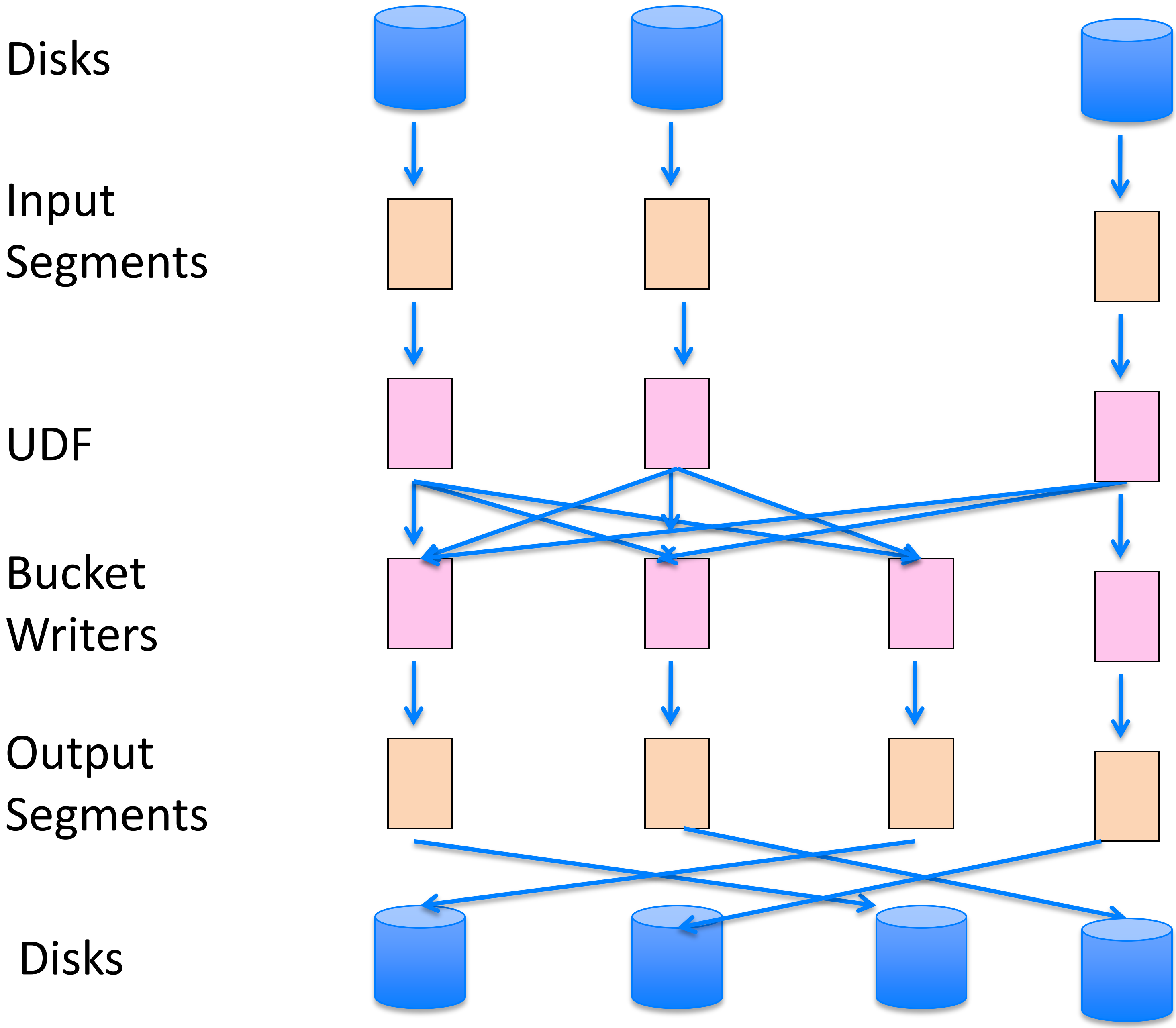
MalStone



MalStone Benchmark

	MalStone A	MalStone B
Hadoop	455m 13s	840m 50s
Hadoop streaming with Python	87m 29s	142m 32s
Sector/Sphere	33m 40s	43m 44s
Speed up (Sector v Hadoop)	13.5x	19.2x

Sector/Sphere 1.20, Hadoop 0.18.3 with no replication on Phase 1 of Open Cloud Testbed in a single rack. Data consisted of 20 nodes with 500 million 100-byte records / node.

