



Building a Cloud-Native
High-Performance Computing
Platform for Biomedical Needs



LIN Yu-Ting, Senior Staff Engineer, 2022-03-15

yuting.lin@atgenomix.com

Agenda:



1. About Speaker / About Atgenomix
2. Challenges of Biomedical Industry
3. Workflow Description Languages
4. Akka Actor System
5. Apache Spark Framework
6. Conclusions



About Speaker:

- 2010 台灣大學電機系學士
- 2012 台灣大學電信所碩士
- 2017 巴黎高等電信學院博士

- Research Engineer at Orange Labs, France, 2013-2016
- Big Data Engineer at Société Générale, France, 2017-2020
- Senior Staff Engineer at Atgenomix, Taiwan, 2020-now

- Telecom System Dimensioning and Performance Optimization
- Big Data System Development and Operation



ATGENOMIX empowers its partners to achieve more with smart high-performance BioMed-IT technologies to deliver on the promise of precision medicine.



Microsoft
for Startups

2019



Gold
Microsoft
Partner



2020 Top Innovative Partner

2020

BIOMED ACCELERATOR
TAIPEI MEDICAL UNIVERSITY



比翼加速器
BE Accelerator

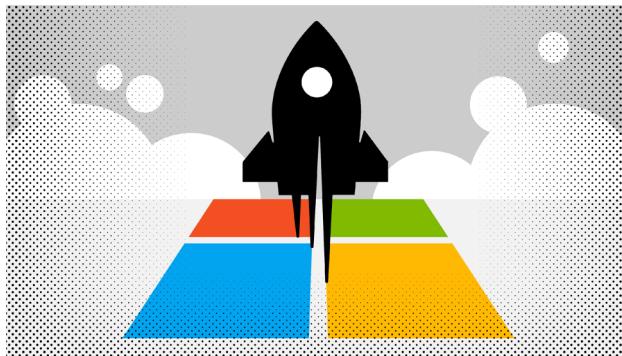
2021

Microsoft Partner Journey to Achieve More

2020 微軟年度最具創新合作夥伴



1. Microsoft for Startups



Microsoft Accelerator Taipei 1st batch 2019.

Microsoft Global Social Entrepreneurship Program.

2. Microsoft Co-sell Program

Gold
Microsoft Partner



IP (Intellectual Property) Co-sell Ready & CSP (Cloud Service Provider) Reseller Incentivized.
Gold competency in Data Platform.

3. Microsoft Commercial Marketplace



Microsoft Cloud for Healthcare

Driving Atgenomix SeqsLab usage/adoption on global Azure.
Engaging on solution partnership of Microsoft Cloud for Healthcare.

Strategic Partnership on NGS Sequencing Analysis



Whole Transcriptome

Whole Exome

Whole Genome

Gene Fusion

Immunome

Cell-free DNA

Targeted Amplicon

Solid Tumor

Liquid Biopsy

Professional Services

Data Governance

Trusted Cloud Security

High Performance Computing

Workflow Automation

Clinical Reporting

Medical Interpretation

Drug-target Database

Bioinformatics



Microsoft Cloud
for Healthcare



IEC 62304:2006+A1:2015
FDA/MDCG Cybersecurity Guidance



Next Generation Sequencing



Wet Lab

Dry Lab



Unified Clinical Workflow Solution From Samples to Actions



- Sample processing
- Library prep
- Custom focus panel

- NGS Sequencing
- Quality control

- Data processing
- Variant calling
- Differential expression

- Functional annotation
- Mutational profiling
- Joint genotyping

- Medical interpretation
- Classification
- Reporting

- Datasets aggregation
- Multi-omics analysis

WGS, WES, Targeted Panel, Ampli-seq, RNA-seq

Knowledge database, Prioritization

CAP-certified NGS Laboratory Services

Clinical Reporting Services

Build your own bioinformatics pipelines

Database, Machine learning, Cohort study

Workflow Design Workbench

Medical Data Repository

Task automation, Process orchestration, Systems integration, Collaboration

Cloud-native Biomedical Informatics Platform

實驗室開發檢測拍板，台灣生技產業接軌國際



LDTs 操作資格分 4 大類

項目	資格	臨床經驗或訓練證明
專任品質主管	醫事檢驗師或專科醫師	且有臨床檢驗品質管理及相關實驗室開發檢測經驗二年以上。
專任技術人員	醫事檢驗師	完成中央主管機關公告之訓練課程及時數，取得訓練單位發給之證明。
專任檢測開發、分析、校正、生物資訊處理及其他相關人員	不限	完成中央主管機關公告之訓練課程及時數，取得訓練單位發給之證明。
核發檢測報告人員	醫事檢驗師或專科醫師	經相關訓練之醫事檢驗師或專科醫師，並得由專任品質主管或專任技術人員擔任

