Big Data, Big Compute, Big Interaction Machines for Future Biology

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

BD Usage Models Differ from EC

Big Data

- Continuous access require based on data generation/ submission rates
- CPU time, I/O and data volume all important
- Data products typically used in future computations via an integration or pipeline
- Data products made available for external users and curated over time

Extreme Compute

- Batch oriented access based on allocations for specific projects
- Mostly CPU time centric
- Output not necessarily used in future runs but often significant time used for visualization
- Output generally (but not always) used “privately” and rarely curated

Policies Need to be Different

- Long term (many years) access commitment at a continuous or increasing level of service
- Support for persistent services
- Storage allocation that grows over time
- Rich software environment with high-performance database support
- Mechanism to publish the data to a community
- Archival support for data, links and citations

Convergence

- Ideal Environment
  - 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

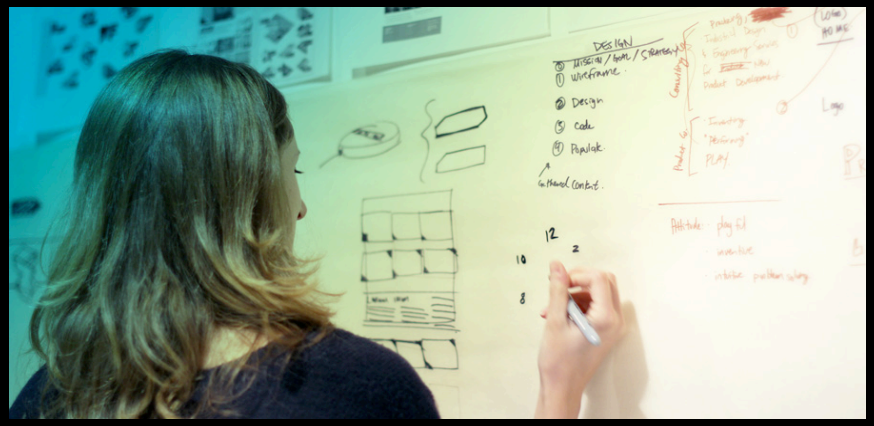

MIRA


DataDirect Networks
CP

Magellan: Our OpenStack Private Cloud for Systems Biology

What Users Want To Do

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

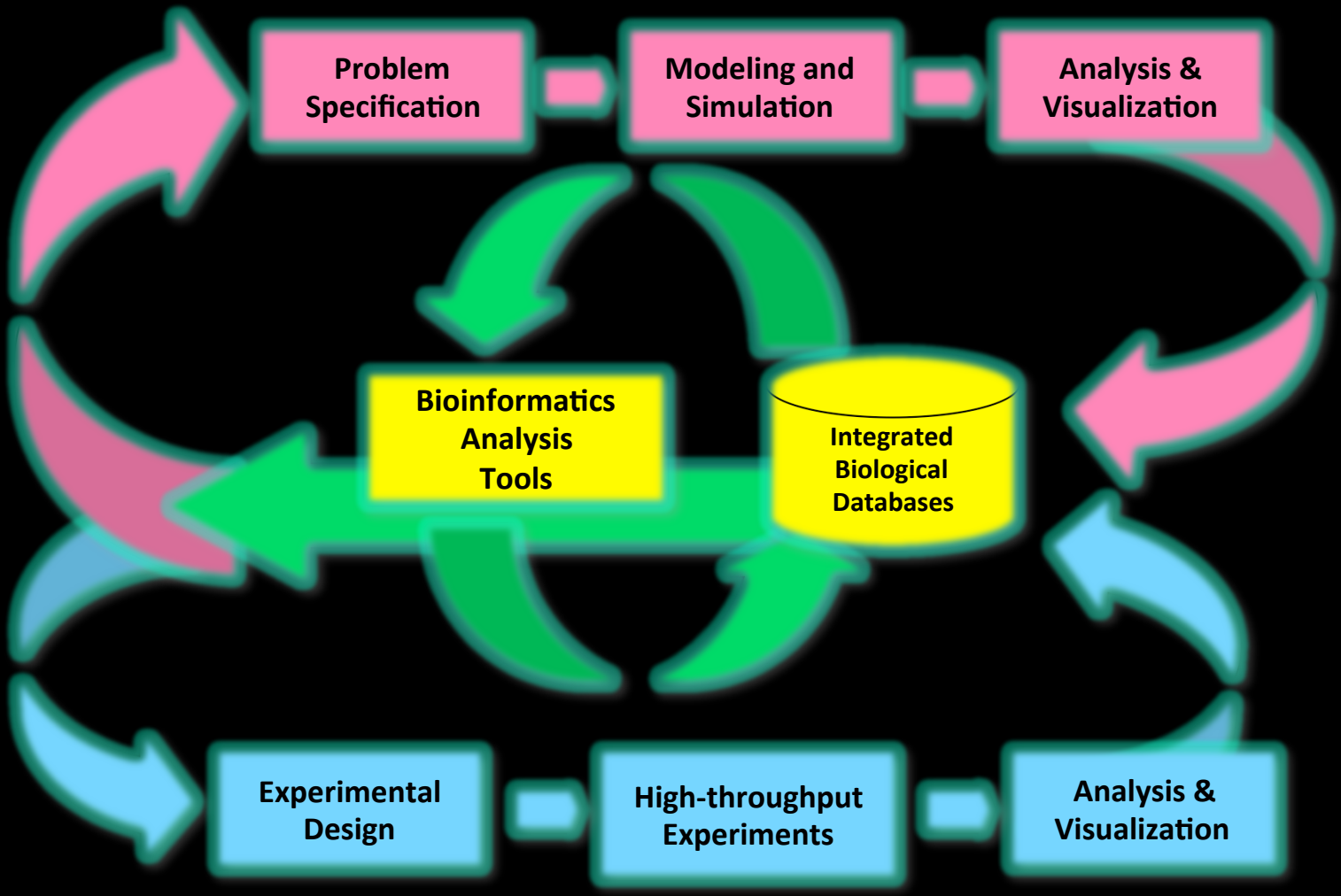


What do we want to do with Data?