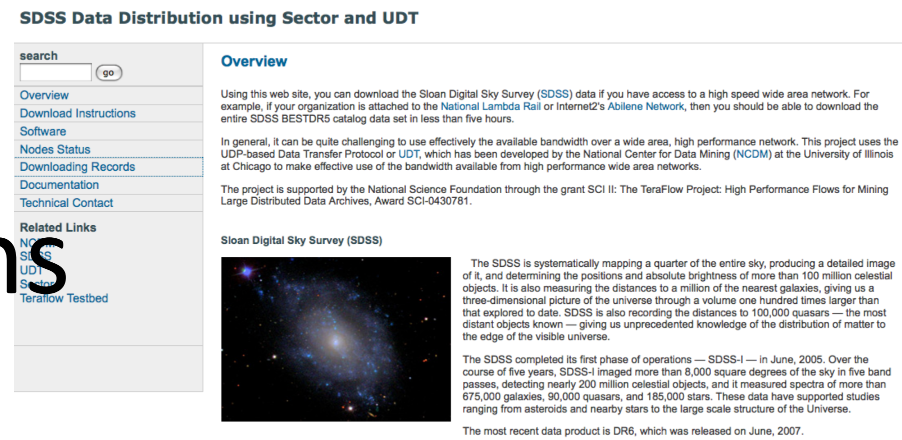


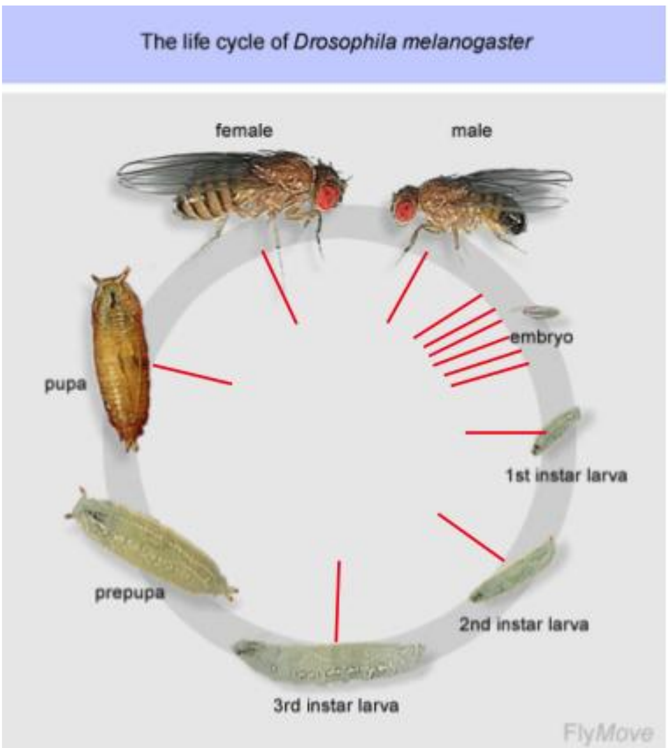
- Files not split into blocks
- Directory directives
- In-memory objects

