Messages, keys and values in genomics

Sébastien Boisvert

@sebhtml

Ph.D. Student, Laval University

15 minutes

Funded by:

CIHR IRSC

Canadian Institutes of Health Research

Instituts de recherche en santé du Canada

BlueGene Active Storage (BGAS) Workshop for Medical Sciences (and beyond):

High-Performance I/O and Scalable Data-centric Analytics

2013-08-07 9:00 --, SciNet, University of Toronto

UNIVERSITÉ LAVAL

SciNet

Calcul Québec

compute • calcul CANADA

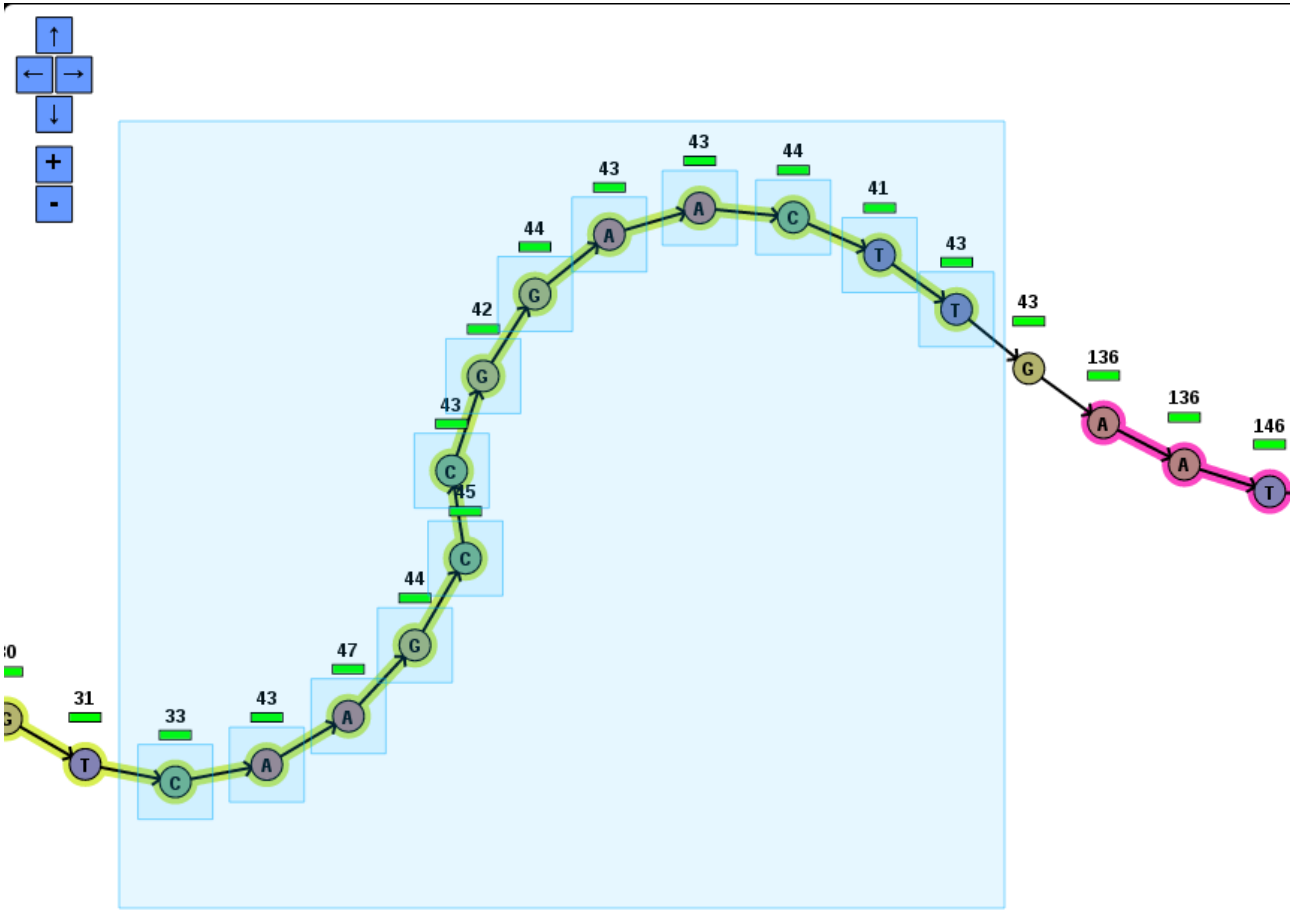
Workflows in genomics

- Alignment + variation calling
- Alignment + gene profiling
- De novo genome assembly