- Generate
- Process
- Analyze
- Annotate
- Visualize
- Understand
- Share
- Publish
- Curate
- Archive
- Integrate
- Move
- Search
- Preserve
- Model
- Compare

GREEN is Interactive

Converging View of Modeling, Simulation, Experiment, Data and Bioinformatics



Big Data Challenges for Bioinformatics

- New types of methods and new algorithms
  - From O(N3) ⇒ O(N^2) ⇒ O(N log N) ⇒ O(N) ⇒ O(K)
  - Non-alignment methods and streaming
- New types of Infrastructure bringing biological data and computing together
  - Users need to have an environment where they don't need to move the data to work
- Ability to share methods, protocols, tools and insights leveraging social networks
  - Enable the best methods to win regardless of where they come from

Sequencing the Environment

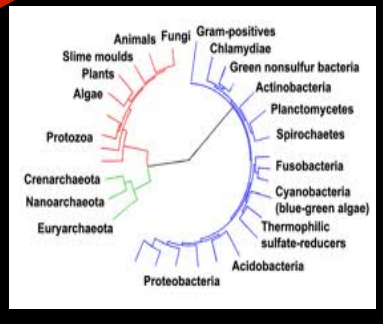
Metagenomic data collection

llecting
samples

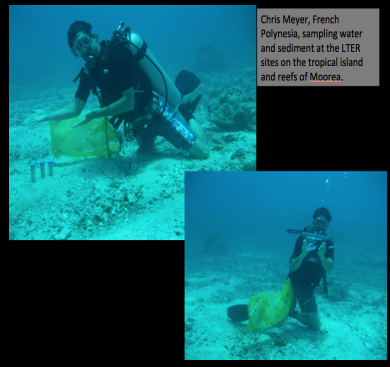
Sequencing

Sequence
fragments

Associating fragments
to taxonomical groups

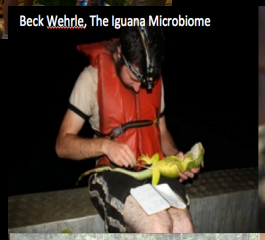


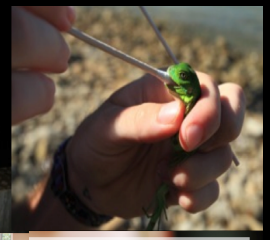
Assembly of most abundant microbes into complete genomes