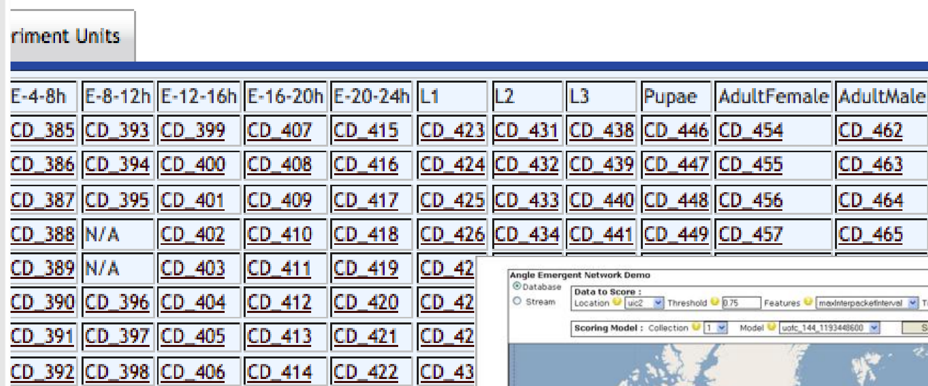
Sector Summary

- Sector is fastest open source large data cloud
  - As measured by MalStone & Terasort
- Sector is easy to program
  - UDFs, MapReduce & Python over streams
- Sector does not require extensive tuning
- Sector is secure
  - A HIPAA compliant Sector cloud is being launched
- Sector is reliable
  - Sector supports multiple active master node servers