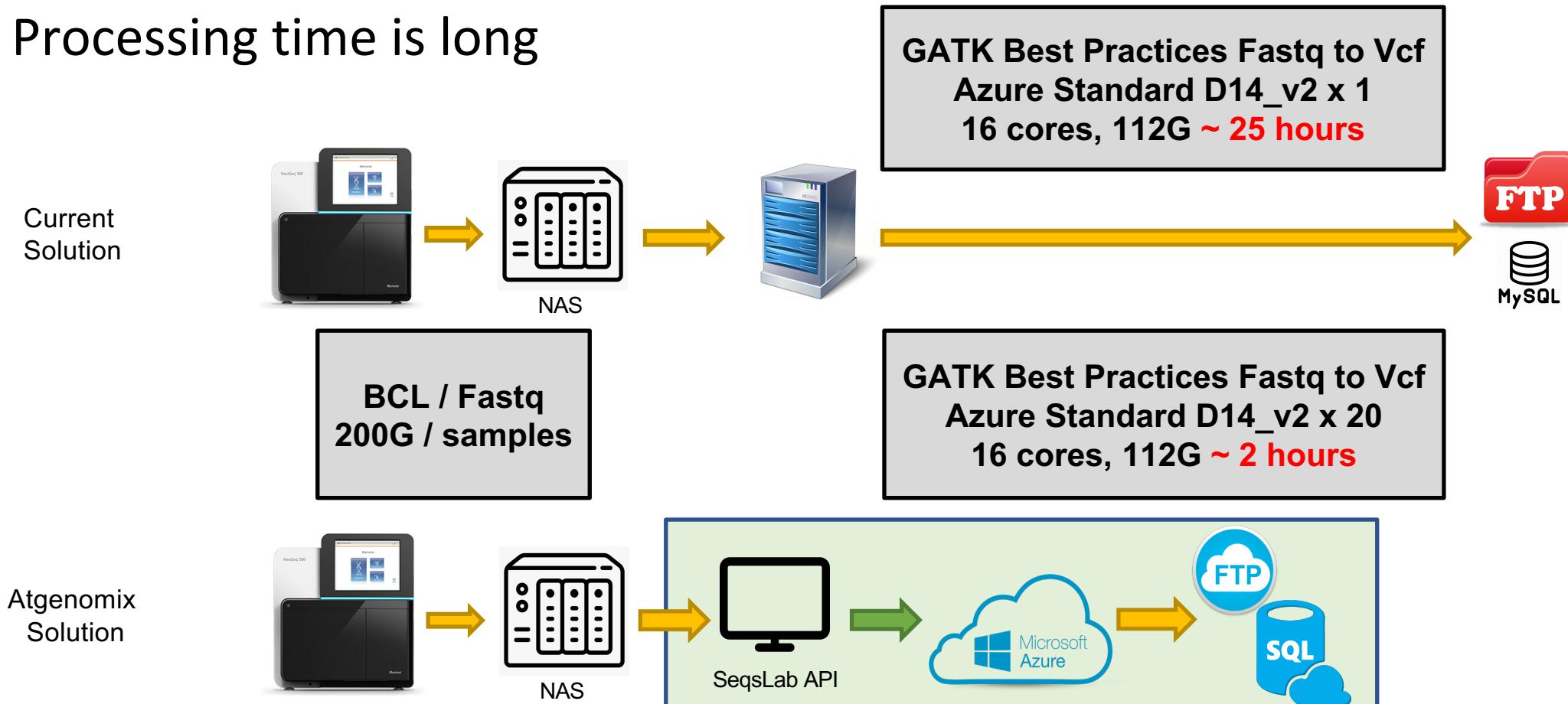
LDTs 7 大項目

- 一、抗癌瘤藥物之伴隨檢測
- 二、癌症篩檢、診斷、治療及預後之基因檢測
- 三、產前及新生兒染色體與基因變異檢測
- 四、藥物不良反應或藥物代謝之基因檢測
- 五、遺傳代謝與罕見疾病之基因檢測
- 六、病原體鑑定、毒力及抗藥性基因檢測
- 七、其他藥物伴隨基因檢測（於藥物仿單中，明載於用藥前應執行檢測）



Challenges of Next Generation Sequencing

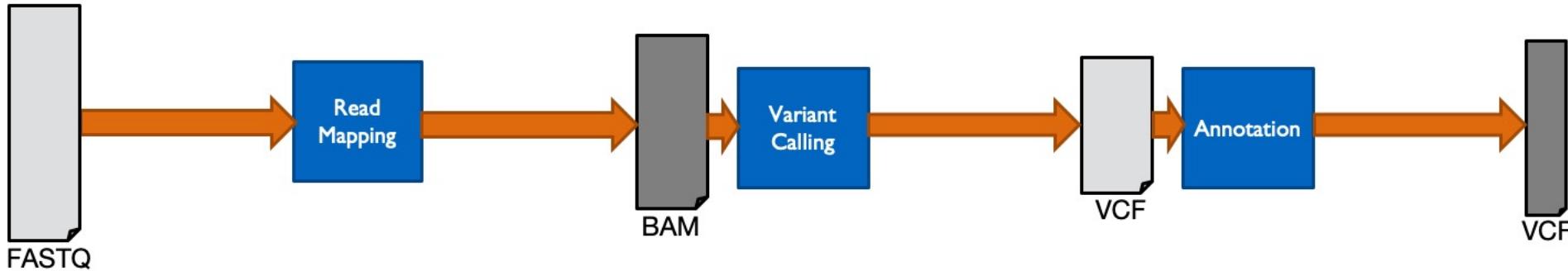
- Data is huge
- Processing time is long