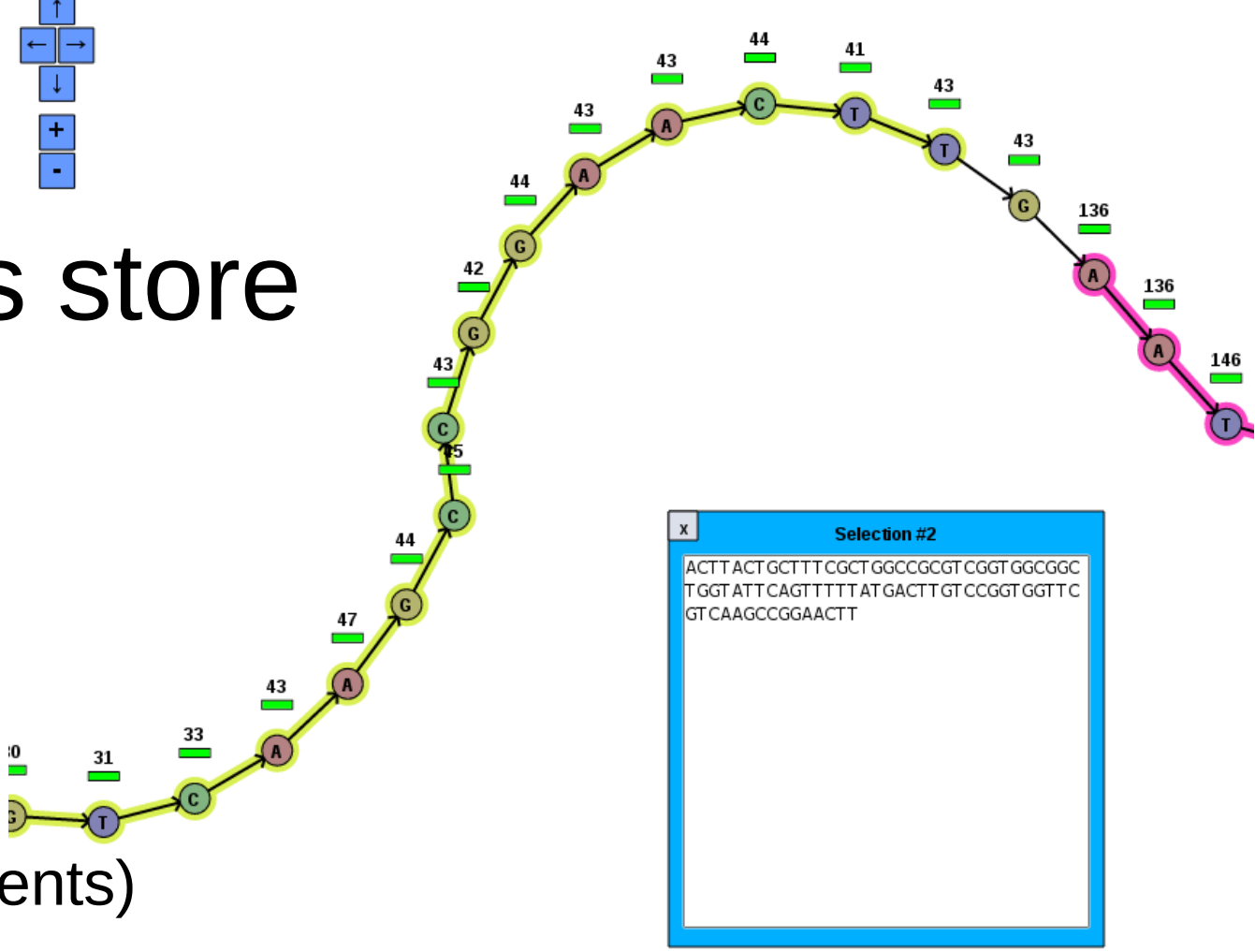
- Trends: RNA-Seq, metagenomics
- Applications of next-generation sequencing http://www.nature.com/nrg/series/nextgeneration/index.html

Processes and messages

- In genomics:
  - protein sequences
  - DNA sequences
  - Annotations
  - abundance profiles
  - Variations
  - Alignments
  - assemblies



How biologists store data ?



- Files:
  - FastQ (reads)
  - SAM / BAM (alignments)
  - VCF / BCF (variations)
  - Fasta (sequences)
- Meta-data: Laboratory Information Management Systems (usually backed by SQL)
- Also: no meta-data or “my dog ate my meta-data” !