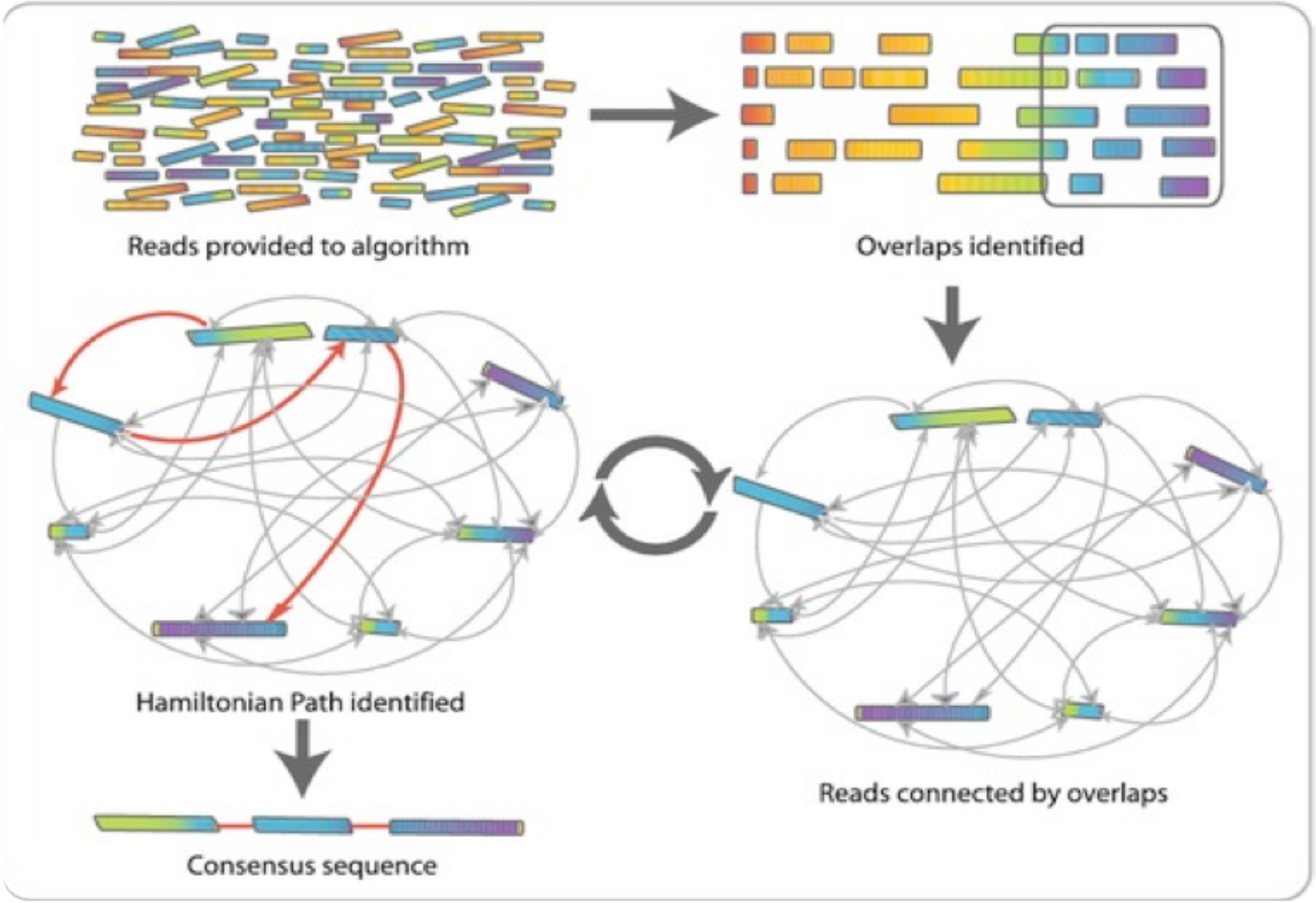


Extreme environments:
Acidic hot springs, Yellowstone–
contributed by Greg Caparoso

Merlot Microbiome:
High school volunteers
Long Island





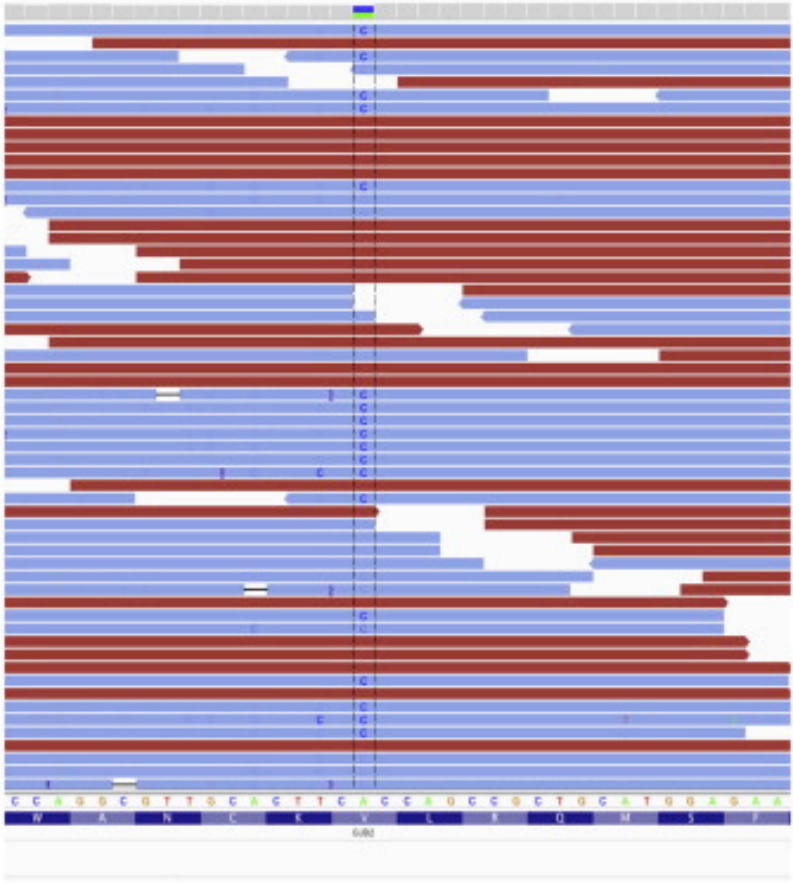

Reads provided to algorithm
Overlaps identified
Hamiltonian Path identified
Reads connected by overlaps
Consensus sequence

a Illumina HiSeq
Generate Millions of Sequencing Reads

```
@HWI-ST898_0069:1:1101:1153:2040#TAGGC/2
GACGATTCATTCACATGTTCCACCTCTTGTTTCTGTGAGGCATTGATATCCACTCTCATATCATCAAGAAAATGTTACTAGGAAAATTTTTAGAAAAGCA
+HWI-ST898_0069:1:1101:1153:2040#TAGGC/2
9999999999999999999999999999999999999999\9999999999999999egedgeeegggcgggecgggdfeegeedfgeeegggeadafgeged
@HWI-ST898_0069:1:1101:1206:2067#TAGGC/2
ATGCAAAAAAGTACATCTTGTCTCTTCCTCTAGGTGTGTCTGCACAGCTCACCAGTACCCGGCACCGTCGGAACACATCCGTAGGAACACCGTTTTGGAT
+HWI-ST898_0069:1:1101:1206:2067#TAGGC/2
fffdfeffffdeeeaefgcegggfgggggegdffecedcdfeeee_^___`U`VcX``Y[_addddedfbd\cc\cdacaebb\acT]]][]^[Y__`_c\`
@HWI-ST898_0069:1:1101:1216:2082#TAGGC/2
TCATTCTCCACGTTCTGAGATGTATGCCACCTGCCTTAGTTACCACAAGCAACTAACACTAAGCTGCTAGCAGGAATCTGTCTGCCCTCTATCTGTCGCT
+HWI-ST898_0069:1:1101:1216:2082#TAGGC/2
gggggggggggggggggdafffcfdggdgfgggggggggggagfffdfdedefdgg_gd_ecfeddcfggeggcgggedgeefda_dfgceefggged
@HWI-ST898_0069:1:1101:1459:2015#TAGGC/2
CCTGATGGCAGGGGTGGGGGACACTCACACAGTCCTGTAGTACCAGTTCATGAGGACAAAGAGCATGGCGGCCAGGAACAGGAGGATAGCCAGGACGATG
+HWI-ST898_0069:1:1101:1459:2015#TAGGC/2
ggggggdgggffhgXbddedaceeegegge_bccfdd7edadbeee`eedfedfffcgffgggggdfgggdgeed7adcfcbbd\]TRY^]`][RBUTR
@HWI-ST898_0069:1:1101:1421:2025#TAGGC/2
AAACGTGAGCAGGAGGTGAACATCCTGAAGAAGACCCTGGAGGAGGAGGCCAAGACCCACGAGGCCCAGATCCAGGAGATGAGGCAGAAGCACTCACAGG
+HWI-ST898_0069:1:1101:1421:2025#TAGGC/2
```