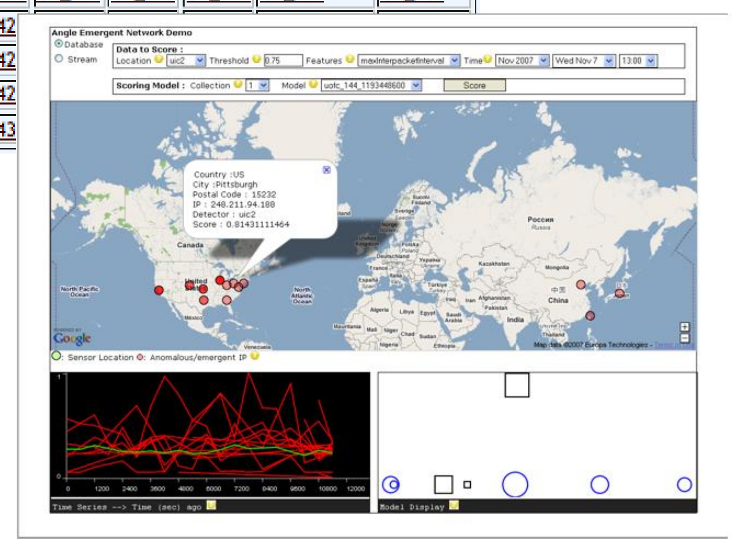
Part 6.
Sector Applications



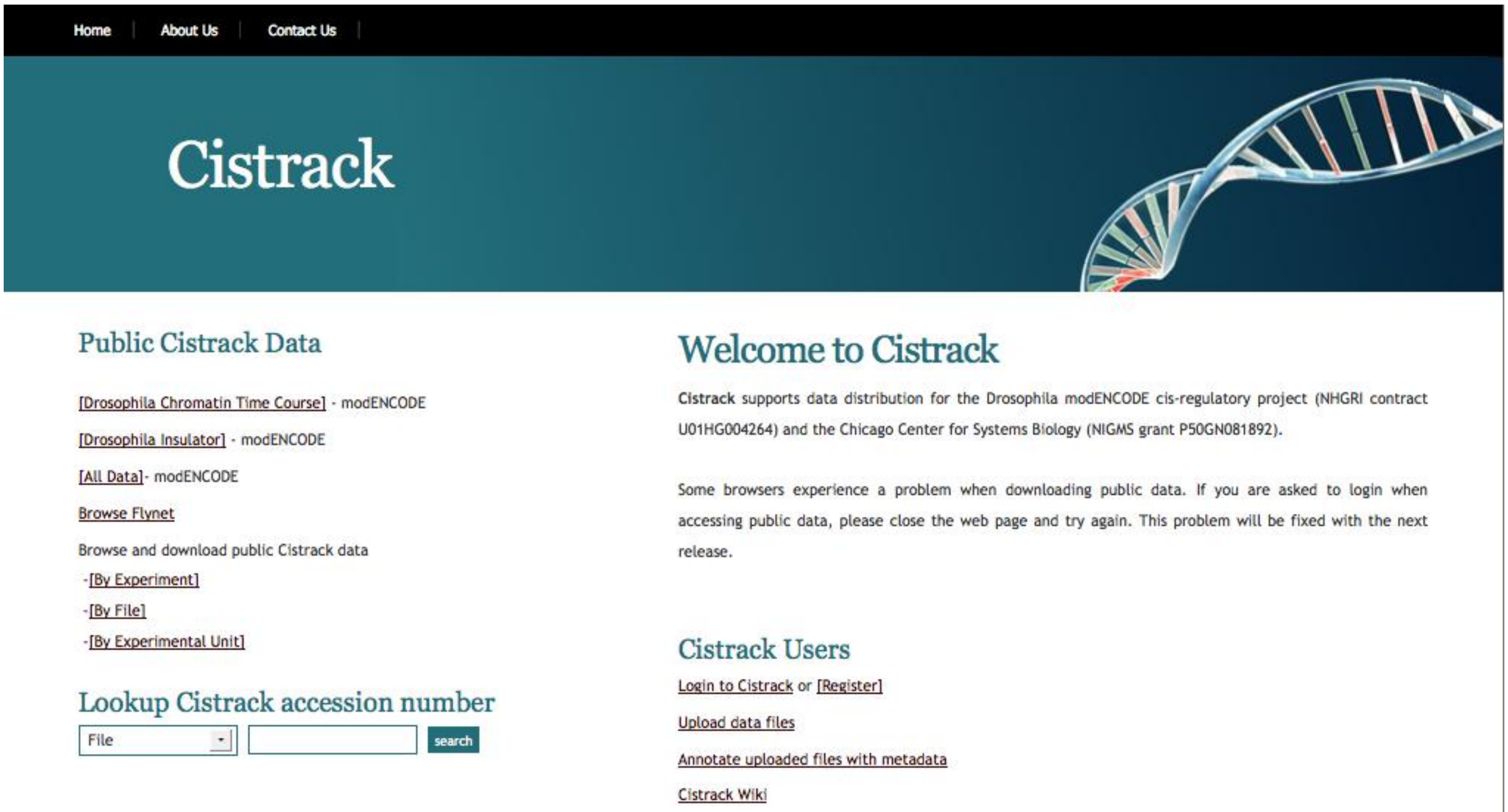



riment Units

E-4-8h	E-8-12h	E-12-16h	E-16-20h	E-20-24h	L1	L2	L3	Pupae	AdultFemale	AdultMale
CD_385	CD_393	CD_399	CD_407	CD_415	CD_423	CD_431	CD_438	CD_446	CD_454	CD_462
CD_386	CD_394	CD_400	CD_408	CD_416	CD_424	CD_432	CD_439	CD_447	CD_455	CD_463
CD_387	CD_395	CD_401	CD_409	CD_417	CD_425	CD_433	CD_440	CD_448	CD_456	CD_464
CD_388	N/A	CD_402	CD_410	CD_418	CD_426	CD_434	CD_441	CD_449	CD_457	CD_465
CD_389	N/A	CD_403	CD_411	CD_419	CD_42					
CD_390	CD_396	CD_404	CD_412	CD_420	CD_42					
CD_391	CD_397	CD_405	CD_413	CD_421	CD_42					
CD_392	CD_398	CD_406	CD_414	CD_422	CD_43					