Sequence retrieval

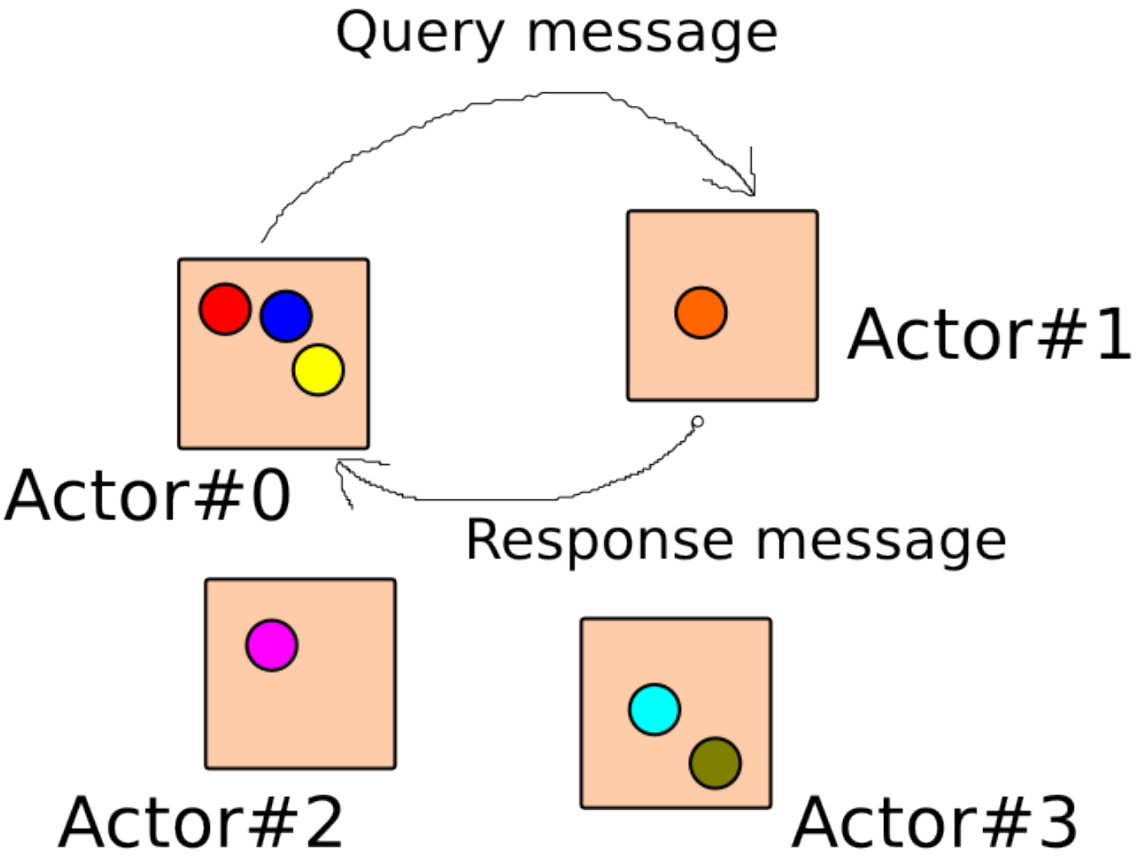
- FastQ can't provide random access to sequences
- Genome assemblers usually use in-memory storage for reads, distributed or not

Cloud genomics

- DNAnexus (Amazon EC2 + Amazon S3) and others (http://dskernel.blogspot.ca/2013/06/a-survey-of-burgeoning-industry-of.html)
- Idea: use NoSQL / key-value stores instead of files
- Can we reuse these ideas in high performance computing for genomics ?
- Possible benefits:
  - speed up existing tools,
  - More scalable, distributed,
  - easier to program than “roll your own”

Distributed data and computation

- **Multiple Instruction, Multiple Data**
  - Multiple instruction streams, multiple data stream
- **Single Program, Multiple Data**
  - One executable provides instructions for all actors / ranks / processes



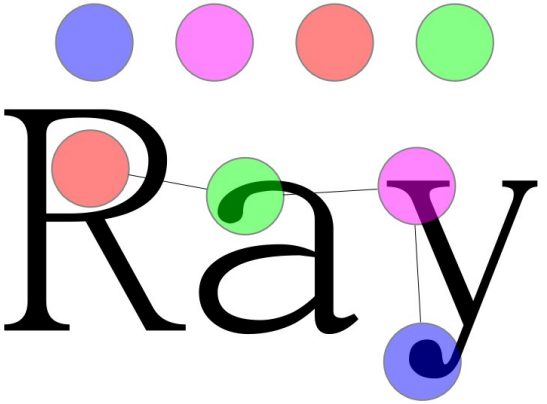
Distributed de novo assemblers

- Ray (Laval University)
- ABySS (Canada's Michael Smith Genome Science Centre) Jared Simpson is in the audience !!!
- Kiki (Argonne / University of Chicago)
- Ray and ABySS => distributed de Bruijn graph