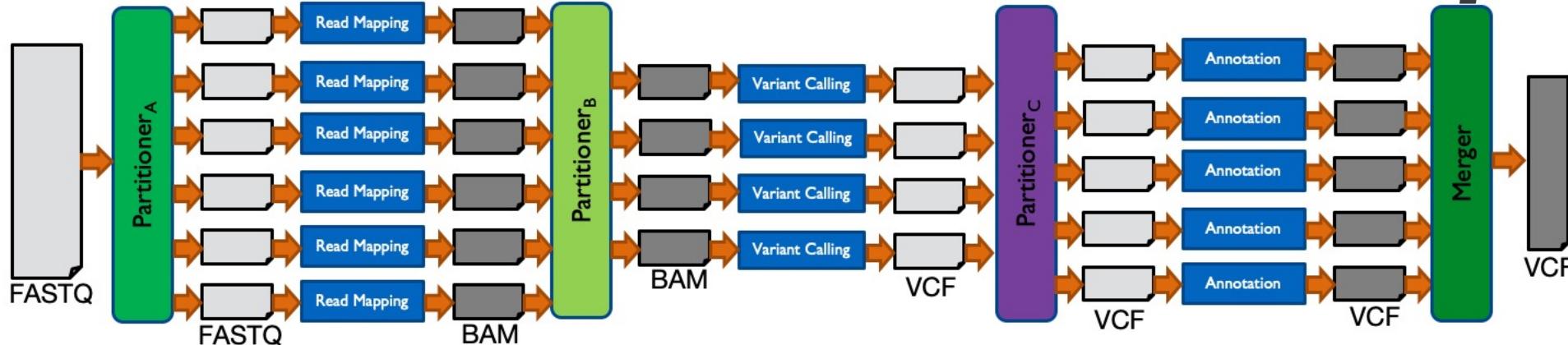
Next-Gen Computational Framework for Bioinformatics



Non-Data Parallelization Mode



Data Parallelization Mode



PROVISIONAL PATENT
APPLICATION



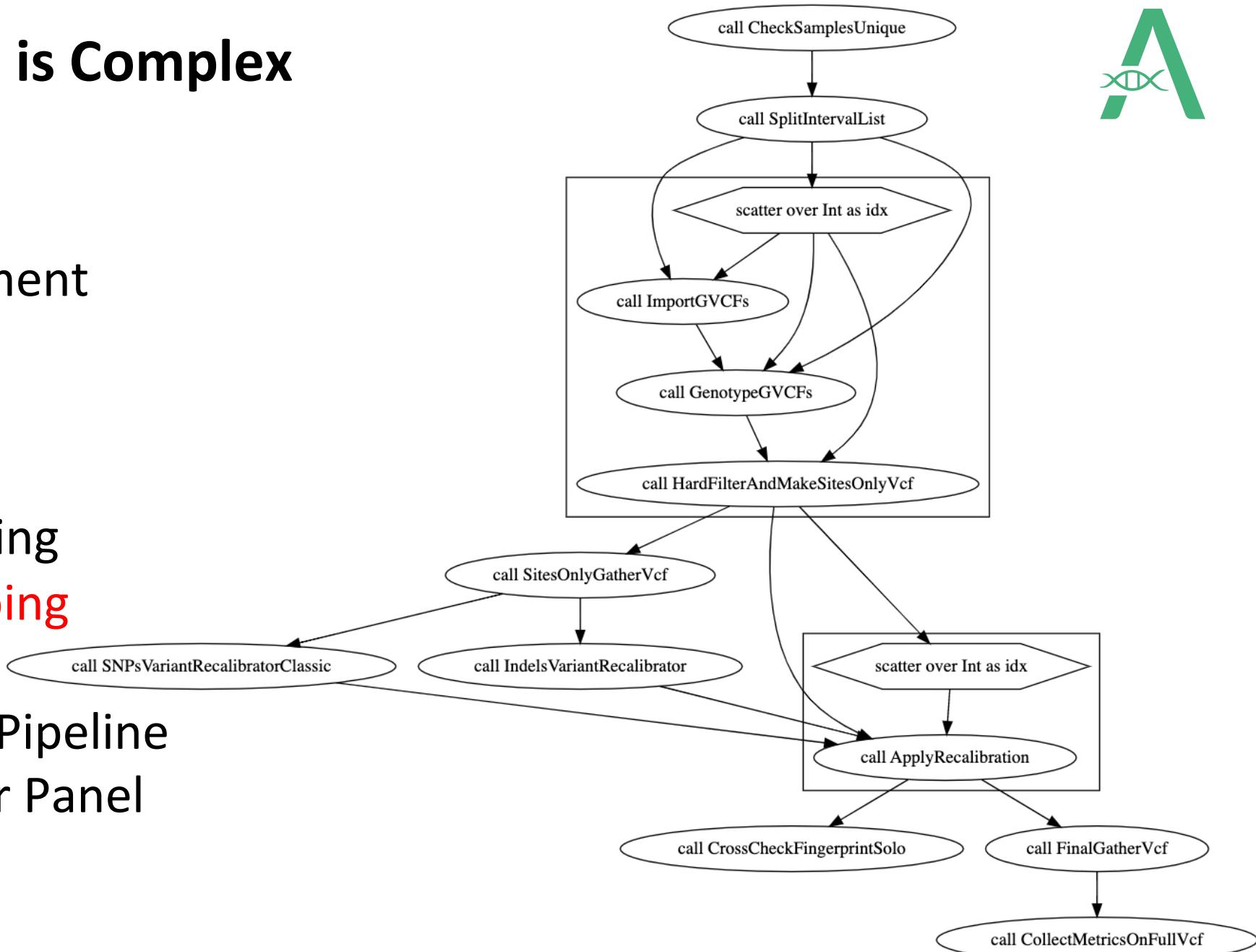
Biomedical Pipeline is Complex



- Multiple-Threading
- Workflow Management

Example:

- GATK Snp Indel Calling
- **GATK Joint Genotyping**
- NCKU Nipt Pipeline
- Yourgene Covid-19 Pipeline
- Actgenomics Cancer Panel





Workflow Description Language

```
1  version 1.0
2
3  import "../../gatk4-data-processing/processing-for-variant-discovery-gatk4.wdl" as mapper
4  import "../../gatk4-germline-snps-indels/haplotypecaller-gvcf-gatk4.wdl" as caller
5
6  # WORKFLOW DEFINITION
7  workflow e2e_HaplotypeCallerGvcf_GATK4 {
8      input {
9          String sample_name
10         String ref_name
11         Array[File] fastq_files
12         File ref_fasta
13         ...
14     }
15     call mapper.PreProcessingForVariantDiscovery_GATK4 {
16         input:
17             sample_name = sample_name,
18             ref_name   = ref_name,
19             fastq_files = fastq_files,
20             ...
21     }
22     call caller.HaplotypeCallerGvcf_GATK4 {
23         input:
24             input_bam      = PreProcessingForVariantDiscovery_GATK4.analysis_ready_bam,
25             ref_dict      = ref_dict,
26             ref_fasta    = ref_fasta,
27             ...
28     }
29     output {
30         File e2e_analysis_ready_bam = PreProcessingForVariantDiscovery_GATK4.analysis_ready_bam
31         File e2e_output_vcf = HaplotypeCallerGvcf_GATK4.output_vcf
32         ...
33     }
34 }
```

Open-Sourced by Broad Institute

- Human Readable
- Standardized Pipeline

Executed by Engines:

- Cromwell
- MiniWDL
- dxWDL
- **Seqslab**

Workflow-Integrated Infrastructure as Code



```
1 # Mark duplicate reads to avoid counting non-independent observations
2 task MarkDuplicates {
3     input {
4         File input_bam
5         String output_bam_basename
6         String metrics_filename
7
8         Int compression_level
9         Int preemptible_tries
10        Int disk_size
11        Float mem_size_gb = 7.5
12
13        String docker_image
14        String gatk_path
15    }
16    Int command_mem_gb = ceil(mem_size_gb) - 2
17    # Task is assuming query-sorted input so that the Secondary and Supplementary reads get marked correctly.
18    # This works because the output of BWA is query-grouped and therefore, so is the output of MergeBamAlignment.
19    # While query-grouped isn't actually query-sorted, it's good enough for MarkDuplicates with ASSUME_SORT_ORDER="queryname"
20    command {
21        ~{gatk_path} --java-options "-Dsamjdk.compression_level=~{compression_level} -Xms~{command_mem_gb}G" \
22            MarkDuplicates \
23            --INPUT ~{input_bam} \
24            --OUTPUT ~{output_bam_basename}.bam \
25            --METRICS_FILE ~{metrics_filename} \
26            --VALIDATION_STRINGENCY SILENT \
27            --OPTICAL_DUPLICATE_PIXEL_DISTANCE 2500 \
28            --ASSUME_SORT_ORDER "queryname" \
29            --CREATE_MDS_FILE true \
30            --READ_NAME_REGEX null
31    }
32    runtime {
33        preemptible: preemptible_tries
34        docker: docker_image
35        memory: "~{mem_size_gb} GiB"
36        disks: "local-disk " + disk_size + " HDD"
37    }
38    output {
39        File output_bam = "~{output_bam_basename}.bam"
40        File duplicate_metrics = "~{metrics_filename}"
41    }
42 }
```



Data Repository Integration

Orchestrate, govern, and streamline workload integration of diverse input and output across heterogeneous data repositories in a standard way.

Tooling Docker Containers

Build, secure, and manage trusted pipeline command tools to assure quality, consistency, and compliance across environments.

