Filling Gaps in the Argonne History and

Can We converge? Big Data, Big Compute, Big Interaction

Rick Stevens

stevens@anl.gov

Argonne National Laboratory

The University of Chicago

There are no solved problems. There are only more-or-less solved problems.

Henri Poincaré

Scientific Simulation Initiative

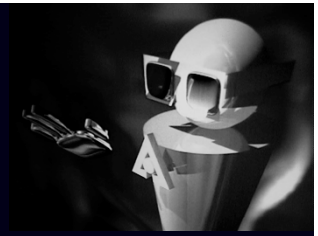
In FY2000 the U.S. Department of Energy requested funding to initiate the **Scientific Simulation Initiative**. The SSI seeks to:

"… revolutionize the way that the Department of Energy solves its most demanding, mission-critical problems by fully utilizing the power of advanced computational simulation."

Ernest J. Moniz
Undersecretary of Energy
December 1997

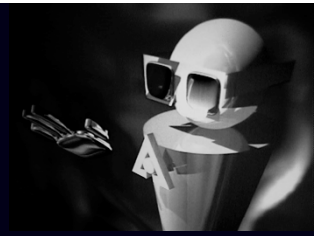
mcs

Status of DOE SSI Initiative



- $241M FY00 over request to administration (Aug'98)
- $70M request for FY00 from Administration to Congress
- Current Congressional budget Marks moved SSI funding to the NSF CISE budget
  - Language in current bill very restrictive
- NSF and DOE are planning a joint Hardware and Computer Science Center solicitation in the Fall
- DOE has appointed Thom Dunning as head of SSI office
- DOE has appointed Ed Oliver as head of OASCR
- OASCR is setting up a SCAC (Scientific Computing Advisory Committee)

Status of DOE SSI II



- IT^2/NITRD Bill asks for an increment of $366M for NSF/DOE/NASA/DARPA/EPA/NIH for IT
  - DOE is a zero increment currently
  - NITRD is a five year authorization that grows budgets according to the PITAC report
- Dunning's vision for the Scientific Simulation target's a roughly $500M program by 2004
- DOE FY01 Budget planning includes SSI
- DOE SSI program should strongly support future DOE Scientific and Facilities Initiatives, Nanoscience, Biocomplexity, etc.

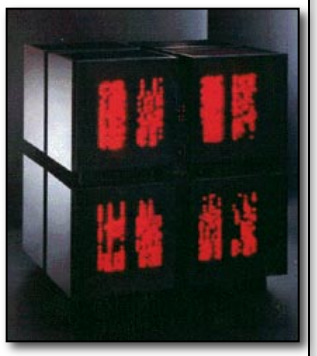
Footnote NSF Initiatives for FY00

- Computing, Nanotechnology, Biocomplexity

Computing at MCS before the ALCF

The ACRF: 1985-1993

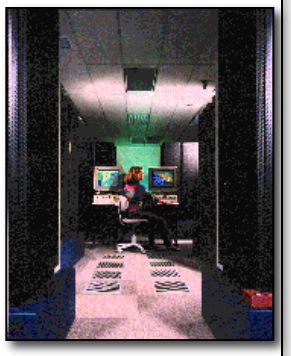
Systems: CM-2, Encore, Sequent, BBN Butterfly, Intel Gamma, …



Mission:
Computer Science & Outreach

The IBM SP-1: 1993-1997

System: 128 Nodes
The First IBM SP installed.
(Simultaneous with Cornell.)