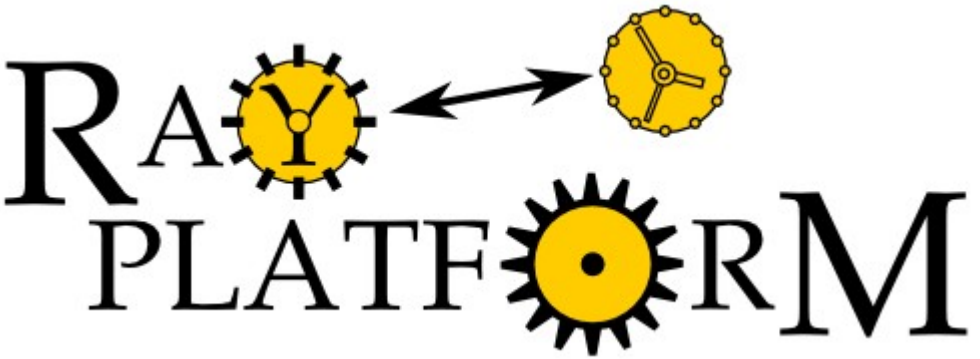
Genomic data storage

- Kiki: greedy with read overlap
- Ray and AbySS use distributed hash table
- Kiki has storage actors and computation actors (Users, Dealers, Farmers)

Ray



- GPL
- Single executable
- Built on top of RayPlatform (LGPL) framework
- C++ 1998 / MPI 1.0
- Portable (Cray XE6, Blue Gene/Q, iDataPlex, Sun Constellation, laptop, Raspberry Pi, Amazon EC2, ...)
- Checkpointing



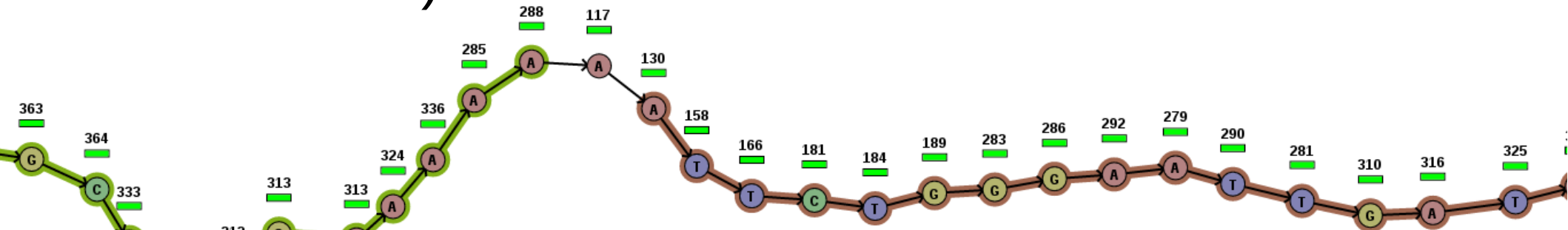
Ray papers

- Metagenomics with Ray: http://genomebiology.com/2012/13/12/R122
- Ray for large genomes: http://www.gigasciencejournal.com/content/2/1/10/abstract
- Ray for bacterial genomes: http://online.liebertpub.com/doi/abs/10.1089/cmb.2009.0238

Objects in Ray

- Short DNA sequences (input reads)
- Very short DNA sequences (vertices)
- Long DNA sequences (graph paths)

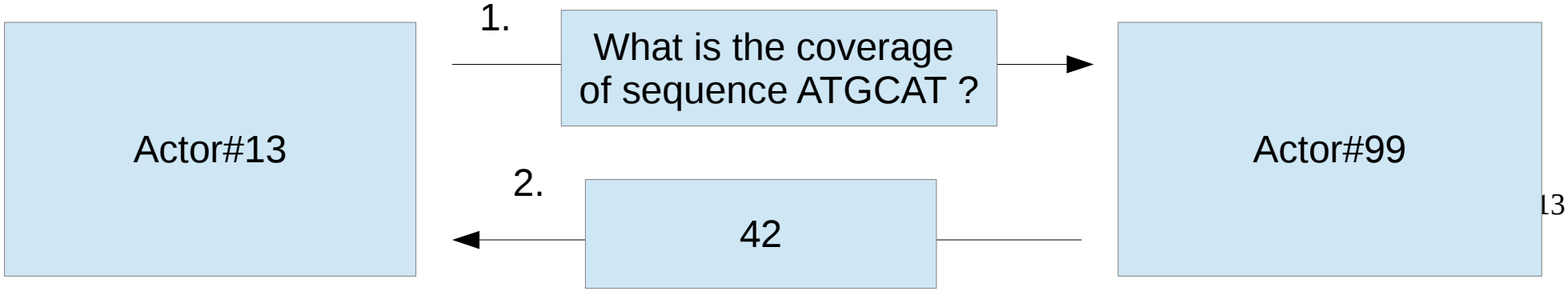
- Vertices: stored in a in-memory distributed hash table, accessible using messages (MPI)
- Note: each of these object types have custom storage code (needs more programming than key-value store)