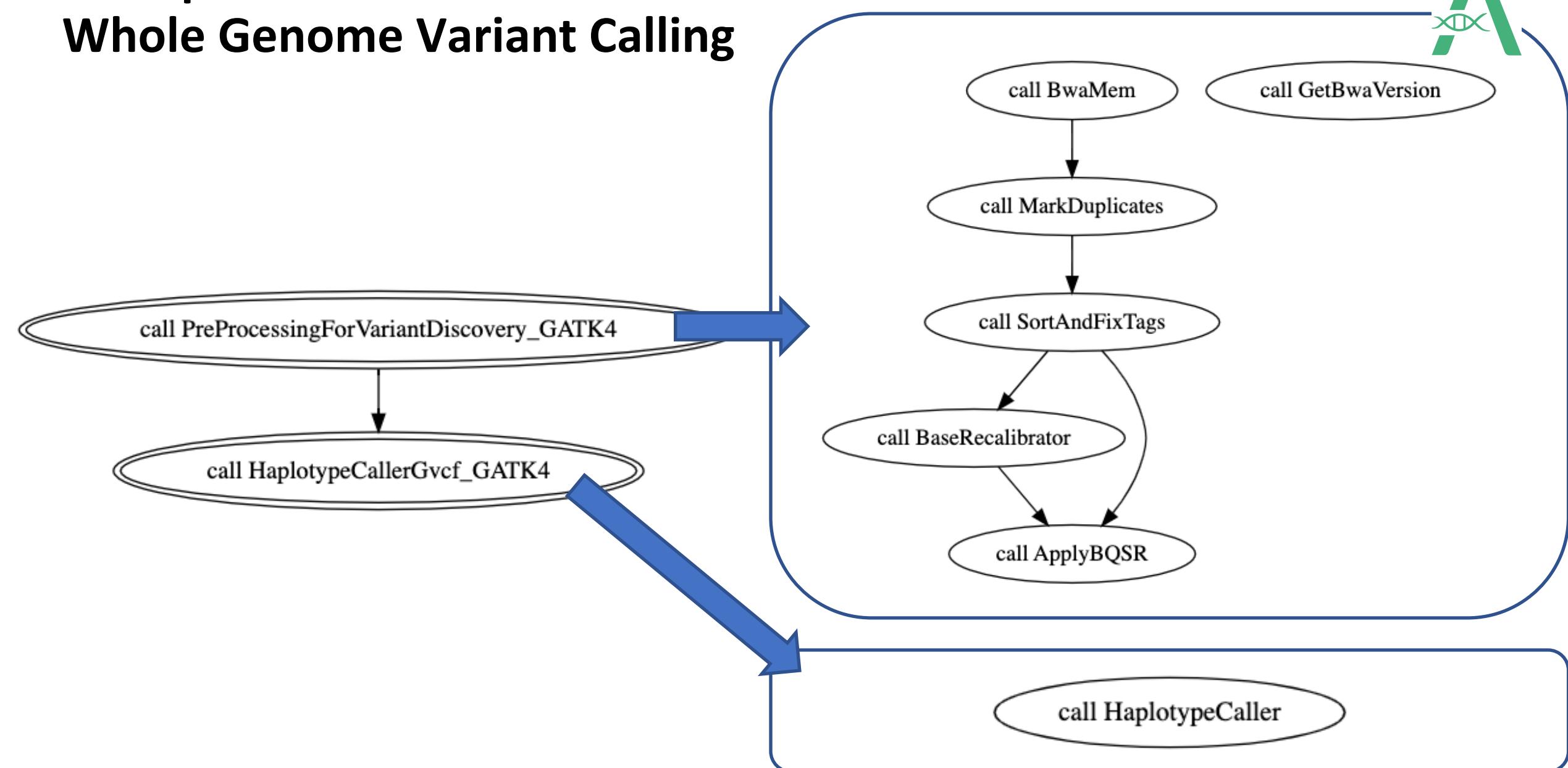
On-demand Immutable Infrastructure

Automate and optimize workflow execution with elastic and fully-managed high-performance computing in parallel.