Mission:
Grand Challenge Apps
Work with IBM

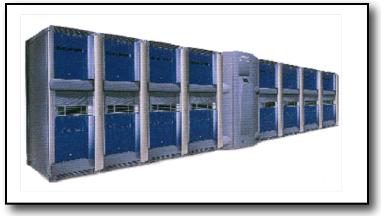
The IBM SP-2: 1996-present

System: 80 Nodes
Upgrade of SP-1

Mission:
Grand Challenge Applications

The SGI Origin 2000: 1998 -2001

System: 128 CPUS
12 IR Pipes

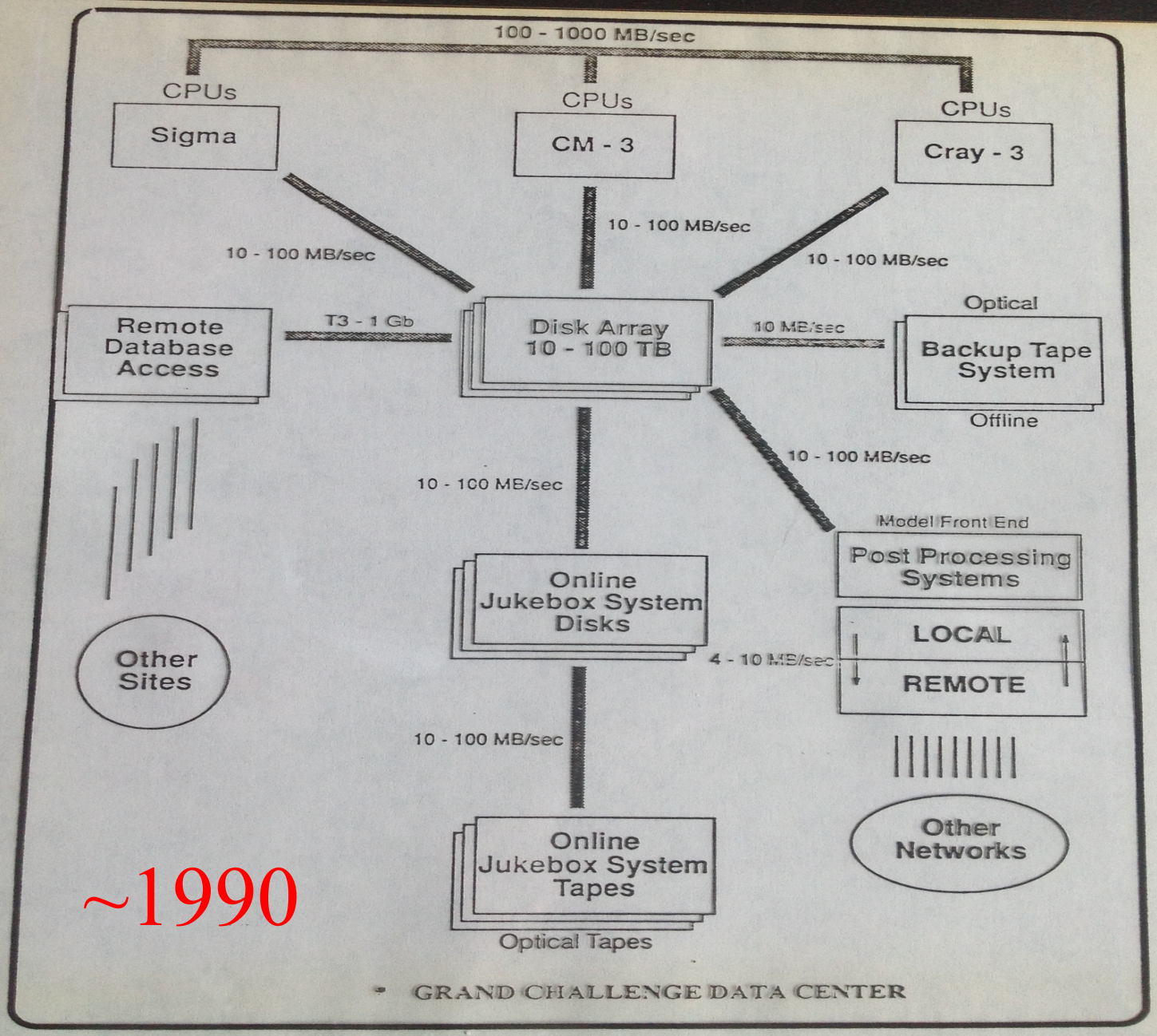


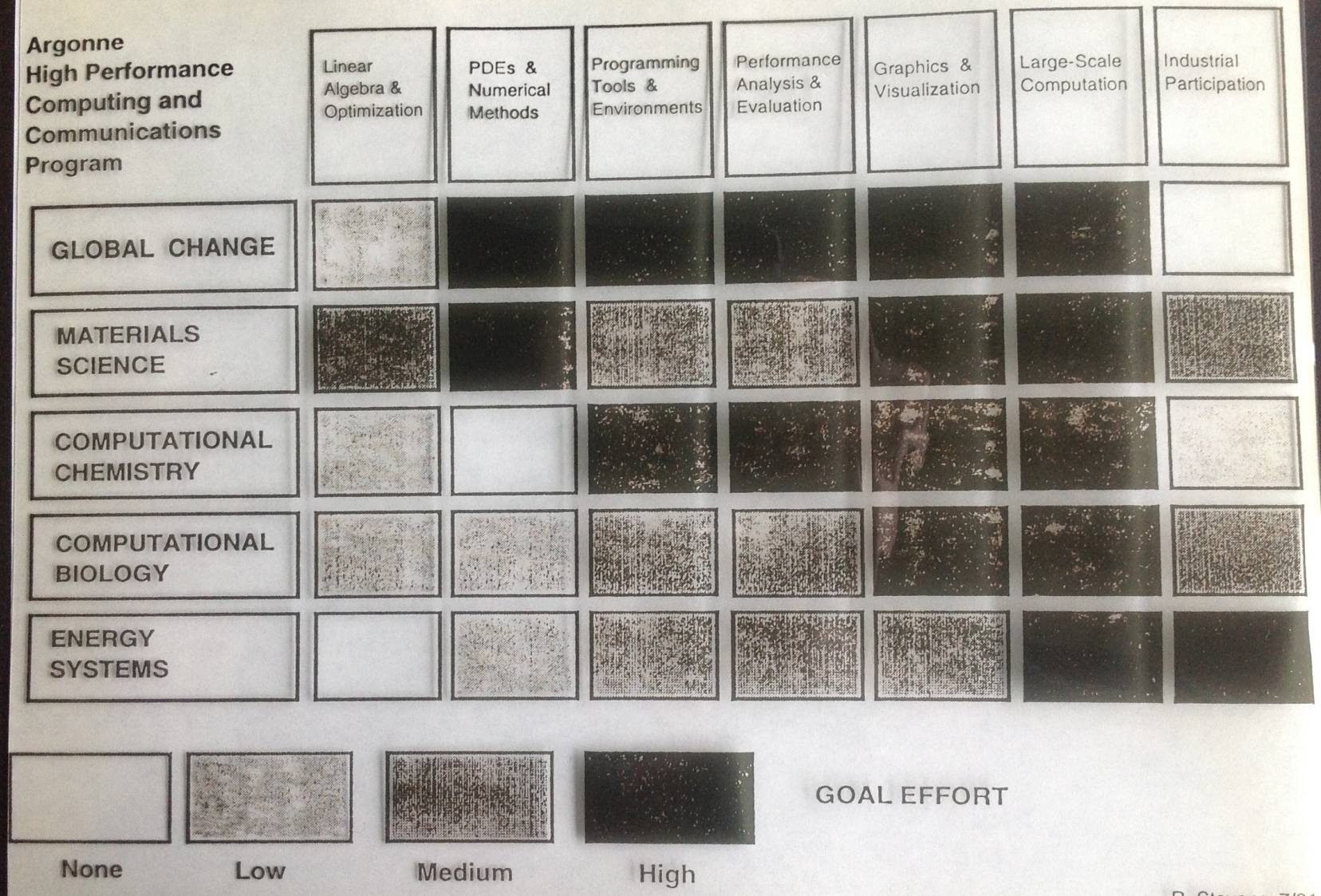
Missions:
Visualization Research
SMP Testbed for CS
Grand Challenge Apps

Division Organization

- Small division
- Simple organizational structure
- Minimizes management levels
- Encourages interdisciplinary projects

100% COTTON
ACA
101
PREWASHED


100 - 1000 MB/sec
CPUs
Sigma
CPUs
CM - 3
CPUs
Cray - 3
10 - 100 MB/sec
10 - 100 MB/sec
10 - 100 MB/sec
Remote Database Access
T3 - 1 Gb
Disk Array 10 - 100 TB
10 MB/sec
Optical
Backup Tape System
Offline
10 - 100 MB/sec
10 - 100 MB/sec
Model Front End
Post Processing Systems
Online Jukebox System Disks
LOCAL
4 - 10 MB/sec
REMOTE
Other Sites
10 - 100 MB/sec
Other Networks
Online Jukebox System Tapes
Optical Tapes
~1990
GRAND CHALLENGE DATA CENTER


Argonne High Performance Computing and Communications Program
Linear Algebra & Optimization
PDEs & Numerical Methods
Programming Tools & Environments
Performance Analysis & Evaluation
Graphics & Visualization
Large-Scale Computation
Industrial Participation
GLOBAL CHANGE
MATERIALS SCIENCE
COMPUTATIONAL CHEMISTRY
COMPUTATIONAL BIOLOGY
ENERGY SYSTEMS
None
Low
Medium
High
GOAL EFFORT
R. Stevens, 7/91

~1991

1991

New Problems in Understanding Architectures

Understanding the performance of the next generation of parallel systems is becoming increasingly difficult.

- Parallelism is present at many levels

CPU
  - length and type of pipelines
  - number and non-homogeneity of functional units

TOPOLOGY
  - number of shared memory nodes in a cluster
  - special purpose hardware for routing/switching

SCALE
  - number of nodes
  - number and location of disks and tapes

- Simple performance models are of limited use

- However, the scalability of an algorithm and its implementation are increasingly important as machines get larger.

MCS

High-End Machine Specifications

~1991

	100 GFLOPS		TFLOPS		PFLOPS
	1993	1994	1995		2000??
Peak (Gflops)	150	400	1000		1000000
RAM (GBytes)	32	128	300		30000
Disk (GBytes)	256	512	1000		1000000
#CPUs	1000+	1000+	1000+		10K-100K
Cost	$30M	$30M	$30M		$30-50M

3 orders of Magnitude in < 5 years

How long CAN THIS RATE BE MAINTAINED ??

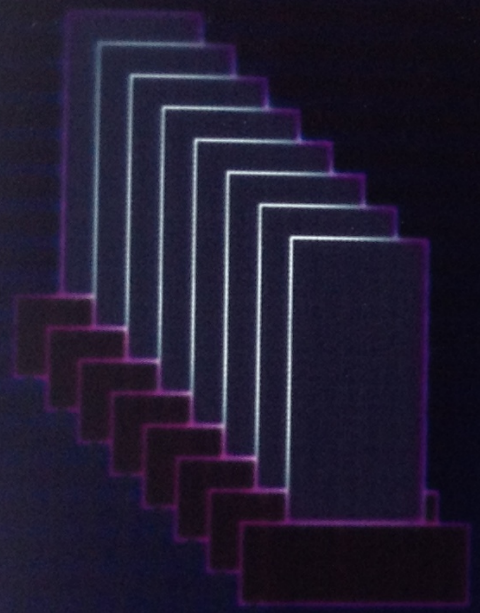
"WE ARE RUNNING OUT of COMPUTER SCIENCE!"

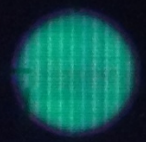
JUSTIN Rattner

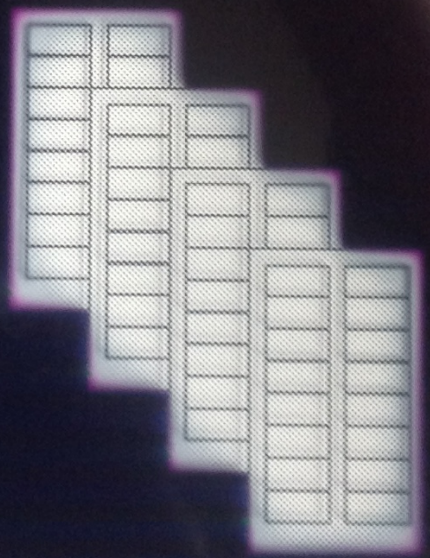
Technology Fellow Intel Corp.

Argonne IBM SP1 Configuration

~1993







128 node IBM SP1

125 Mflops/node
128 MB/node
1 GB disk/node
Omega Switch

ANL gigabit LAN

950 I/O servers
2 GB buffers

9570 arrays
220 GB disk

HIPPI
framebuffer

Computing at ANL in the year 2000

~1993

- access to local 10 Teraflop computer
- access to local 100 TB archive
- 2000 portable computers (3000 employees)
- 2500 universal workstations (Gflops, Gbytes)
- 1000 portable communications devices
- 1500 homes on-line via ISDN
- 3000 gigabit (optical) network nodes
- 4000 old style phones
- 1000 video phones / 100 portable (wireless)

Sample Scenario

~1993

- pocket computer with radio network (merging with cell phones)
- tablet based PCs (smart paperwork)
- universal workstation (PC, Mac, Unix software)
- gigabit to each desktop (video + voice + data)
- transparent access to MPPs and Clusters
- merging of office, home, travel computing environments
- universal archive > 100 TB

A Moore's Law for Software? 1996

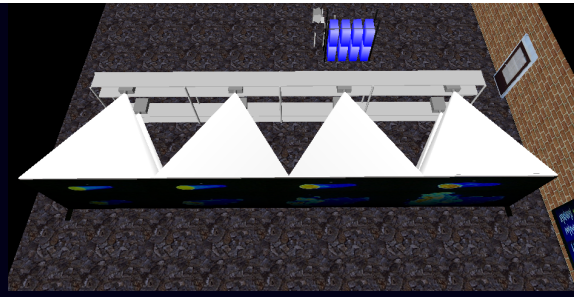
- The time it takes for some software metric "X" to double
- X ∈ {Capability, Performance, Reliability, Scalability, Portability, Usability, etc.}
- Capturing even some of these assessments will provide some indication that we do or do not understand the rate of change/improvement of software on a similar footing as that of hardware

- Conjecture #1: "*While hardware improves exponentially, software improves polynomially if at all!*"
- Conjecture #2: *"Even Open Source and Internet can't turn polynomial improvement into exponential improvement"*