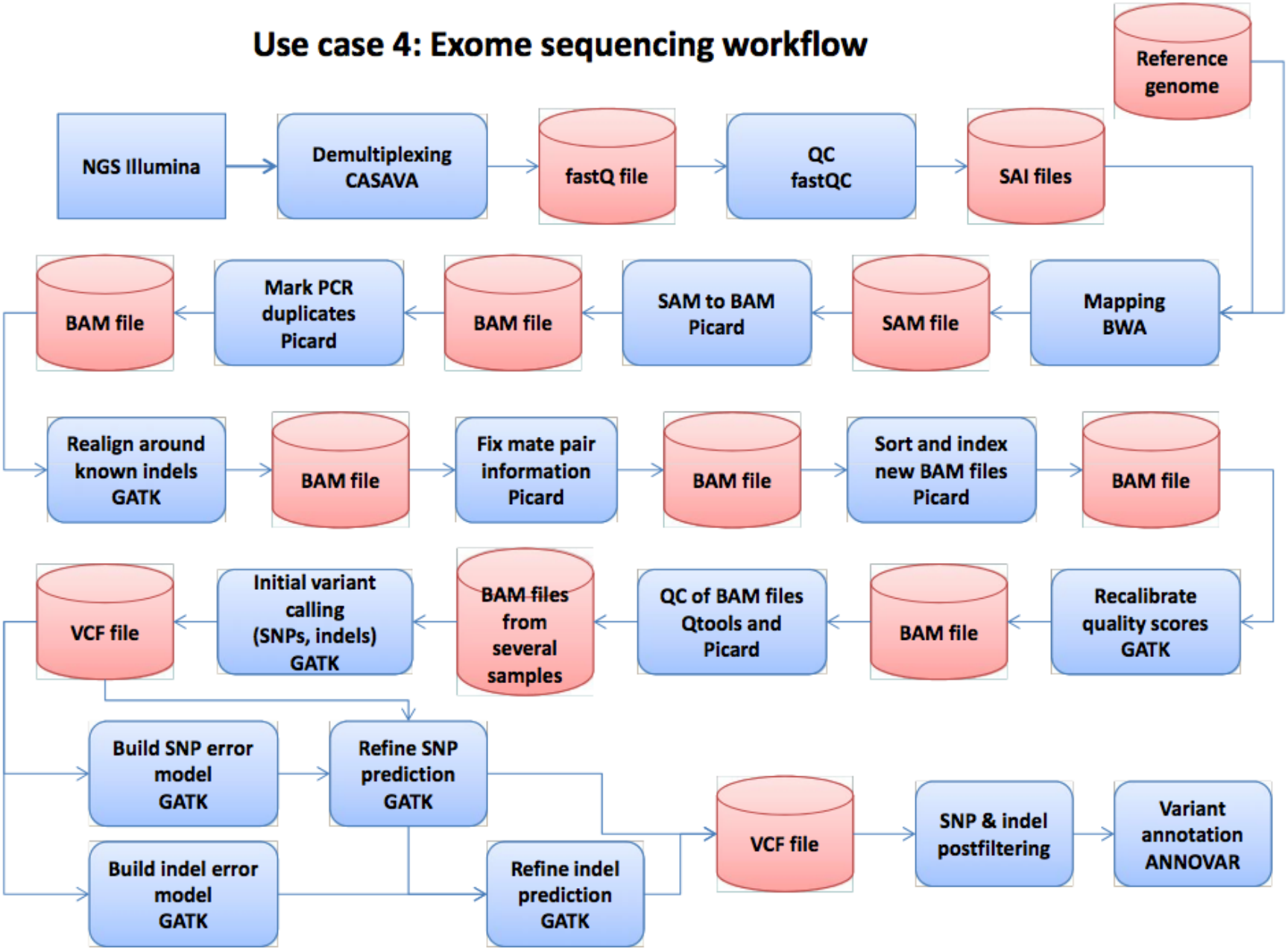
b Raw Sequencing Read Output

c Align Millions of Sequencing Reads to the Reference Genome

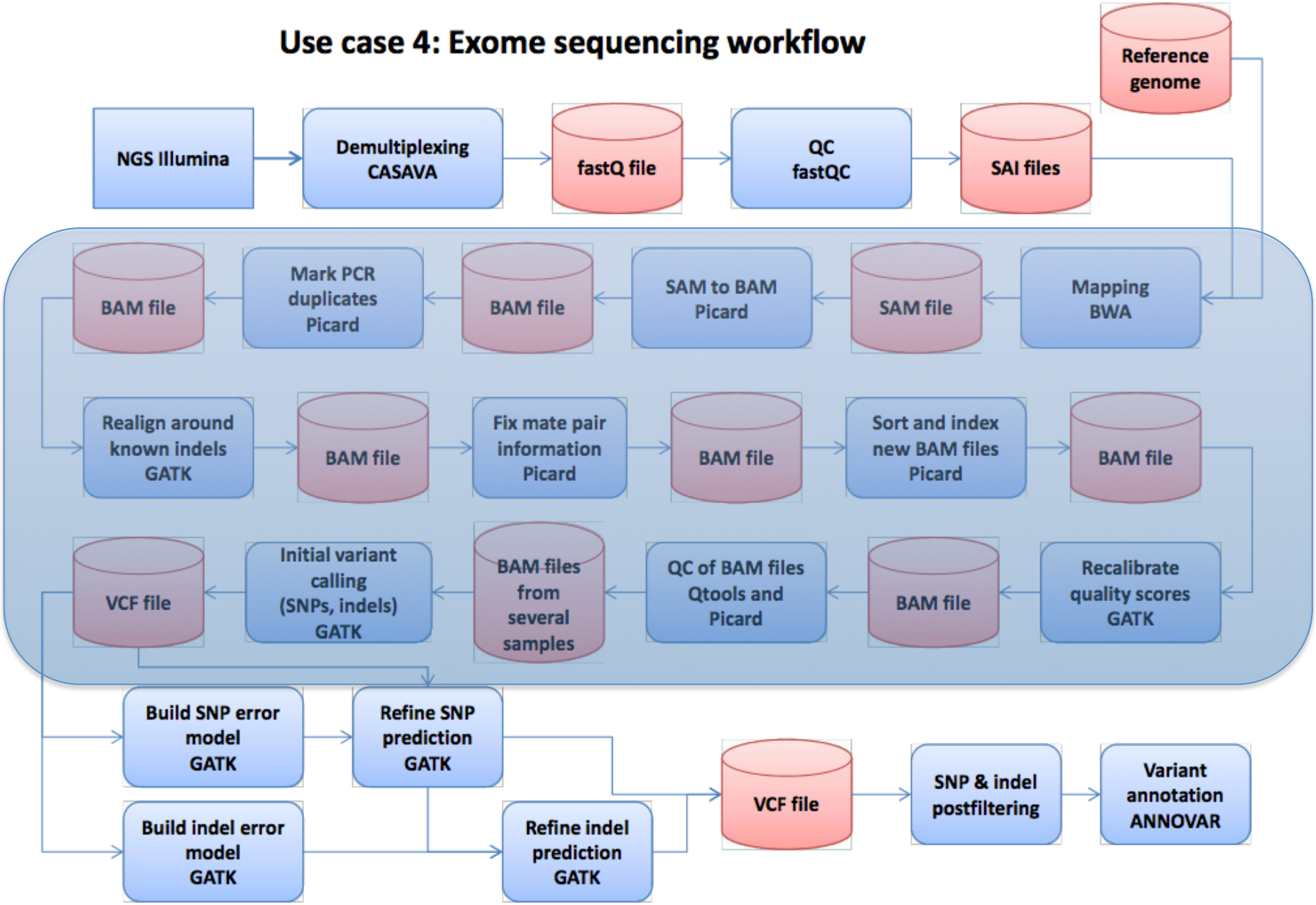


d Identify and Annotate Variant:
GJB2 Val167Met carrier

Use case 4: Exome sequencing workflow