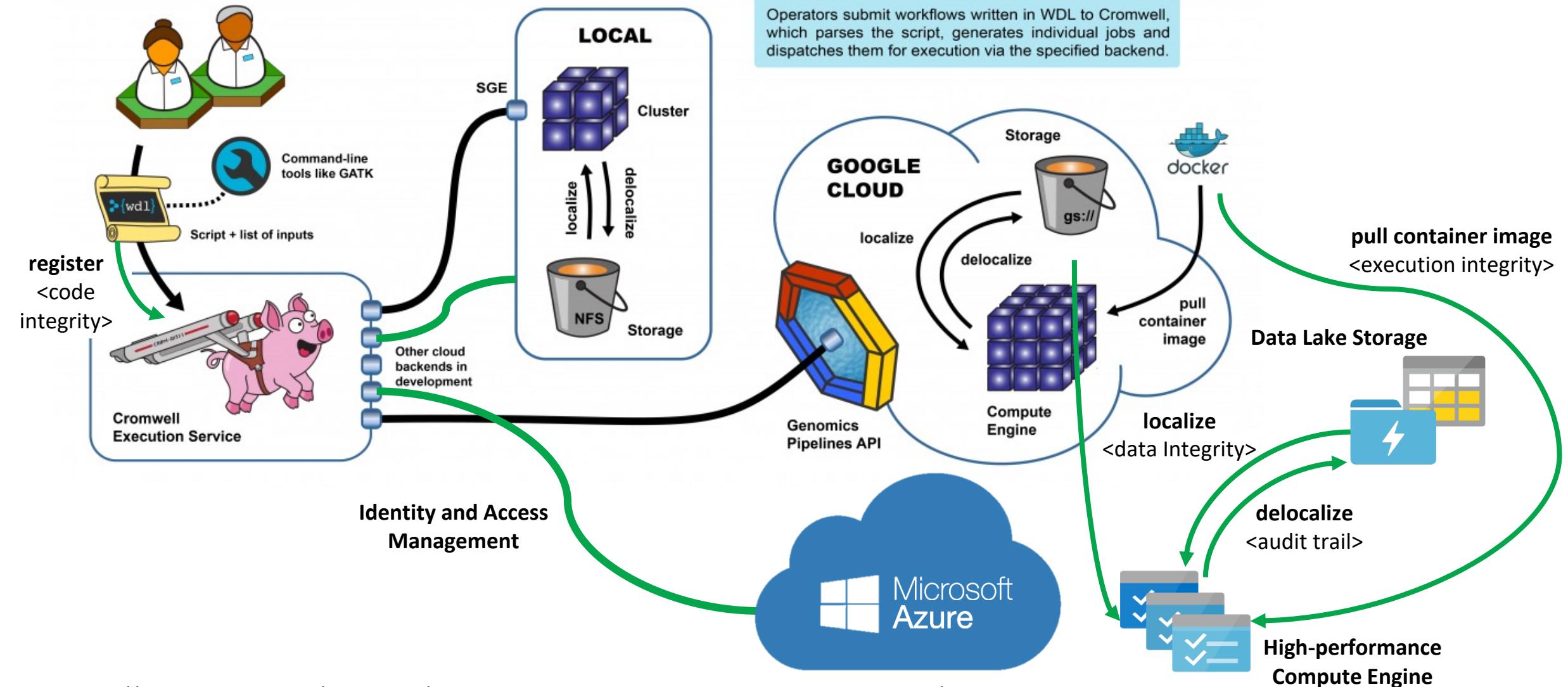
Global Network, Regional Provisioning

Control owned infrastructure and scale to service needs compliant with regional data governance requirements.

Example - GATK Whole Genome Variant Calling



Workflow Management Optimized for Cybersecurity, Usability, and Scalability





On-Demand Resources Provisioning

Batch Explorer Pools > e2e-haplotypecallergvcf-gatk4-1f52b-run-g5ulah1bl8dpcqu

No favorite items pinned 3

POOLS

Search (Startwith)

e2e-haplotypecallergvcf-gatk4-0d62a-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-1f52b-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-1f925-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-2f133-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-32f0a-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-36aab-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-3b8d1-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-3dc84-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-42f20-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-4943a-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-53192-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-56190-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-571d0-run-g5ulah1bl8dpcqu active

e2e-haplotypecallergvcf-gatk4-5c4ec-run-g5ulah1bl8dpcqu active

Dedicated cores quota 2947/3794 (77%)

e2e-haplotypecallergvcf-gatk4-1f52b-run-g5ulah1bl8dpcqu
standard_d14_v2, steady
Last resized 53 minutes ago
UbuntuServer 18.04-LTS

4 USD 4.78/h

Bar chart showing resource usage over time (This month):

Time Period	Value
0 - 1 hour	~3.0
1 - 2 hours	~1.0
2 - 3 hours	~0.0

Graphs Configuration Nodes

Heatmap Available nodes Running tasks CPU Memory Disk IO Disk usage Network

No tags.

Legend:

- idle (3)
- running (1)
- waitingforstarttask (0)
- offline (0)
- preempted (0)
- Transition states
 - creating (0)
 - starting (0)
 - rebooting (0)
 - reimaging (0)
 - leavingpool (0)
- Error states
 - starttaskfailed (0)
 - unusable (0)
 - unknown (0)

seqlabwu2c11c8batch Admin/Developer workspace GMT+8 Python backend disconnected No current background tasks

Brain Storming – Design a Software System ?



scatter over Array[File] as fastq_pair

call e2e_Haplotype

call BwaMem

call GetBwaVersion

call MarkDuplicates

Multi-Threading Program Design ?

Event-Driven Architecture ?