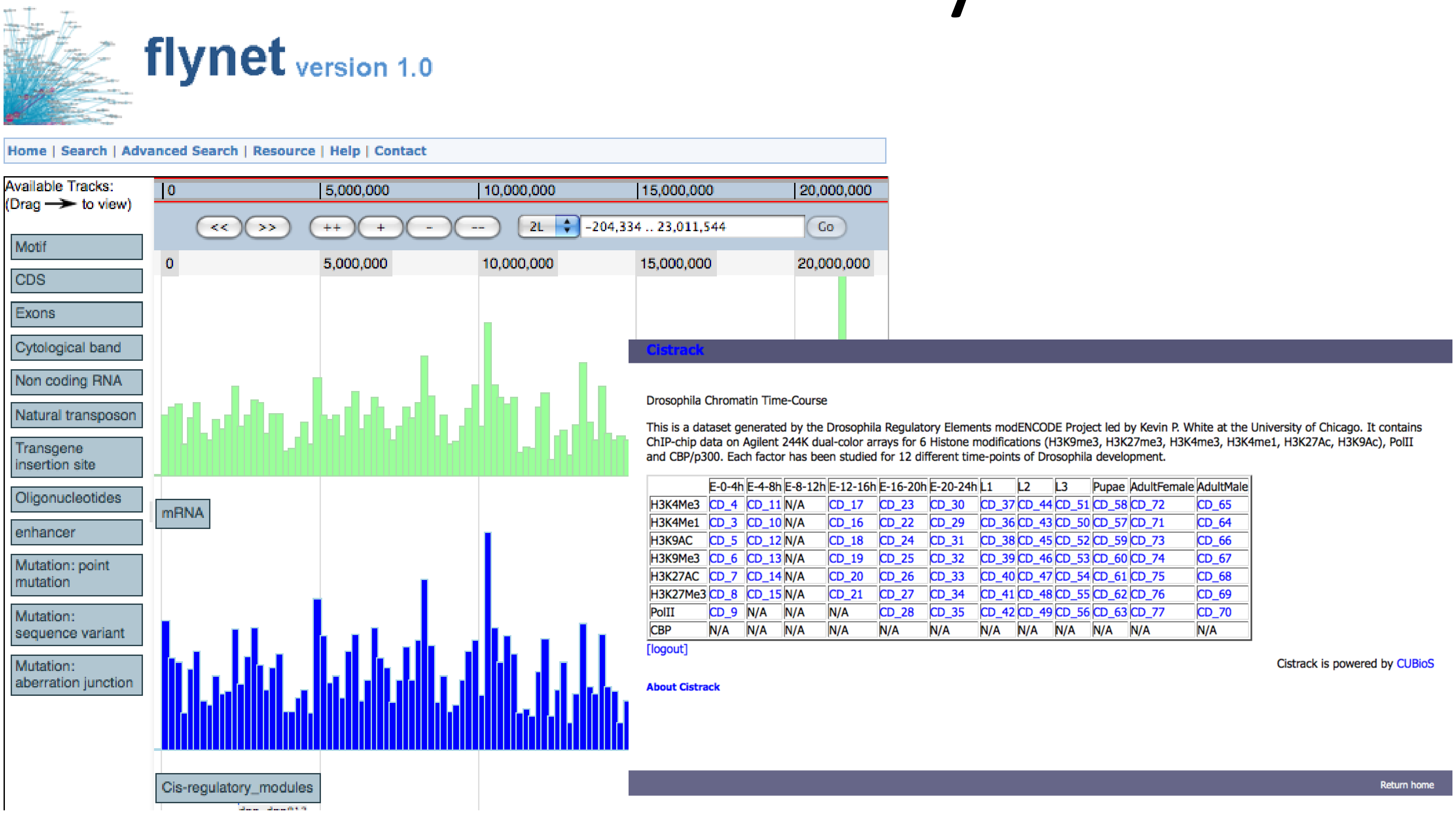


App 1: Bionimbus

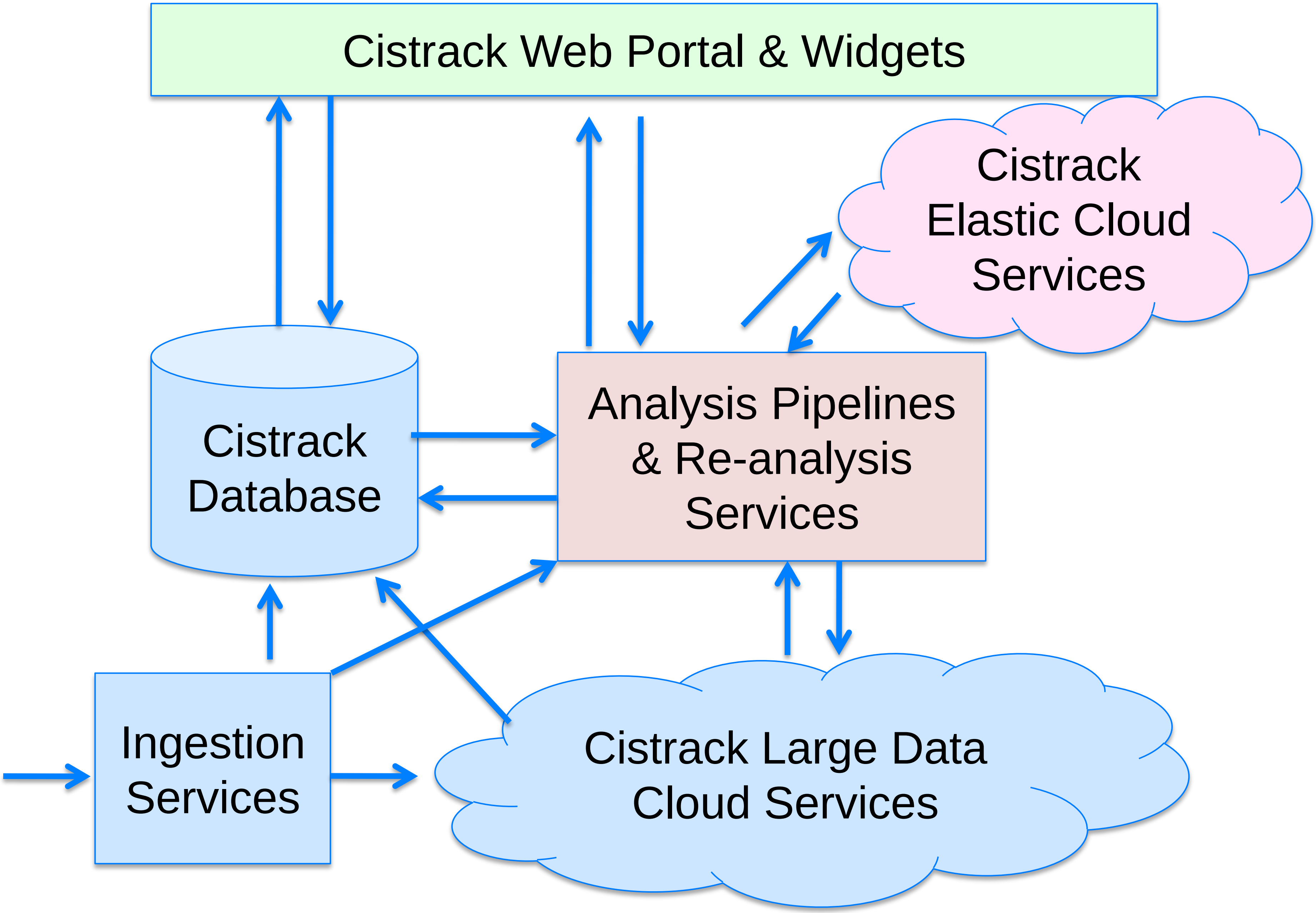


www.bionimbus.org

App 2. Sector Application: Cistrack & Flynet



	E-0-4h	E-4-8h	E-8-12h	E-12-16h	E-16-20h	E-20-24h	L1	L2	L3	Pupae	AdultFemale	AdultMale
H3K4Me3	CD_4	CD_11	N/A	CD_17	CD_23	CD_30	CD_37	CD_44	CD_51	CD_58	CD_72	CD_65
H3K4Me1	CD_3	CD_10	N/A	CD_16	CD_22	CD_29	CD_36	CD_43	CD_50	CD_57	CD_71	CD_64
H3K9AC	CD_5	CD_12	N/A	CD_18	CD_24	CD_31	CD_38	CD_45	CD_52	CD_59	CD_73	CD_66
H3K9Me3	CD_6	CD_13	N/A	CD_19	CD_25	CD_32	CD_39	CD_46	CD_53	CD_60	CD_74	CD_67
H3K27AC	CD_7	CD_14	N/A	CD_20	CD_26	CD_33	CD_40	CD_47	CD_54	CD_61	CD_75	CD_68
H3K27Me3	CD_8	CD_15	N/A	CD_21	CD_27	CD_34	CD_41	CD_48	CD_55	CD_62	CD_76	CD_69
PolII	CD_9	N/A	N/A	N/A	CD_28	CD_35	CD_42	CD_49	CD_56	CD_63	CD_77	CD_70
CBP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A