Use case 4: Exome sequencing workflow



Data-centric Computing Using BG/Q Active Storage

IBM

High Compute Density

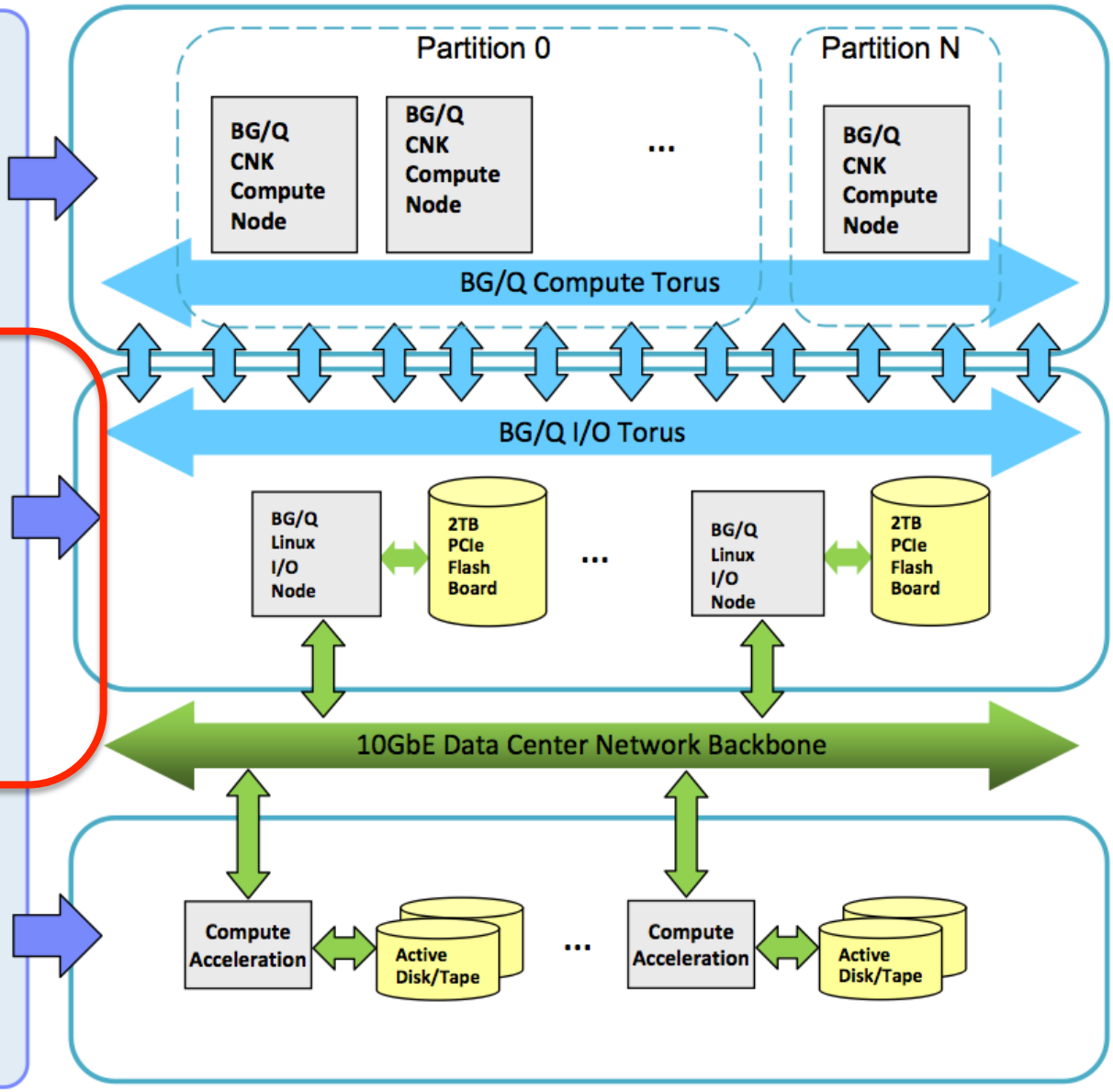
- BG/Q compute fabric 1k – 100k nodes
- DRAM memory
- 5D Compute Node Torus
- CNK, ZeptoOS, FuseOS
- I/O Links to D1 Layer (4:1 ratio)