call PreProcessingFor

call HaplotypeCallerGvcf_GATK4

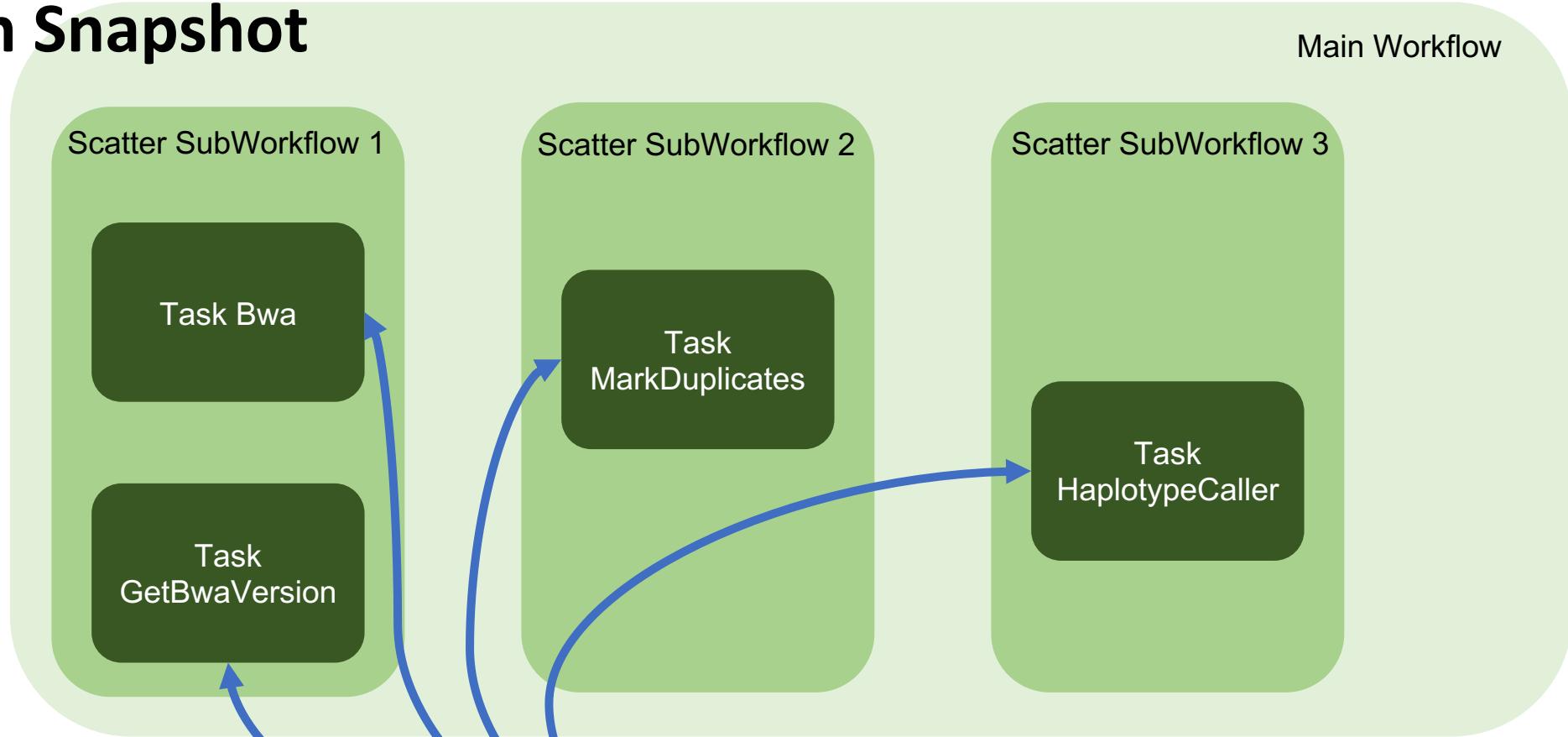
call ApplyBQSR

call HaplotypeCaller

System Snapshot



Main Workflow



Akka Actor System:



“The actor model in computer science is a mathematical model of concurrent computation that treats actor as the universal primitive of concurrent computation.”

Wikipedia

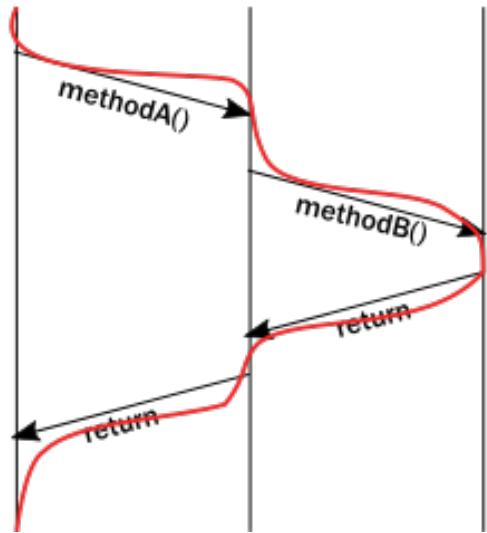
When a message it receives, an actor can:

1. make local decisions
2. create more actors
3. send more messages
4. determine how to respond to the next message received.

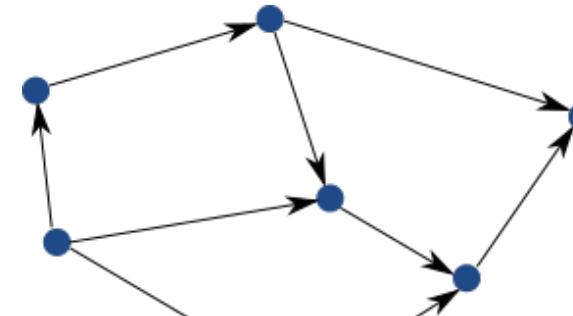
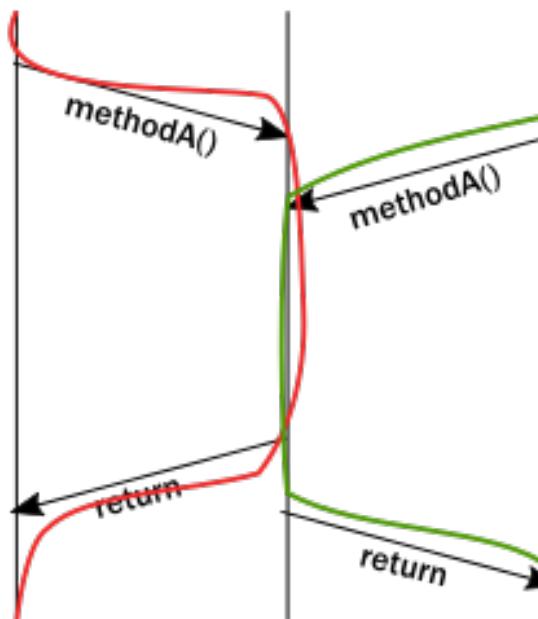