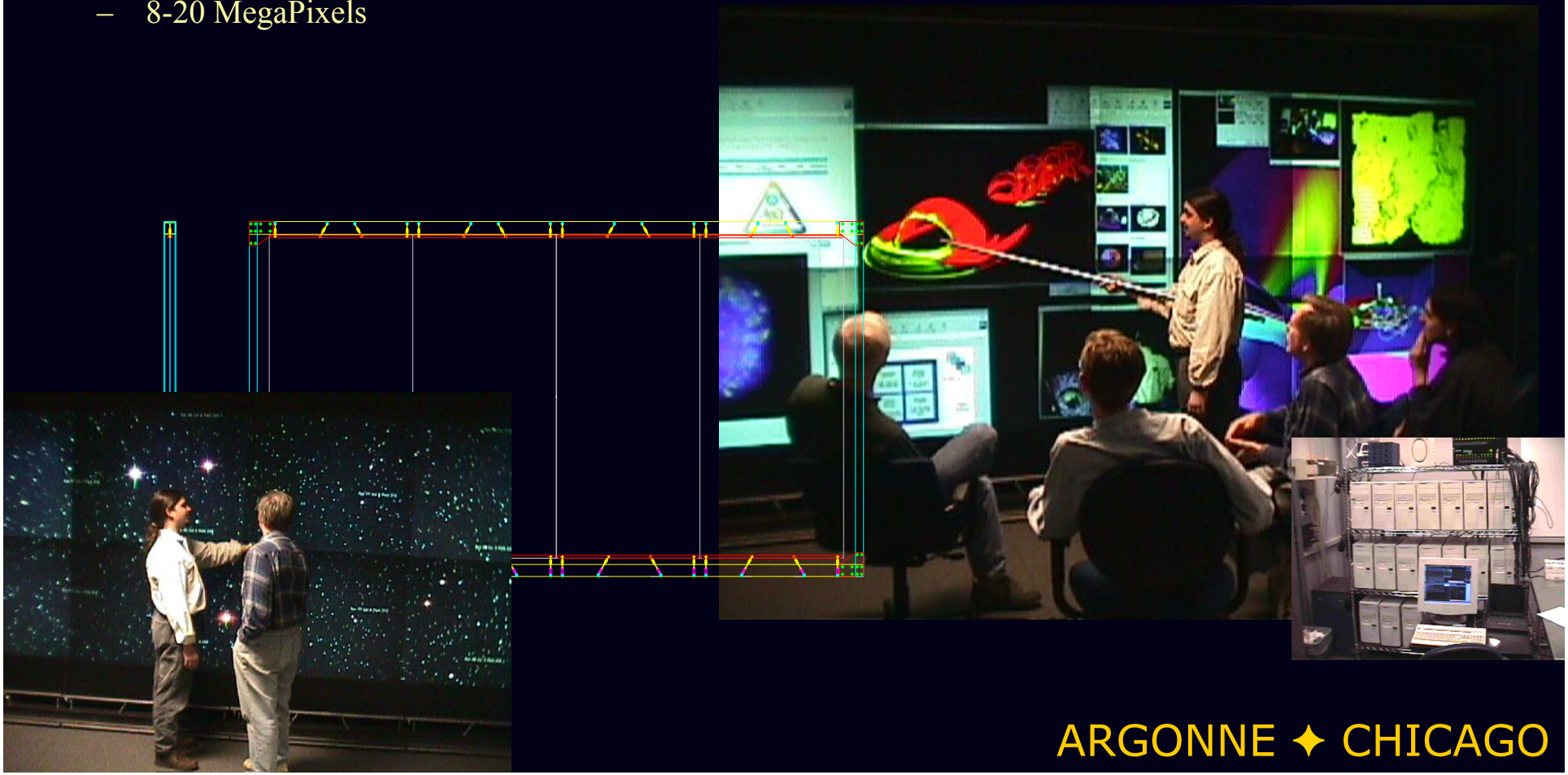
Universal Programming Literacy 1996

- "Reading, Writing, Arithmetic, and Programming"
- Everyone must know how to program
- Economic/technological justice requires everyone be capable of being both a consumer and producer of programming
- Must close the divide between those that develop technology and those that only "use" technology
- Enables spontaneous customizability

*Active*Mural, Giant Display Wall

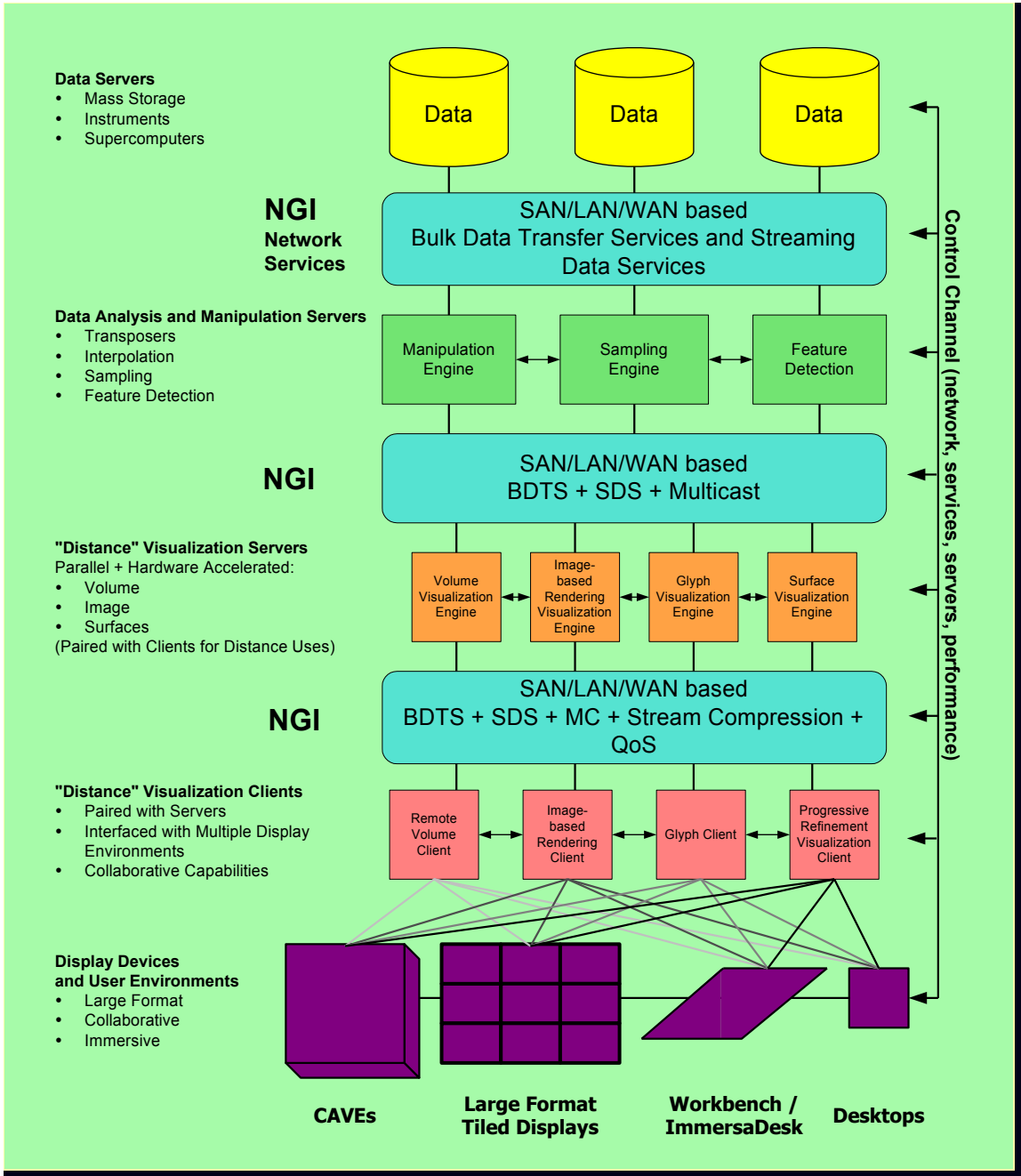


- Argonne, Princeton UIUC Collaboration
- 8’ x 16’ display wall
  - Jenmar Visual Systems BlackScreen™ technology, > 10000 lumens
  - 8 LCD → 15 DLP → 24 DLP
  - 8-20 MegaPixels