Active Storage

- 8 – 4096 Linux BG/Q I/O Nodes
- DRAM + 2TB SLC Flash per Node
- 2 GBps bandwidth to storage
- GPFS / KV services
- I/O links to each node (4GBps/node)
- All to All Comm. Via I/O Torus
- DB2, Infosphere Streams, Hadoop, MVAPICH, SLURM

Data Center Storage

- GPFS file system
- External Disk Controller Racks



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.

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-

Workflow End User

Workflow definition

Domain Language (scripting?)

Collection of heroically coded parallel operators

Domain Data Model

(e.g. FASTA, FASTQ → K/V Datasets)

Global Storage Layer

GPFS, K/V, etc

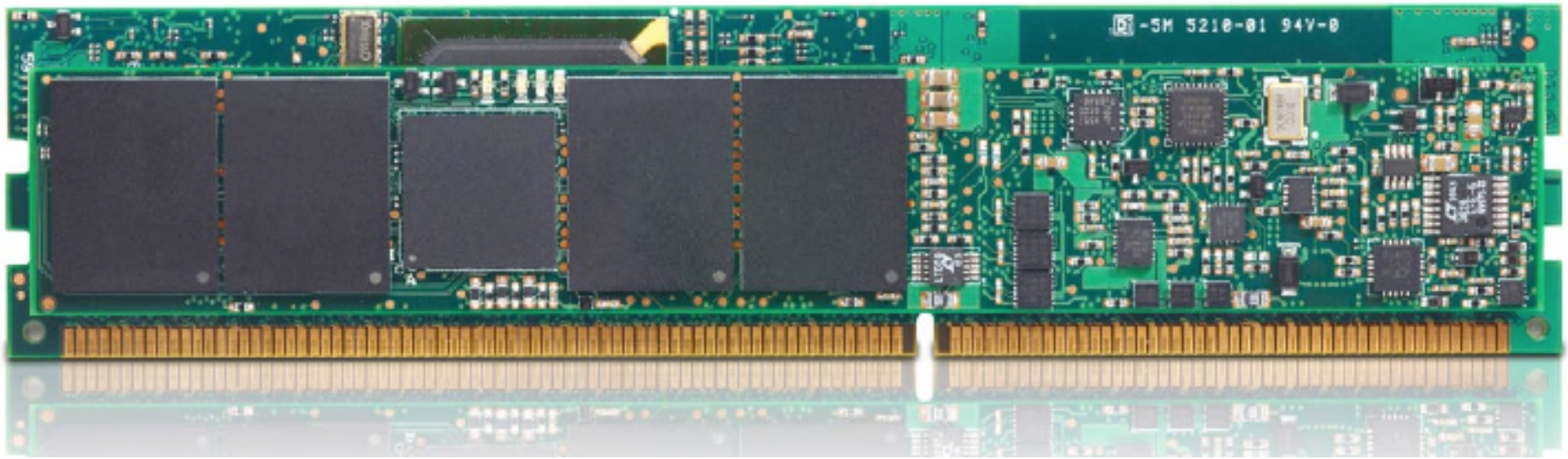
Memory/Storage Controllers

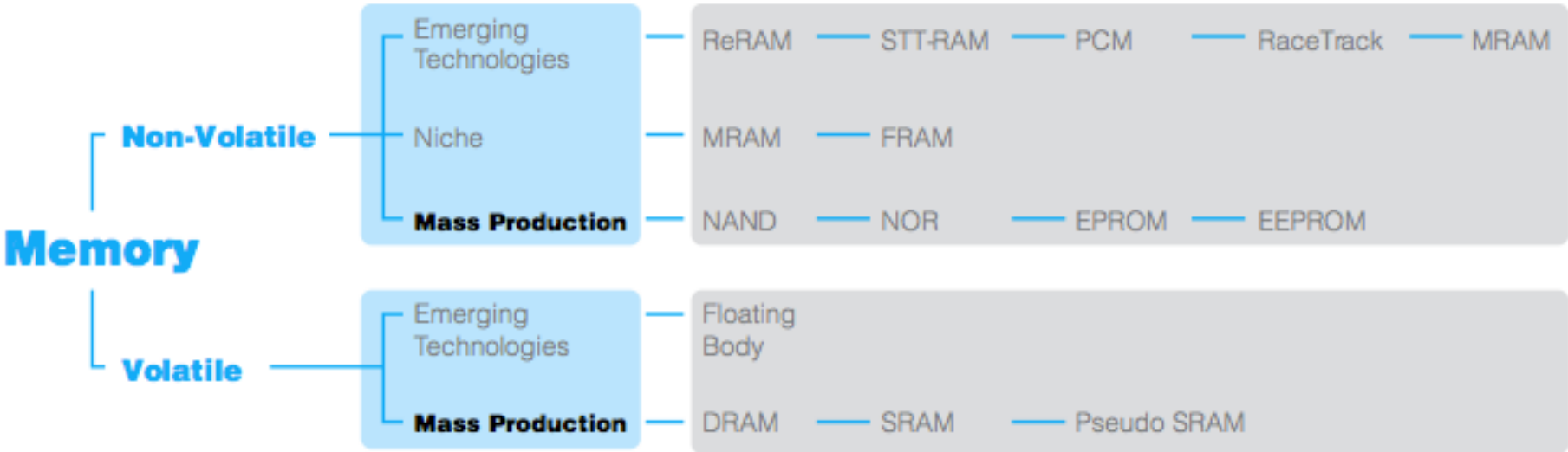
Support offload to Node/FPGA

Hybrid Non-volatile Memory

DRAM + Flash + PCM?

NVDIMM




Memory
Non-Volatile
Emerging Technologies
ReRAM
STT-RAM