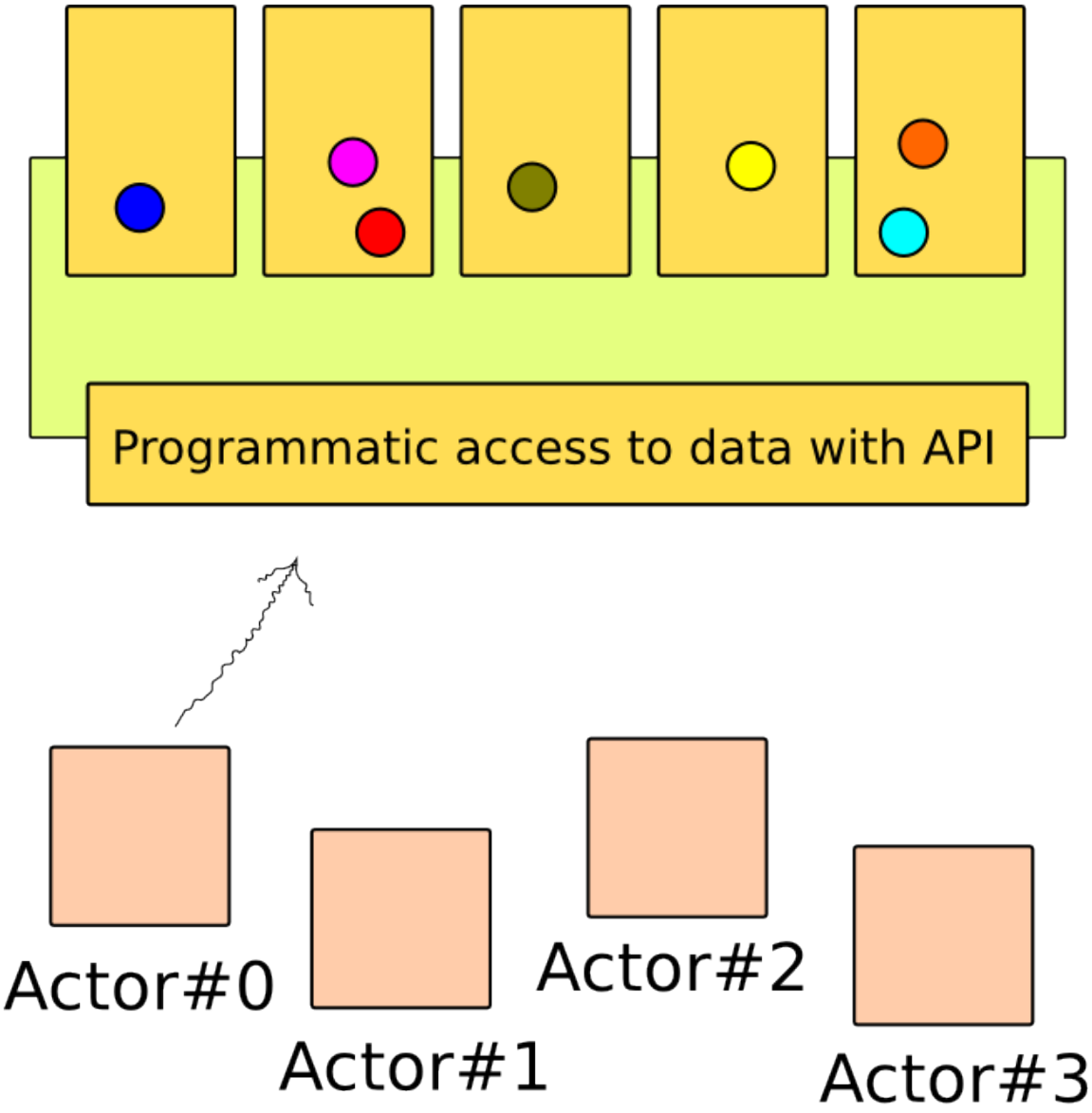
Ray

- Huge distributed graph
- Ray's messaging API allows:

- Add vertex
- Get vertex (sequencing depth, parents, children)
- And more (operations on reads, and so on)



Key-value store



Key-value store

- Usually CRUD (Create, Read, Update, Delete):
- Insert(Key, Value)
- Get(Key) **returns a Value**
- Delete(Key)
- Update(Key, Value)