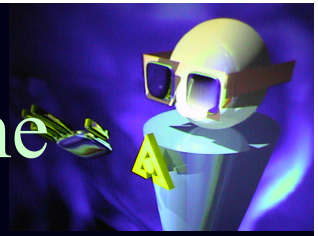


Midwest Networked CAVE and ImmersaDesk Sites Enabled by EMERGE

Alliance Virtual Machine Room Extends Diversity of Architectures

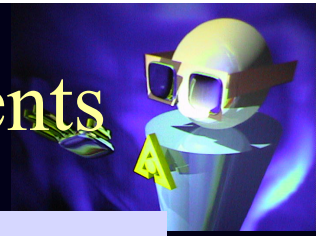


CorridorOne

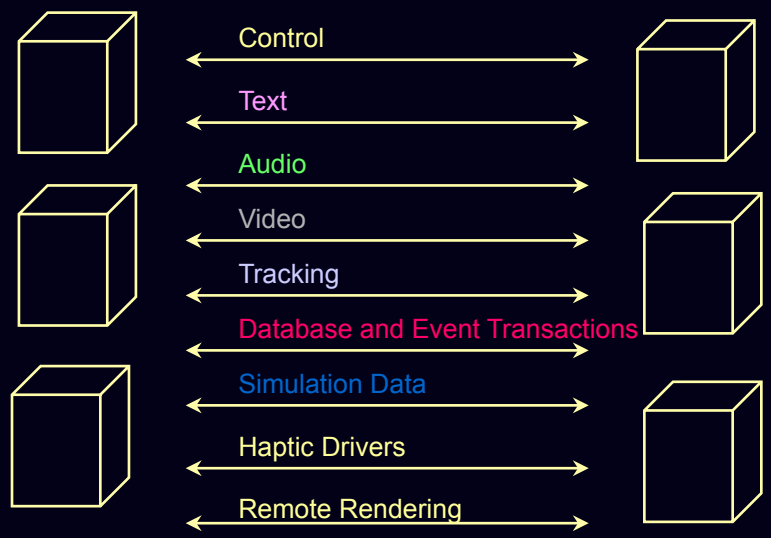


- Data Servers
- Analysis and Manipulation Engines
- Visualization Backend Servers
- Visualization Clients
- Display Device Interfaces
- Advanced Networking Services

Teleimmersion Networking Requirements



Type	Latency	Bandwidth	Reliable	Multicast	Security	Streaming	DynQos
Control	*< 30 ms*	64Kb/s	*Yes*	No	*High*	No	Low
Text	*< 100 ms*	64Kb/s	*Yes*	No	Medium	No	Low
Audio	*< 30 ms*	Nx128Kb/s	No	*Yes*	Medium	*Yes*	Medium
Video	*< 100 ms*	Nx5Mb/s	No	*Yes*	Low	*Yes*	Medium
Tracking	*< 10 ms*	Nx128Kb/s	No	*Yes*	Low	*Yes*	Medium
Database	*< 100 ms*	*> 1GB/s*	*Yes*	Maybe	Medium	No	*High*
Simulation	*< 30 ms*	*> 1GB/s*	Mixed	Maybe	Medium	Maybe	*High*
Haptic	*< 10 ms*	> 1 Mb/s	Mixed	Maybe	High	Maybe	*High*
Rendering	*< 30 ms*	*>1GB/s*	No	Maybe	Low	Maybe	Medium

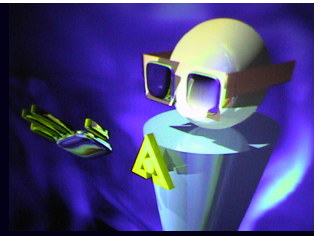


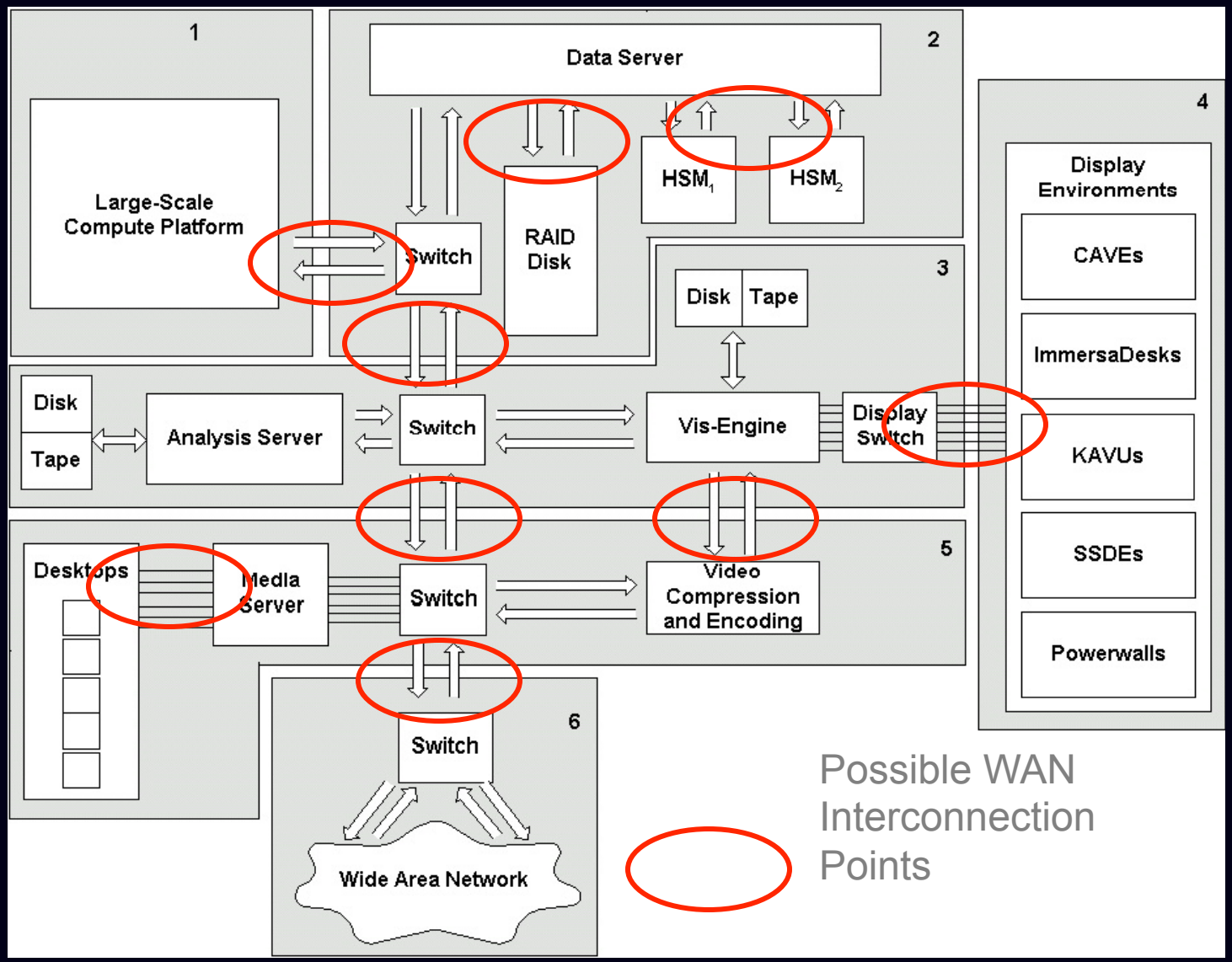
- Immersive environment
- Sharing of objects and virtual space
- Coordinated navigation and discovery
- Interactive control and synchronization
- Interactive modification of environment
- Scalable distribution of environment

Networking Requirements

Interaction	# Streams		Bandwidth		Aggregate		Average
		(Gb/s)			(Tb/s)		(Tb/s)
Steering	2	10	1	10	0.002	0.1	0.03
Remote HP Viz	2	10	1	10	0.002	0.1	0.03
Rem Desktop Viz	20	100	0.1	1	0.002	0.1	0.03
Remote Database	20	100	0.1	1	0.002	0.1	0.03
Data Exploration	5	20	0.5	5	0.0025	0.1	0.03
Collaboration	30	60	0.1	1	0.003	0.06	0.02
Instruments	10	20	0.08	5	0.0008	0.1	0.04
subtotal	89	320			0.01	0.66	0.23
Remote I/O	2	3	30	600	0.06	1.8	0.79
Computations	1	3	100	1000	0.1	3	1.10
TOTAL	92	326			0.17	5.46	2.12

Distributed Data and Visualization Corridor





~2000

Chiba City - the Argonne Scalable Cluster

256 computing nodes.
512 PIII CPUs.

32 visualization nodes.

8 storage nodes.
4TB of disk.

Myrinet interconnect.

Mission: Scalability and open source software testbed.