PCM
RaceTrack
MRAM
Niche
MRAM
FRAM
Mass Production
NAND
NOR
EPROM
EEPROM
Volatile
Emerging Technologies
Floating Body
Mass Production
DRAM
SRAM
Pseudo SRAM

Convergence Features

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

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 provisioing

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

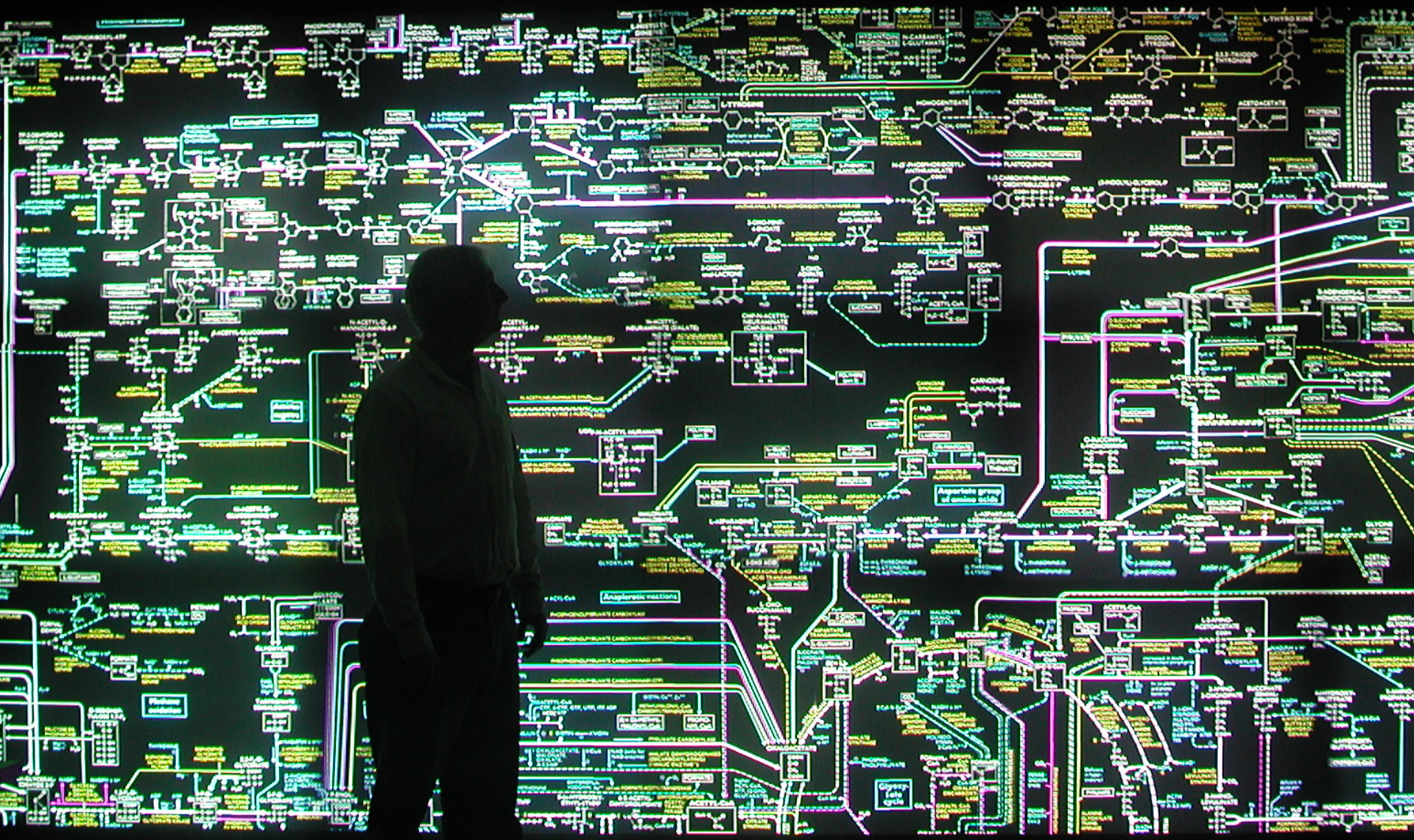
Bandwidth per DRAM byte is constant

Needed Convergence 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

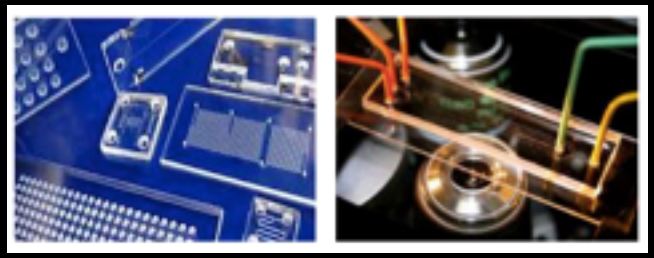
- Support for acquiring multiple I/O nodes with multiple external network connections
- Support for composing connections with outboard rendering engines, etc.
- Flexible input devices (cameras, tracking, audio, etc.)
- Support for jobs proxies in social media, interactive devices, mobile
- Capture and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