Programming interfaces

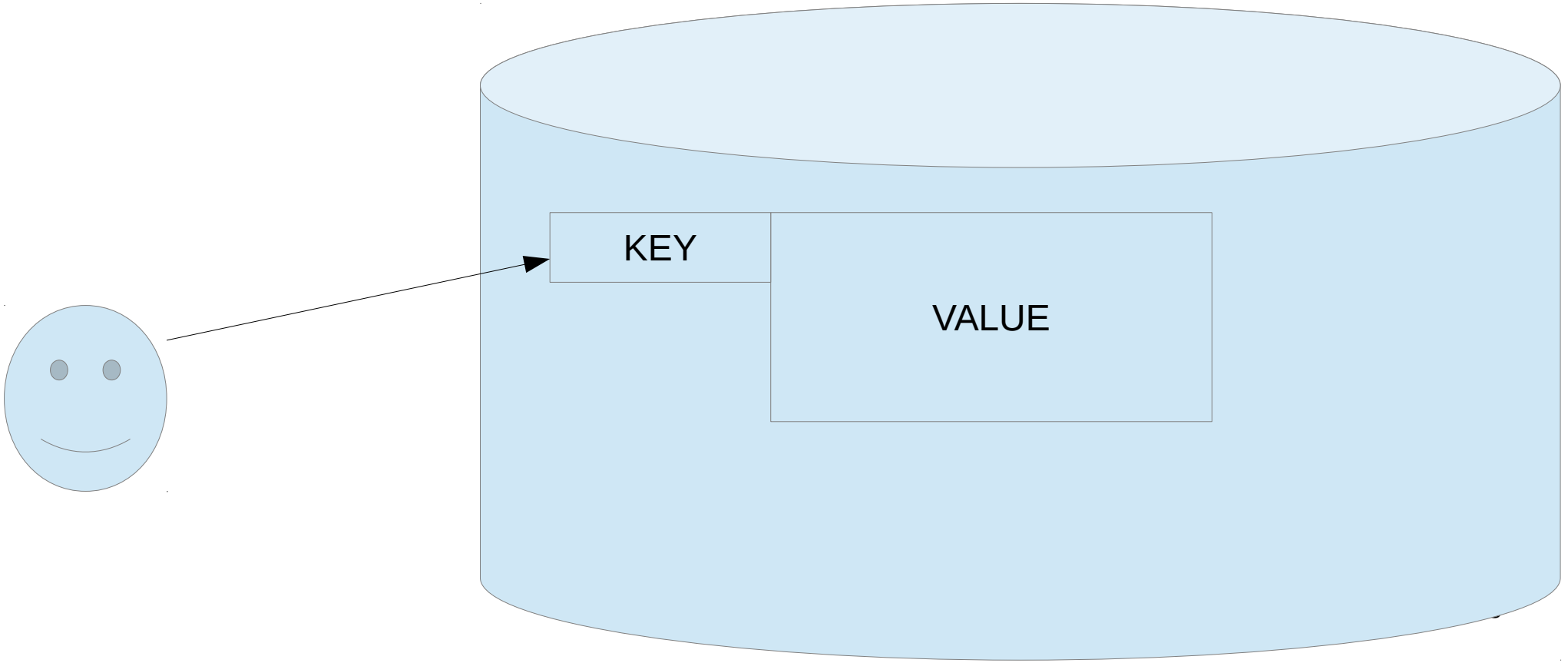
- synchronous or asynchronous
- Data stored/retrieved with API
- Keys can usually be grouped in tables / collections / sets / containers
- A key has a preference list (list of nodes that can handle it) for durability **(implementation)**

Key-value stores for commodity hardware

- Cassandra (Facebook) http://dl.acm.org/citation.cfm?id=1773922
- Dynamo (Amazon.com) http://dl.acm.org/citation.cfm?doid=1323293.1294281
- Redis
- Memcached
- (and others)

Key-value stores in the cloud

- Amazon Simple Storage Service (S3)
- Windows Azure Blob Storage
  http://dl.acm.org/citation.cfm?id=2043571



Key-value stores in high-performance computing

- Need extra-low latency (microseconds)
- Asynchronous
- Parallel In-Memory Database or PIMD (IBM)
- Kelpie (Craig Ulmer @ Sandia, Trilinos project)
- Memcached
- Redis
- Key-value stores in HPC (Wang et al. SC13)

Parallel In-memory Database (IBM)

- http://domino.research.ibm.com/library/cyberdig.nsf/papers/221880A3625C4D4A8525770700557EFE
- Paper: Fitch et al. PDSW 2009
  - http://dl.acm.org/citation.cfm?id=1713086
- C++ API (version 0.1, 2010)
- Distributed in-memory key-value store
- Can dump/load image
- Runs on top of Blue Gene Active Storage (Active Storage Fabric)

Key-value stores and Ray

- Things that could be ported to PIMD:
  - Reads (maybe too small to be efficient ?)
  - Vertices (latency for very small objects ?)
  - Checkpoint images/blobs (probably best low hanging fruit)