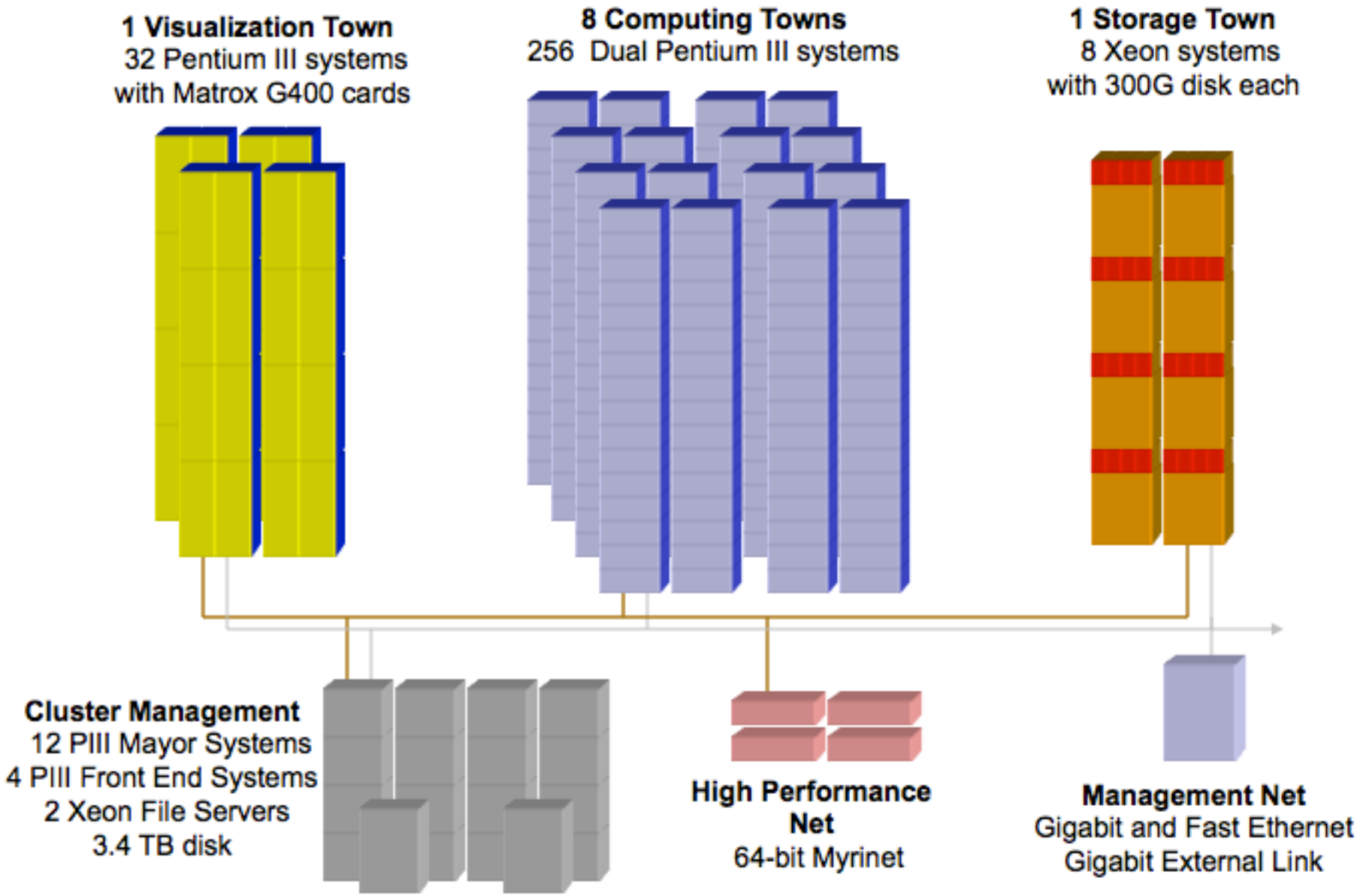
1 of 2 rows of Chiba City:

http://www.mcs.anl.gov/chiba/

mcs

Chiba City
The Argonne Scalable Cluster

~2000



27 Sep 1999

ALCF1 Project Scope

- Five year project including the acquisition, installation and operations a 100TF system in FY2007 and upgrade to 250-500TF in FY2008 and three+ years of operations FY09-FY11

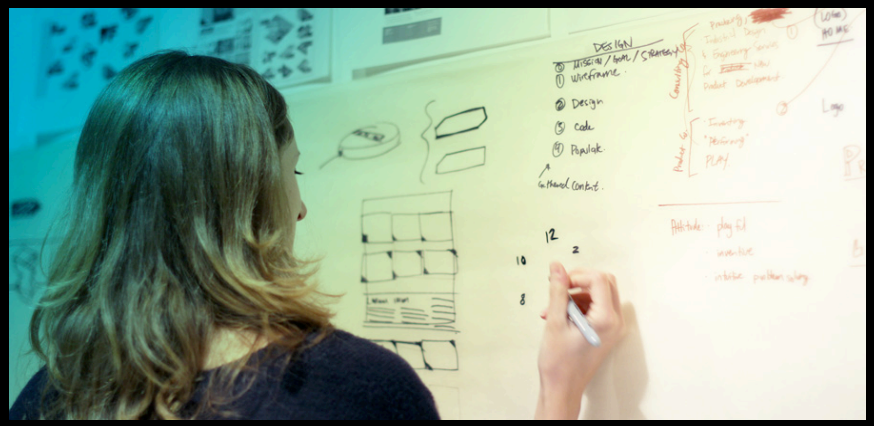
Magellan: Our OpenStack Private Cloud for Systems Biology

MIRA

DataDirect Networks
CP

What We Want Users To Do

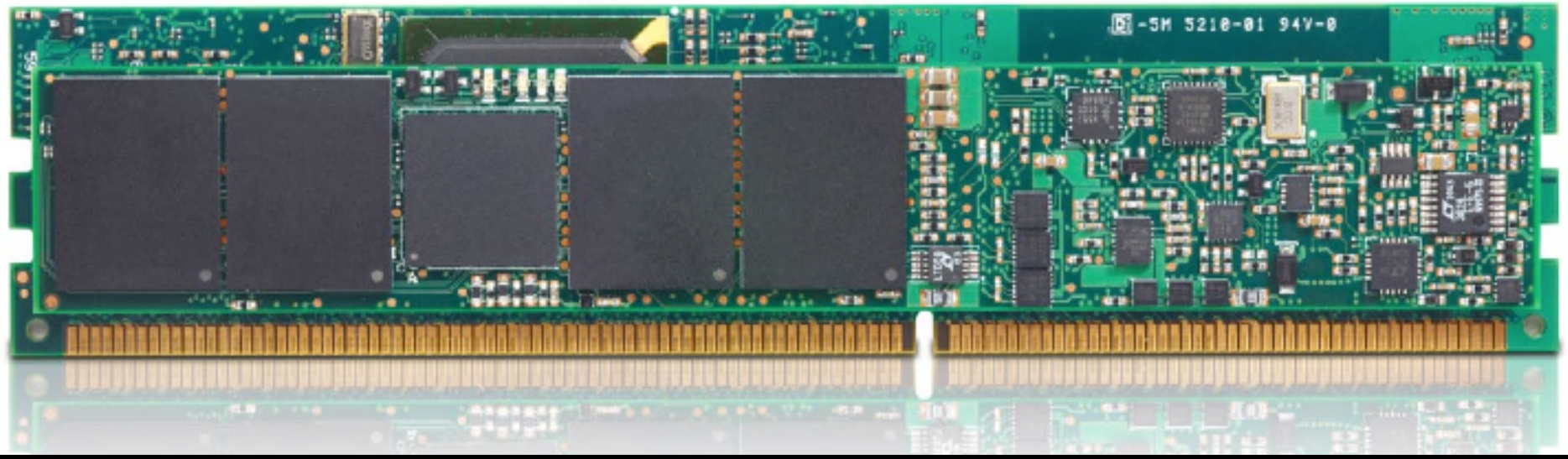
1. Explore
2. Integrate
3. Form Hypotheses
4. Make Inferences
5. Create Models
6. Test Models
7. Discover
8. Disseminate



Convergence

- Interactive parallel prototyping environment
- Seamless scale up to production (10^3x-10^6x)
- Integrated platform for analysis and simulation
- Same platform for publishing
- Persistent data regions in memory
- Programming language support for data analysis
- Large-scale interactive computing
- Seamless visualization and sharing

NVDIMM



Convergence Features

- Multiple node types with different ratios of DRAM to NVRAM
- NVRAM access
  - Memory access semantics
  - K/V access semantics
  - Legacy block device
- Persistence .. attachable memory region
- Distribution/Consolidation operations

The Gradient Machine

- Nodes with various DRAM:NVRAM ratios
  - 16 GB RAM : 64 GB NVRAM (1:4) – comp node
  - 16 GB RAM : 256 GB NVRAM (1:16) – $hybrid_1$ node
  - 16 GB RAM : 1 TB NVRAM (1:64) – $hybrid_2$ node
  - 16 GB RAM : 4 TB NVRAM (1:256) – store node
- Machine consists of sets of nodes of various types (X of comp, Y of store, etc.)
- Supernode could consistent of node collections with dynamic network provisioning

16 GB DRAM : 64 GB NVRAM

16 GB DRAM : 256 GB NVRAM

16 GB DRAM : 1 TB NVRAM

16 GB DRAM : 4 TB NVRAM

Imagine 1 M nodes of each type..