Cistrack Web Portal & Widgets
Cistrack Elastic Cloud Services
Cistrack Database
Analysis Pipelines & Re-analysis Services
Ingestion Services
Cistrack Large Data Cloud Services

App 3: Bulk Download of the SDSS

Source	Destin.	LLPR*	Link	Bandwidth
Chicago	Greenbelt	0.98	1 Gb/s	615 Mb/s
Chicago	Austin	0.83	10 Gb/s	8000 Mb/s

SDSS Data Distribution using Sector and UDT

search

go

- Overview
- Download Instructions
- Software
- Nodes Status
- Downloading Records
- Documentation
- Technical Contact

Related Links

- NCDM
- SDSS
- UDT
- Sector
- Teraflow Testbed

Overview

Using this web site, you can download the Sloan Digital Sky Survey (SDSS) data if you have access to a high speed wide area network. For example, if your organization is attached to the National Lambda Rail or Internet2's Abilene Network, then you should be able to download the entire SDSS BESTDR5 catalog data set in less than five hours.

In general, it can be quite challenging to use effectively the available bandwidth over a wide area, high performance network. This project uses the UDP-based Data Transfer Protocol or UDT, which has been developed by the National Center for Data Mining (NCDM) at the University of Illinois at Chicago to make effective use of the bandwidth available from high performance wide area networks.

The project is supported by the National Science Foundation through the grant SCI II: The TeraFlow Project: High Performance Flows for Mining Large Distributed Data Archives, Award SCI-0430781.

Sloan Digital Sky Survey (SDSS)

The SDSS is systematically mapping a quarter of the entire sky, producing a detailed image of it, and determining the positions and absolute brightness of more than 100 million celestial objects. It is also measuring the distances to a million of the nearest galaxies, giving us a three-dimensional picture of the universe through a volume one hundred times larger than that explored to date. SDSS is also recording the distances to 100,000 quasars — the most distant objects known — giving us unprecedented knowledge of the distribution of matter to the edge of the visible universe.