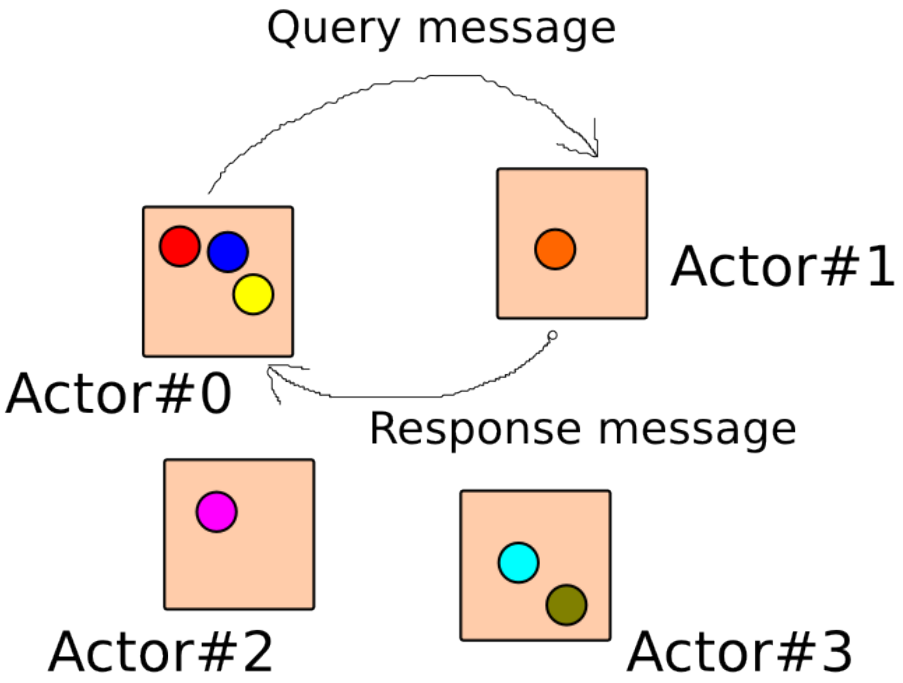
Ray on Blue Gene/Q

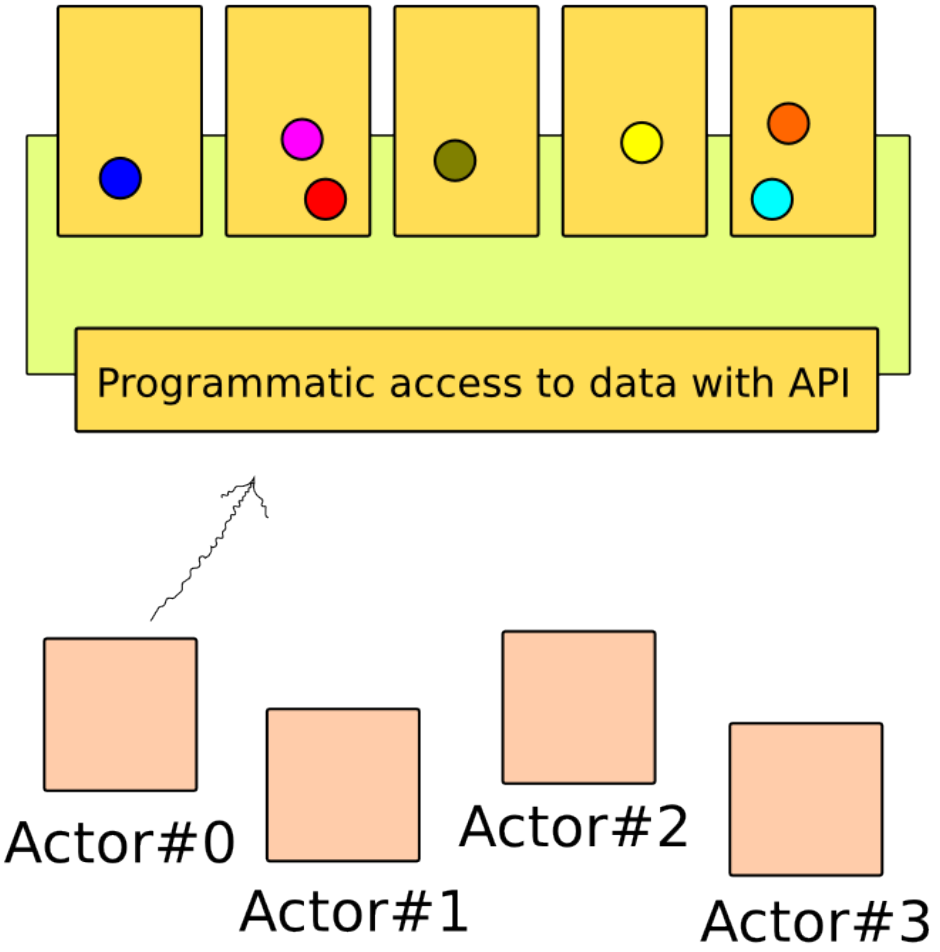
- White spruce (20 Gb huge genome)
- 8.5 billions DNA reads
- ABySS paper for white spruce assembly on commodity cluster:
  http://bioinformatics.oxfordjournals.org/content/29/12/1492.full
- Ray on Blue Gene/Q: used 1024 BG nodes, 4 ranks per node
- 8.5*10^9 reads, about 2 days
- PIMD could be used to store checkpoints instead of GPFS

Acknowledgements

- Daniel Gruner (SciNet) for inviting me to this workshop and also for beta access to Blue Gene/Q
- SciNet for covering travel and lodging costs
- Canadian Institutes of Health Research (scholarship)