64 PB DRAM

5540 PB of NVRAM

85x DRAM storage

Jobs run where storage requirements are met

Data can migrate

Compute can migrate

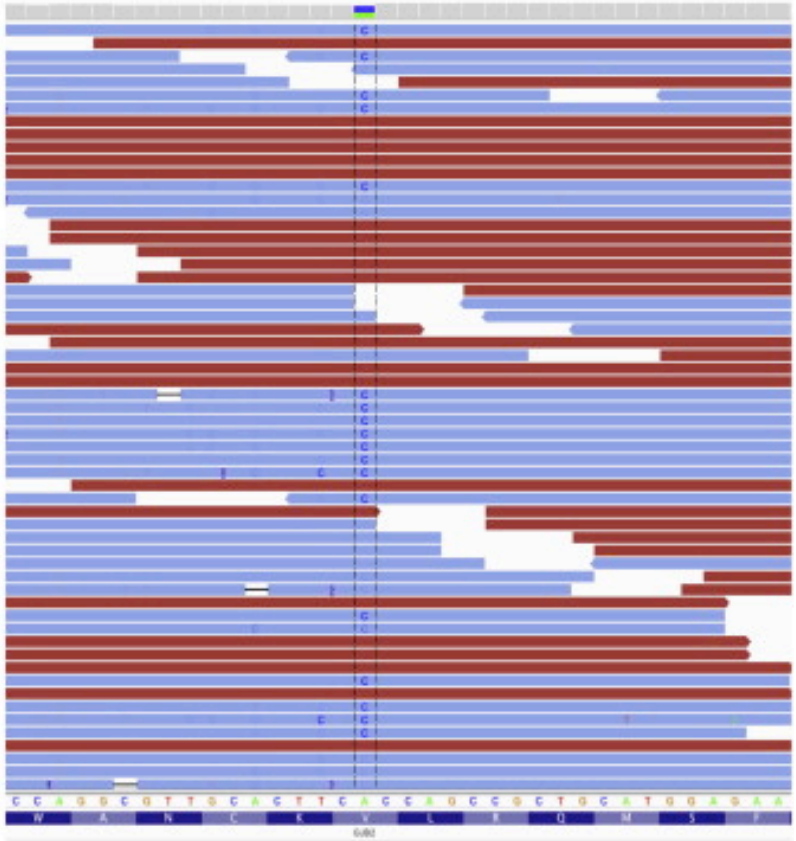
Bandwidth per NVRAM BYTE varies

Bandwidth per DRAM byte is constant

a Illumina HiSeq
Generate Millions of Sequencing Reads

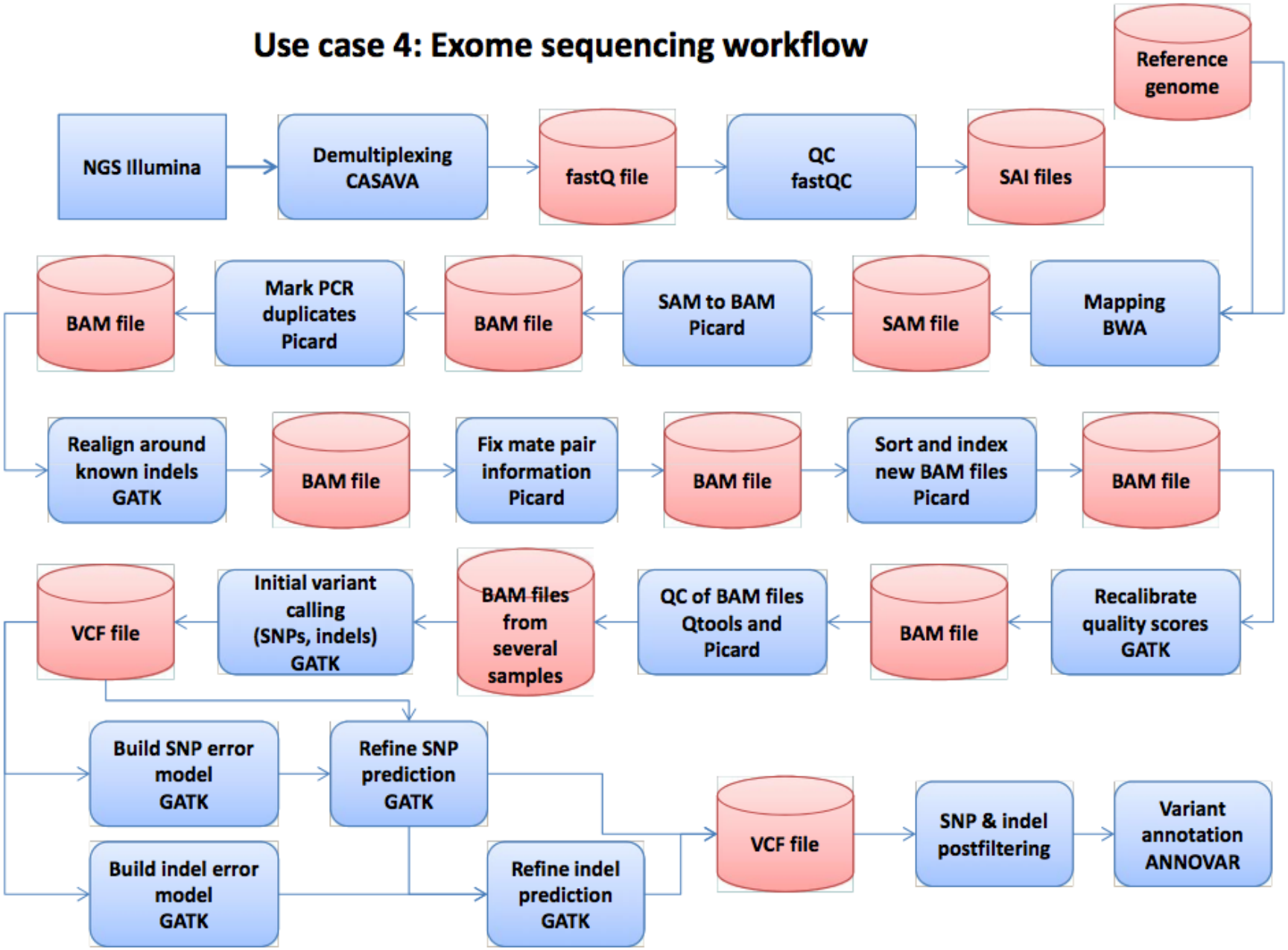
b Raw Sequencing Read Output

c Align Millions of Sequencing Reads to the Reference Genome

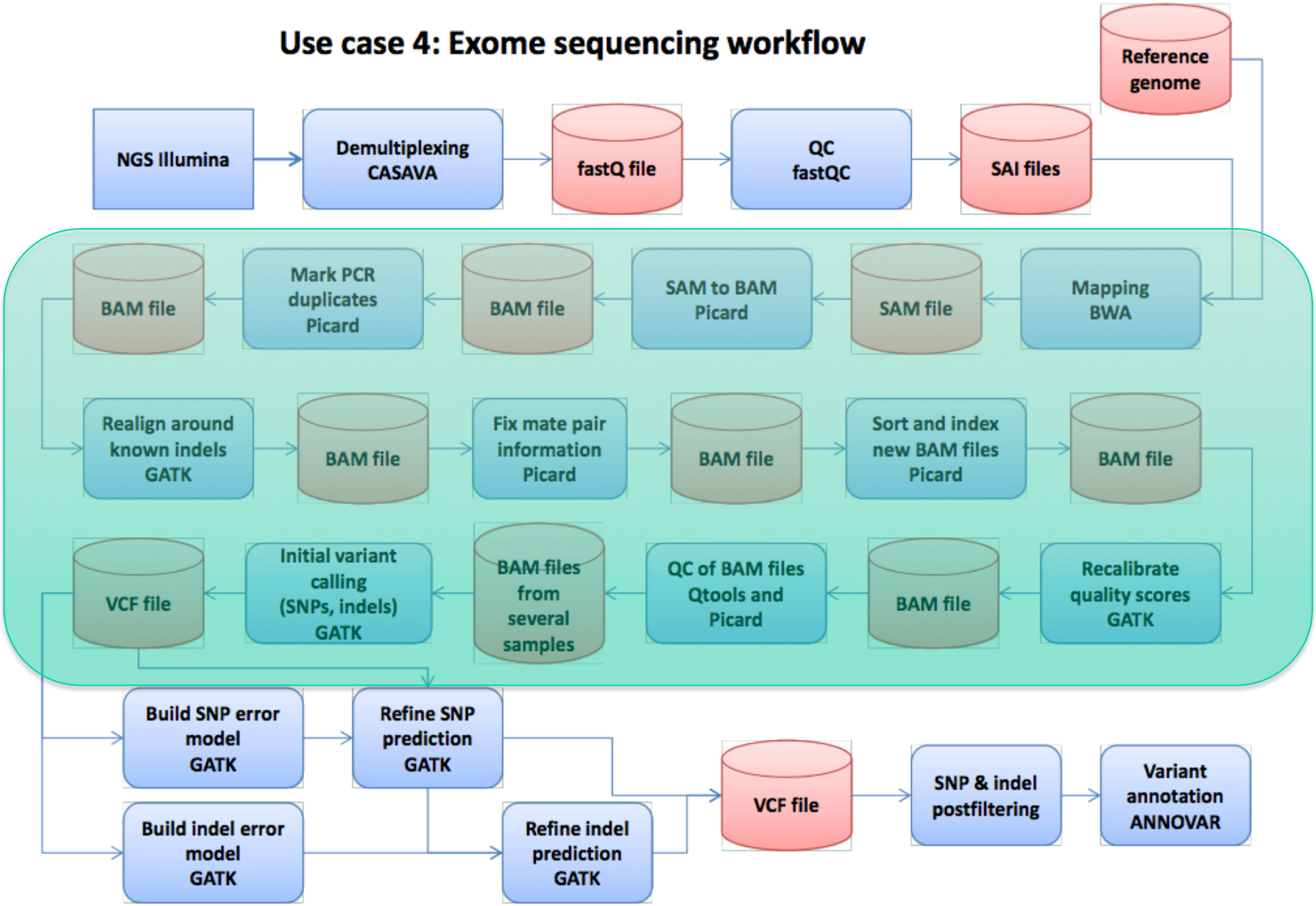


d Identify and Annotate Variant:
GJB2 Val167Met carrier

Use case 4: Exome sequencing workflow



Use case 4: Exome sequencing workflow



KMI: A Domain Specific Library for Extreme-Scale DNA Sequence Analysis

Huiwei Lv[†§*], Fangfang Xia[‡], Pavan Balaji[‡], Ralf Gunter Correa Carvalho[¶]
Guangming Tan[†], Ninghui Sun[†], Rick Stevens[‡¶]
[†]State Key Laboratory of Computer Architecture, ICT, CAS
[‡]Argonne National Laboratory
[§]Graduate University of Chinese Academy of Sciences
[¶]The University of Chicago
{lvhuiwei,tgm,snh}@ncic.ac.cn, {fangfang, balaji, stevens}@anl.gov

ABSTRACT

The use of k-mer, short substrings of length k, is at the heart of many bioinformatics algorithms. We propose K-mer Matching Interface (KMI), a domain-specific library for extreme-scale sequence analysis. It is designed to scale to very large size: petabytes of sequence data across hundreds of thousands of processors. We define a programming model for overlap computation in genome assembly and genome mapping algorithms. The programming model is easy-to-use, flexible and portable. It extracts a common set of low level operations needed for k-mer matching for sequence analysis. We also provide an efficient and scalable implementation of KMI for distributed memory systems, which is capable of storing, indexing and searching hundreds of billions of DNA sequences through a set of well-defined APIs. Experiments on Bluegene/Q show the query throughput of our distributed library for 11.92 TB random strings reaches 3.79×10^8 queries/s on 65,536 cores. We also report our preliminary results on using KMI to improve the performance of a *de novo* metagenomics assembler.

1. INTRODUCTION