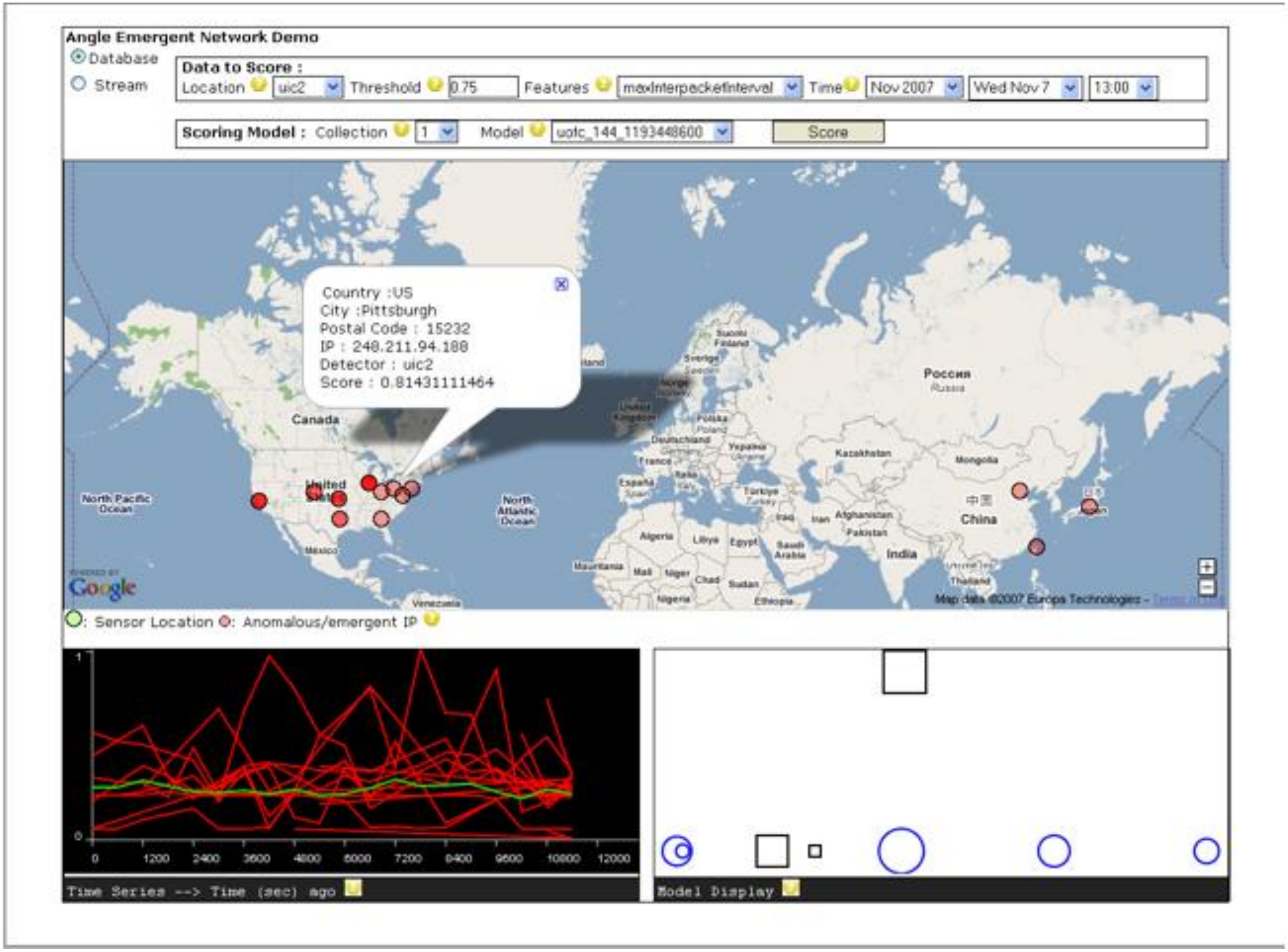
The SDSS completed its first phase of operations — SDSS-I — in June, 2005. Over the course of five years, SDSS-I imaged more than 8,000 square degrees of the sky in five band passes, detecting nearly 200 million celestial objects, and it measured spectra of more than 675,000 galaxies, 90,000 quasars, and 185,000 stars. These data have supported studies ranging from asteroids and nearby stars to the large scale structure of the Universe.

The most recent data product is DR6, which was released on June, 2007.

- •LLPR = local / long distance performance
- • Sector LLPR varies between 0.61 and 0.98

Recent Sloan Digital Sky Survey (SDSS) data release is 14 TB in size

App 4: Anomalies in Network Data



Sector Applications

- Distributing the 15 TB Sloan Digital Sky Survey to astronomers around the world (with JHU, 2005)
- Managing and analyzing high throughput sequence data (Cistrack, University of Chicago, 2007).
- Detecting emergent behavior in distributed network data (Angle, won SC 07 Analytics Challenge)
- Wide area clouds (won SC 09 BWC with 100 Gbps wide area computation)
- New ensemble-based algorithms for trees
- Graph processing
- Image processing (OCC Project Matsu)

Credits

- Sector was developed by Yunhong Gu from the University of Illinois at Chicago and verycloud.com

For More Information

For more information, please visit

sector.sourceforge.net

rgrossman.com (Robert Grossman)

users.lac.uic.edu/~yunhong (Yunhong Gu)