Questions

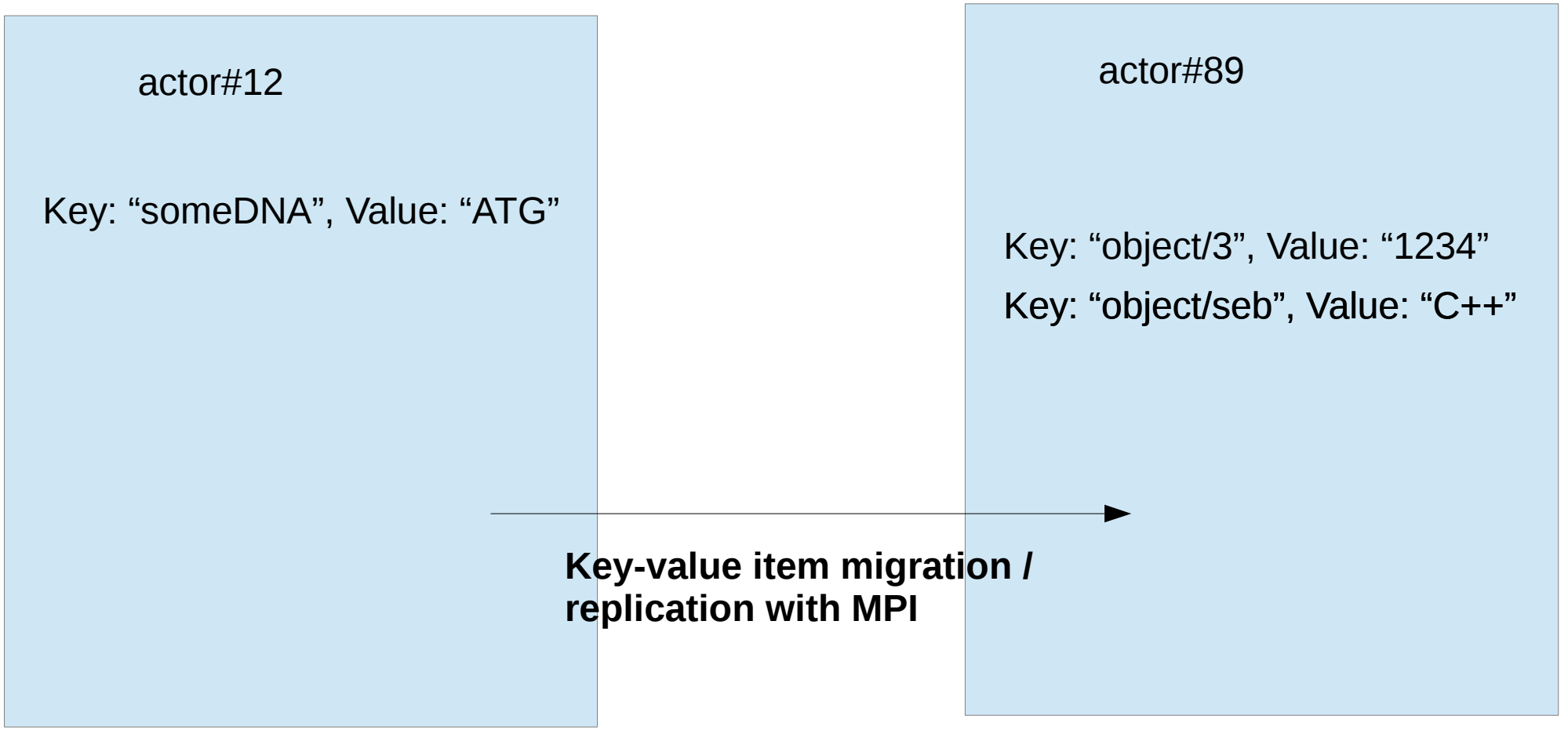

Query message
Actor#1
Actor#0
Response message
Actor#2
Actor#3


Programmatic access to data with API
Actor#0
Actor#1
Actor#2
Actor#3

Extra slides

Use case: synchronization between ranks in RayPlatform

- In progress: RayPlatform has a API to send and receive key-value objects between ranks
- Not global though
- Actors http://www.slideshare.net/drorbr/the-actor-model-towards-better-concurrency
- Actors can retrieve key-value items from remote actor if key is published
- Eases transfer of objects between actors


actor#12
Key: “someDNA”, Value: “ATG”
actor#89
Key: “object/3”, Value: “1234”
Key: “object/seb”, Value: “C++”
Key-value item migration / replication with MPI

Questions

- There is MPI on top of PMPI / PAMI / SPI, so is there something portable for key-value stores ?
- What's the typical latency for retrieving a key with PIMD (let's say key has 32 bytes and value has 512 bytes) ?
- How efficient is PIMD with small values (< 512 KiB) ?