Ever since Watson and Crick discovered the structure of DNA in 1953 [31], scientists have been fascinated by the possibilities of what we might learn from reading our genes. For example, the Human Genome Project [30] is helping us to have a better understanding of cancer and rare genetic diseases, and leads to the birth of pharmacogenomics and gene therapy. Many current sequence analysis algorithms such as *de novo* assembly [4, 28, 34, 20], short read mapping [17, 14], and similarity search [33, 8], perform variations of k-mer matching tasks. However, this common set of k-mer matching operations is reimplemented in different algorithms with different levels of efficiency, often without much consideration for concurrency, scalability and distributed memory systems.

*This work is done when the author is visiting Argonne National Laboratory as a student research intern.

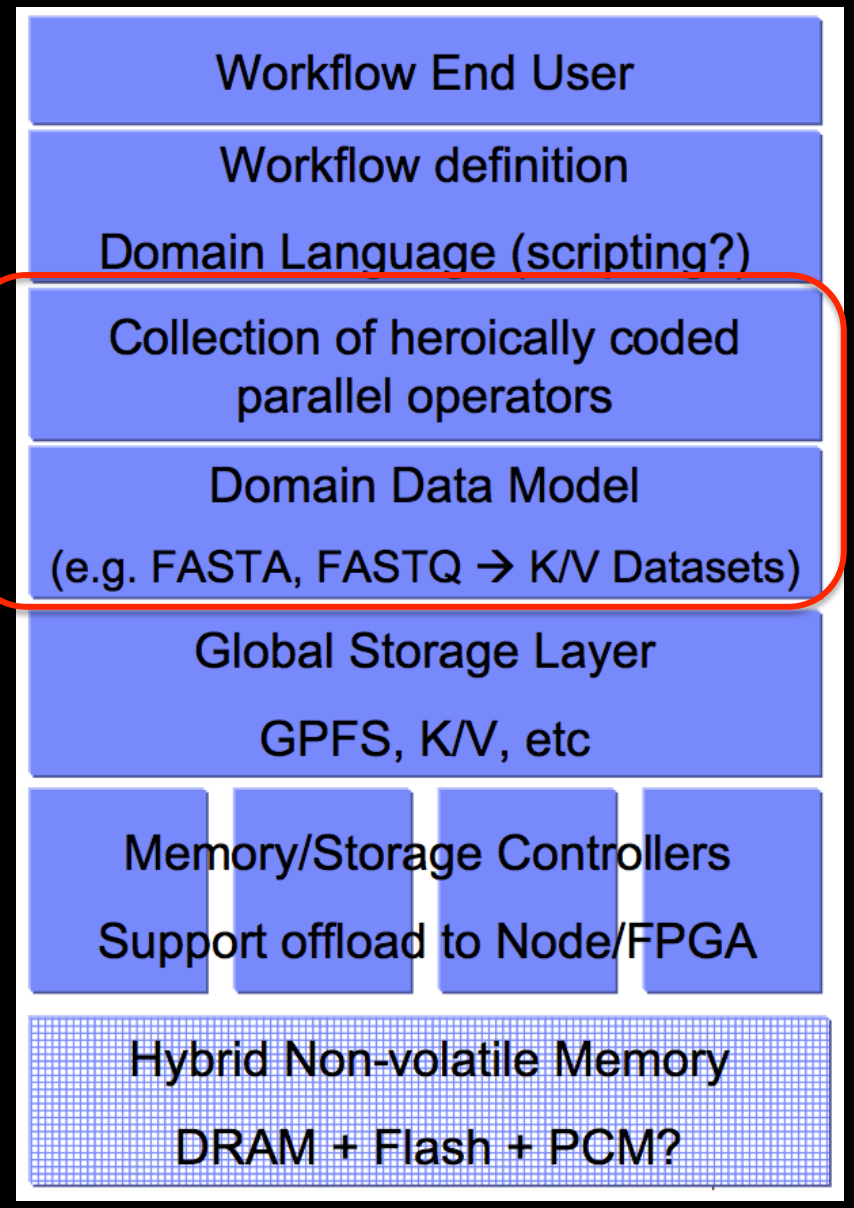
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

Ideally, bioinformatics researchers should focus on the implementation of their core sequence analysis algorithms without too much concern about the low level implementations of data management on different hardware architectures. However, lacking a common library to deal with these data management tasks, new applications need to be built from the ground, duplicating their efforts to reinvent the wheels. Considerable efforts are spent on tackling the problems such as synchronization, load balancing and race conditions in highly parallel architectures. Moreover, these problems are especially serious in metagenomics, where extreme large datasets need to distributed stored and processed. For example, the Illumina reads from the Human Microbiome Project [26] have already exceeded 10 terabytes; and the Earth Microbiome Project [9, 2] is proposing to generate about 2.4 quadrillion base pairs of sequencing data in the next three years.

Existing libraries have partially solved the efficiency, portability, and scalability problems. SeqAn [7] is an efficient and generic sequence analysis library. It is well modularized, providing well-designed algorithmic components for sequence analysis. Some parts of SeqAn are paralleled using OpenMP and CUDA. However, it is not prepared for distributed computing yet, making it incapable to tackle sequence analyses with large datasets. Ray Meta [5] is a distributed metagenome assembler that assembled 3 billion reads with 1,024 cores. It is based on RayPlatform [4], a modularized distributed runtime engine that models each process as a state machine, allowing application developers to develop plugins by registering their own functions for data communication and processing. However, in our opinion, it will be more convenient for users to use APIs to manipulate their data directly instead of developing plugins on their own.