Research Areas: New Algorithms, Software and Hardware Architectures Needed

- Sequence mapping, assembly, alignment, clustering
- Pattern and feature matching and discovery in complex data models
- Domain specific data compression methods sequence, vector spaces
- Error detection and correction methods in sequence, vector spaces
- Heuristic search over complex data models
- Constraint based methods for fitting, mixed/integer linear programming
- Text indexing, search, query methods
- (alg, hw) Sequence assembly and characterization (hw) Pattern matching architectural support
- (alg, sw) Approximate matching methods for patterns in complex data models
- (sw) Workflow infrastructure for parallel systems and cloud based services
- (sw) Interactive workspaces and rapid prototyping environment with DSLs and in memory database

Blue Sky – things that would make a big difference

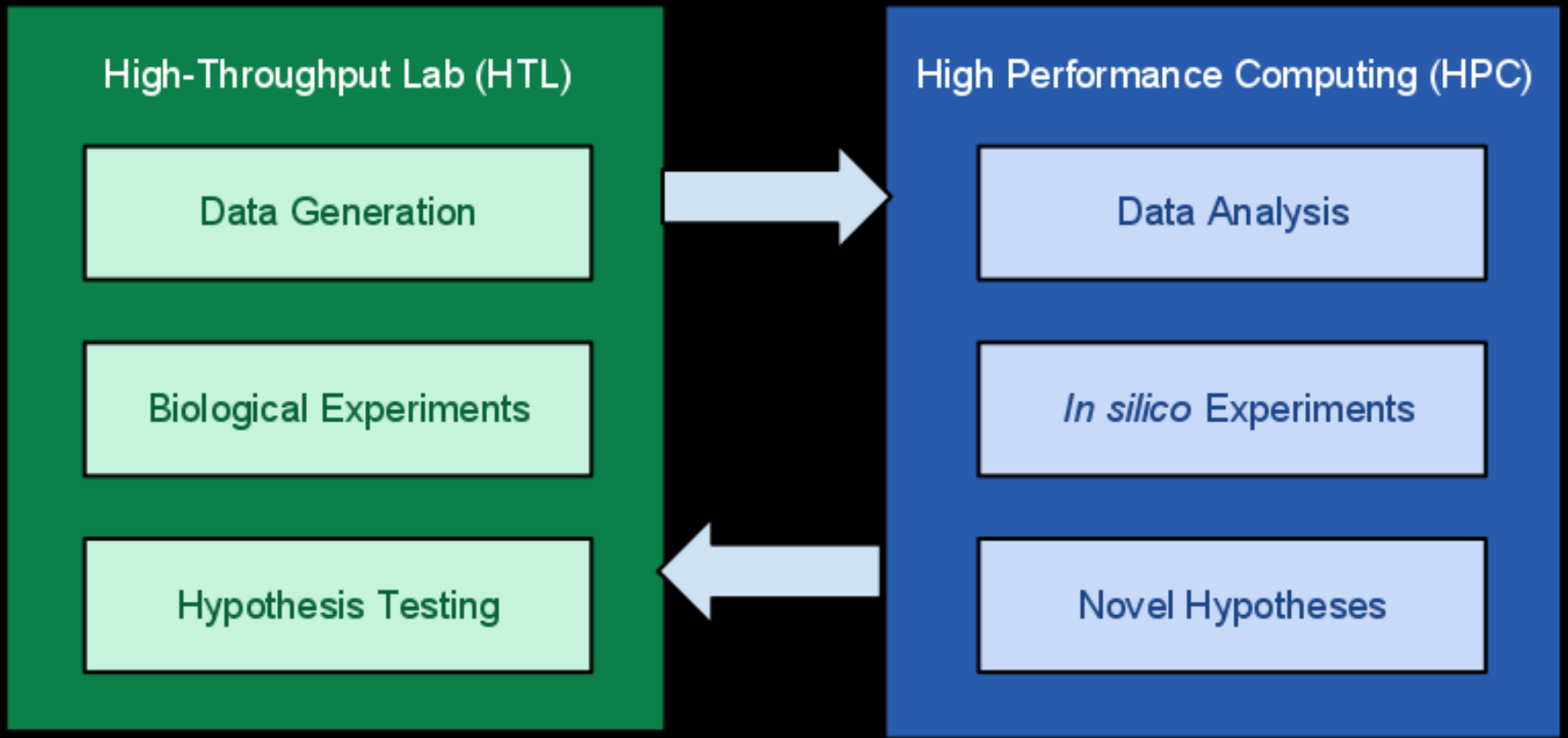
- Novel hardware for key algorithms
  - Match analysis speeds with data generation
- DSL for programming microfluidics devices
  - Accelerate wet lab data generation
- Next Generation Interactive systems
  - [1000 cores, 10TB flat RAM and 1 PB Store]
  - Enable interactive analysis of large bio datasets for less than $100K.



Convergence

- Ideal Environment
  - 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

Automate and Accelerate




30.6
93.5
0.10
20.0
IX