In this paper we propose K-mer Matching Interface (KMI), a domain specific library for extreme-scale DNA sequence analysis. The goal of KMI is to extract a common set of low level operations needed for k-mer matching so that they can be implemented and optimized on a wide variety of HPC hardware architectures. A developer of existing or new sequence analysis tools can then build their applications on top of the standardized interface and achieve portability and performance without explicitly managing the indexing data structure, distributed memory and parallel communication aspects of the system. Specifically, this paper has following contributions:

- Defines a programming model for overlap computation in genome assembly and genome mapping algorithms. It establishes an interface between edge generation and edge analy-

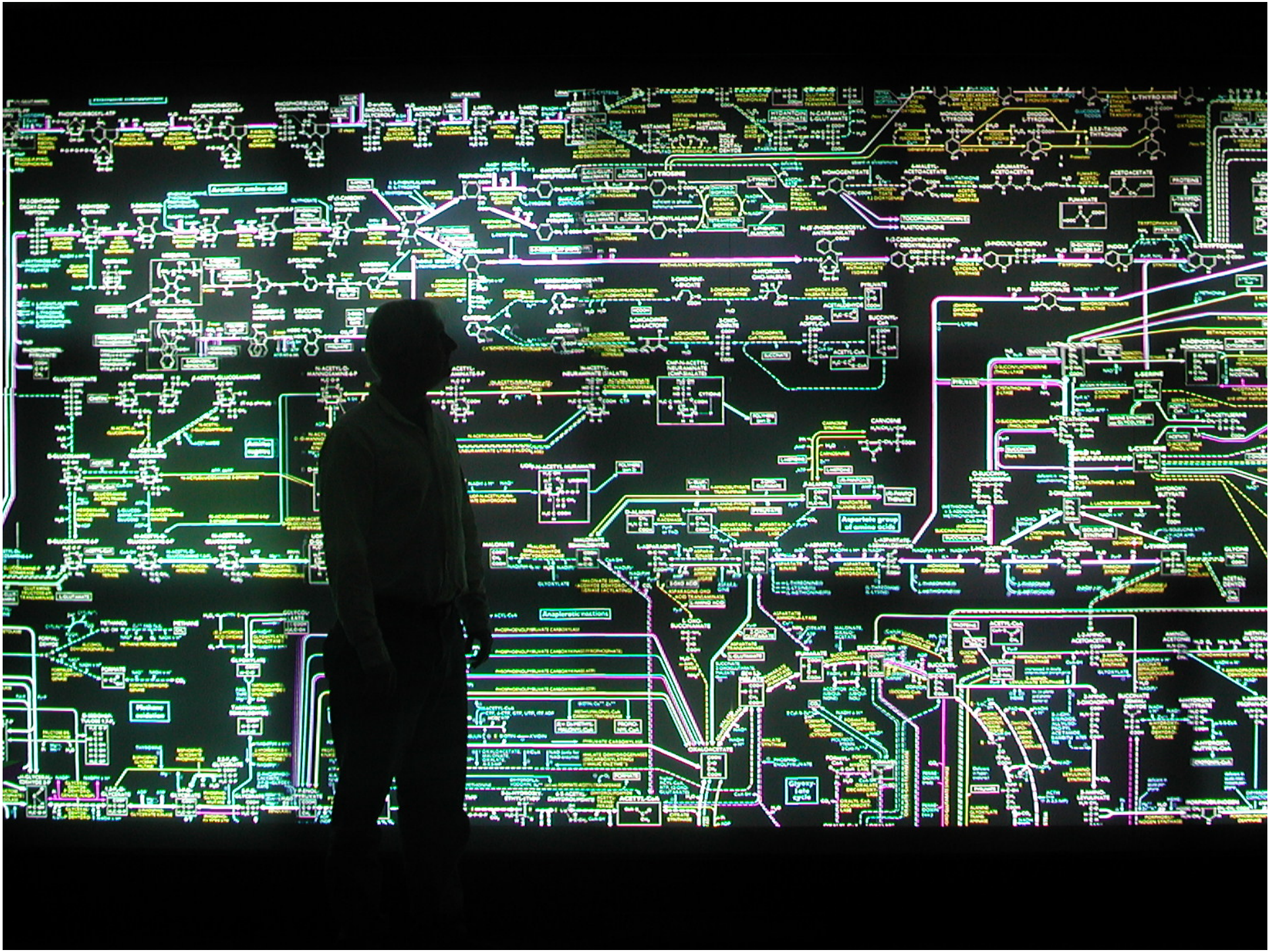


Needed OS Features

- OS support for attaching to memory regions
- OS support for timesharing sets of nodes
- OS support for slowstart/faststart
- OS support for process/storage migration
- OS support for goto store, come to proc
- OS support for neighborhoods (mem servers)

Hardware/Software Approaches

- Hardware support for nv storage on node in memory address space
- Hardware support for variety of operators against storage (hashing, indexing, search, etc.) ⇒ CAM
- Language support for data intrinsics
- Support for scripting DSLs bound to high-performance data specific libraries
- Libraries/filters for replacing explicit I/O



Big Interactivity

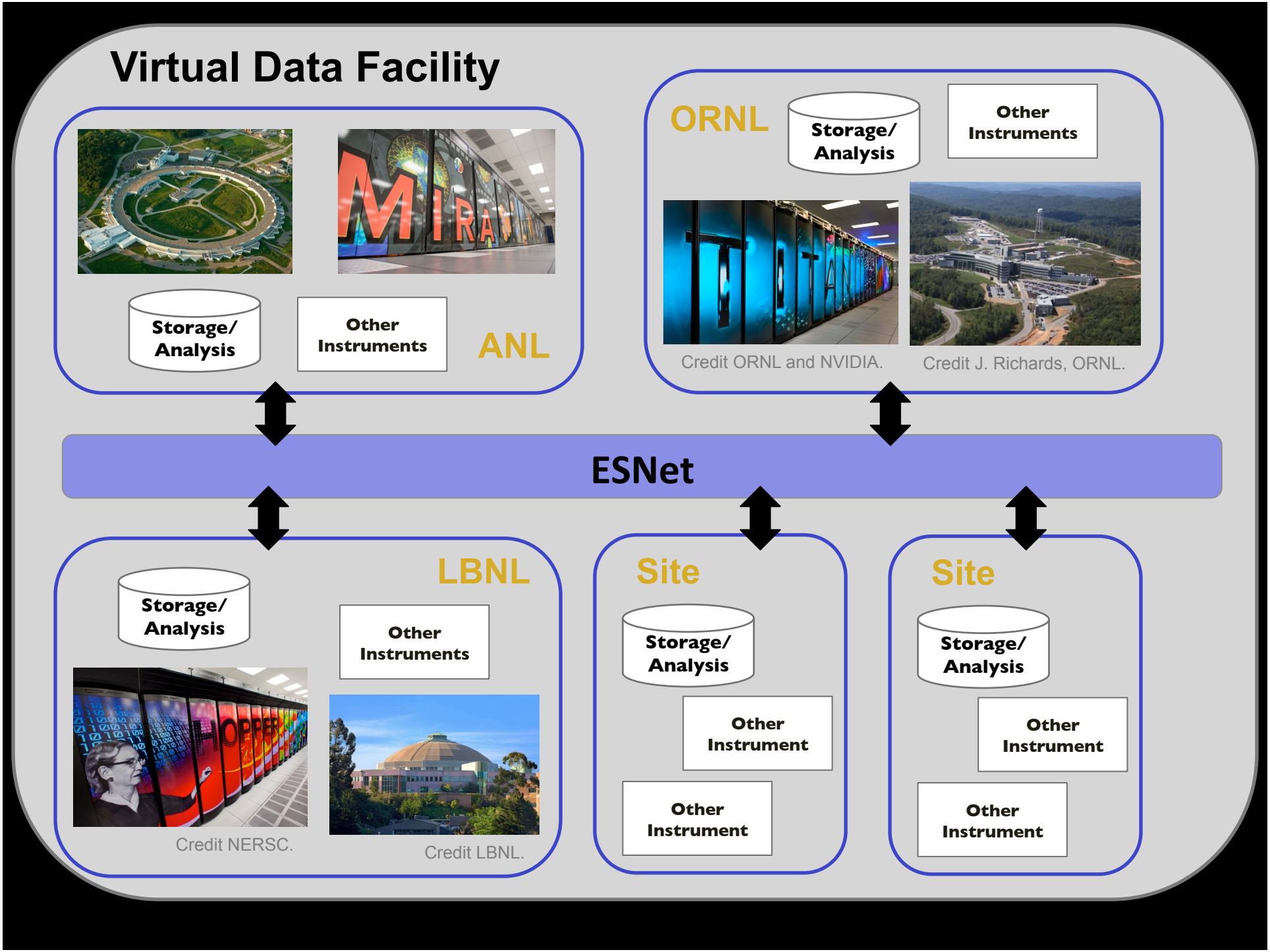
- Support for acquiring multiple I/O nodes with multiple external network connections
- Support for composing connections with outboard rendering engines, etc.
- Flexible input devices (cameras, tracking, audio, etc.)
- Support for jobs proxies in social media, interactive devices, mobile
- Capture and playback support (tutorials)
- Archive and annotate (desktop capture)
- Jobs pause forward and reverse

Argonne and Oak Ridge LCF Partnership

- Collaborative, multi-lab, DOE/SC initiative ranked top domestic priority in *Facilities for the Future of Science: A Twenty-Year Outlook.*
- Mission: Provide the computational and data science resources required to solve the most important scientific & engineering problems in the world.
- Highly competitive user allocation program (INCITE, ALCC).
- Projects receive 100x more hours than at other generally available centers.
- LCF centers partner with users to enable science & engineering breakthroughs (Liaisons, Catalysts).

Oak Ridge National Laboratory

Argonne National Laboratory


Virtual Data Facility
ANL
Storage/ Analysis
Other Instruments
ORNL
Storage/ Analysis
Other Instruments
Credit ORNL and NVIDIA.
Credit J. Richards, ORNL.
ESNet
LBNL
Storage/ Analysis
Other Instruments
Credit NERSC.
Credit LBNL.
Site
Storage/ Analysis
Other Instrument
Other Instrument
Site
Storage/ Analysis
Other Instrument
Other Instrument