Multi-Threading Challenge

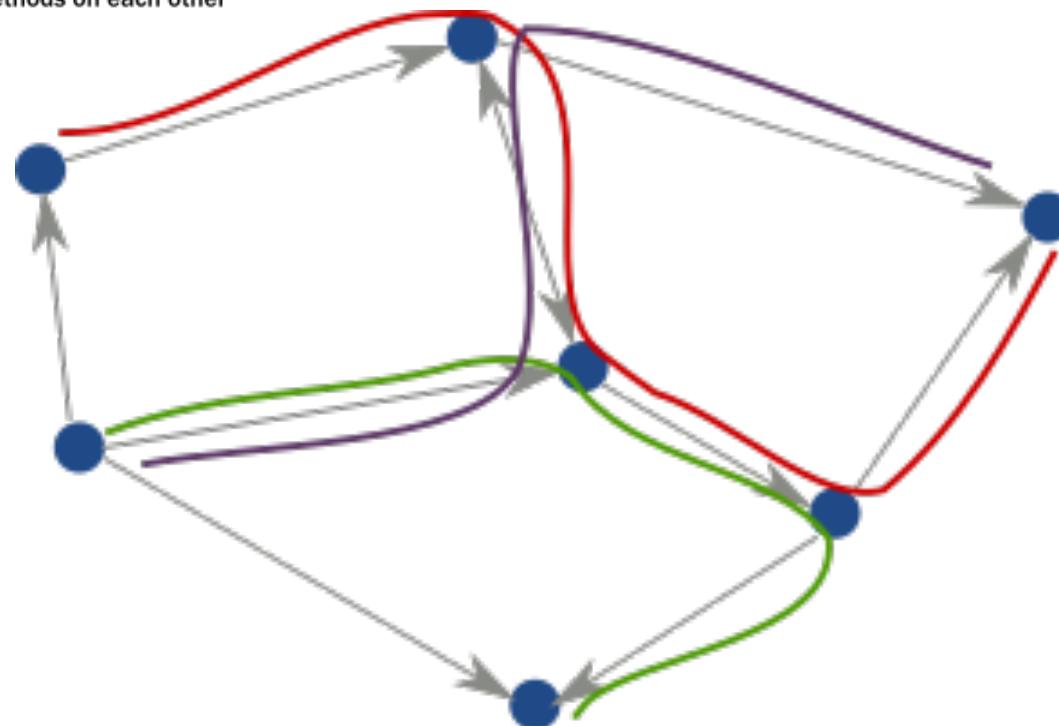
Object1 Object2 Object3



Object1 Object2 Object3

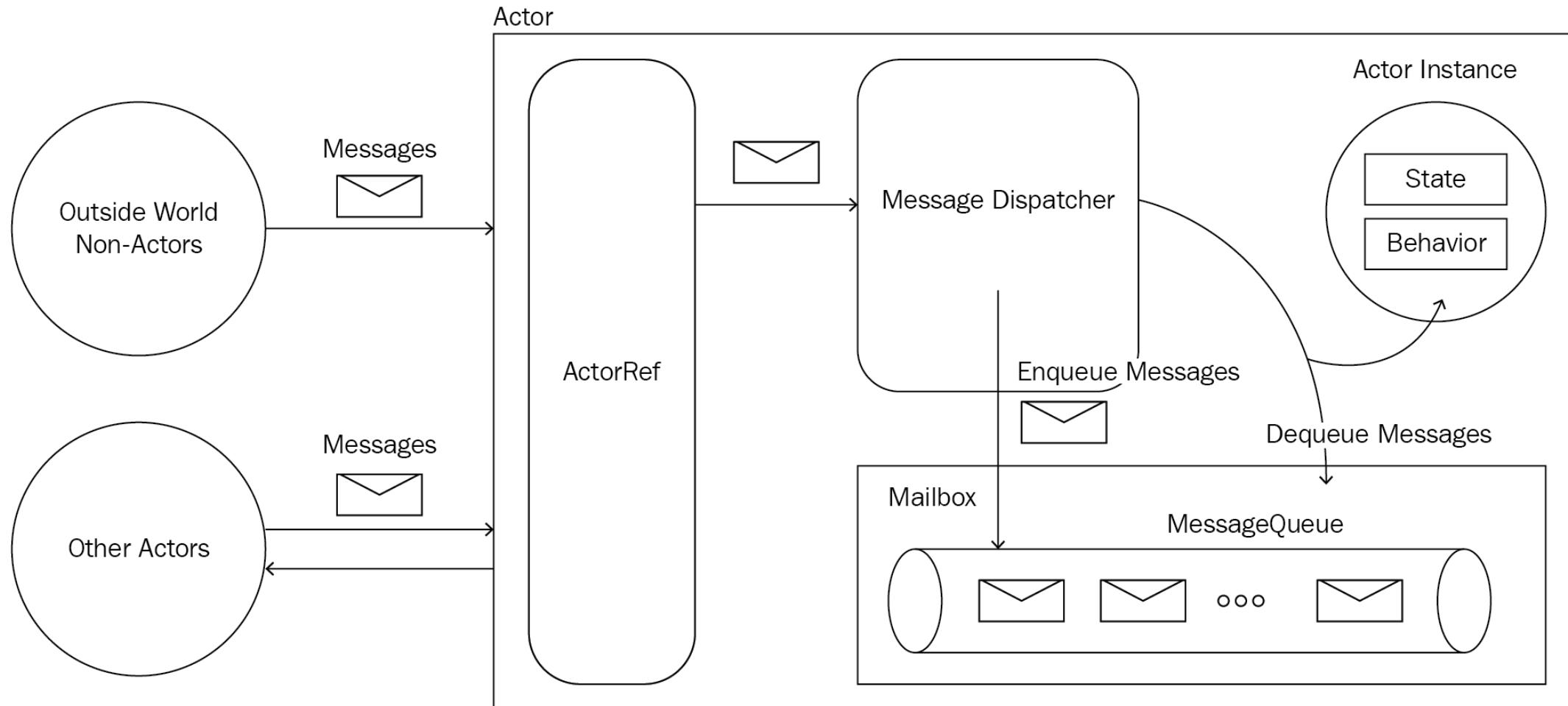


Objects interacting with each other
by calling methods on each other



Objects interacting with each other
Threads A, B, C, interacting with each other,
traversing method calls on objects

Akka Actor System



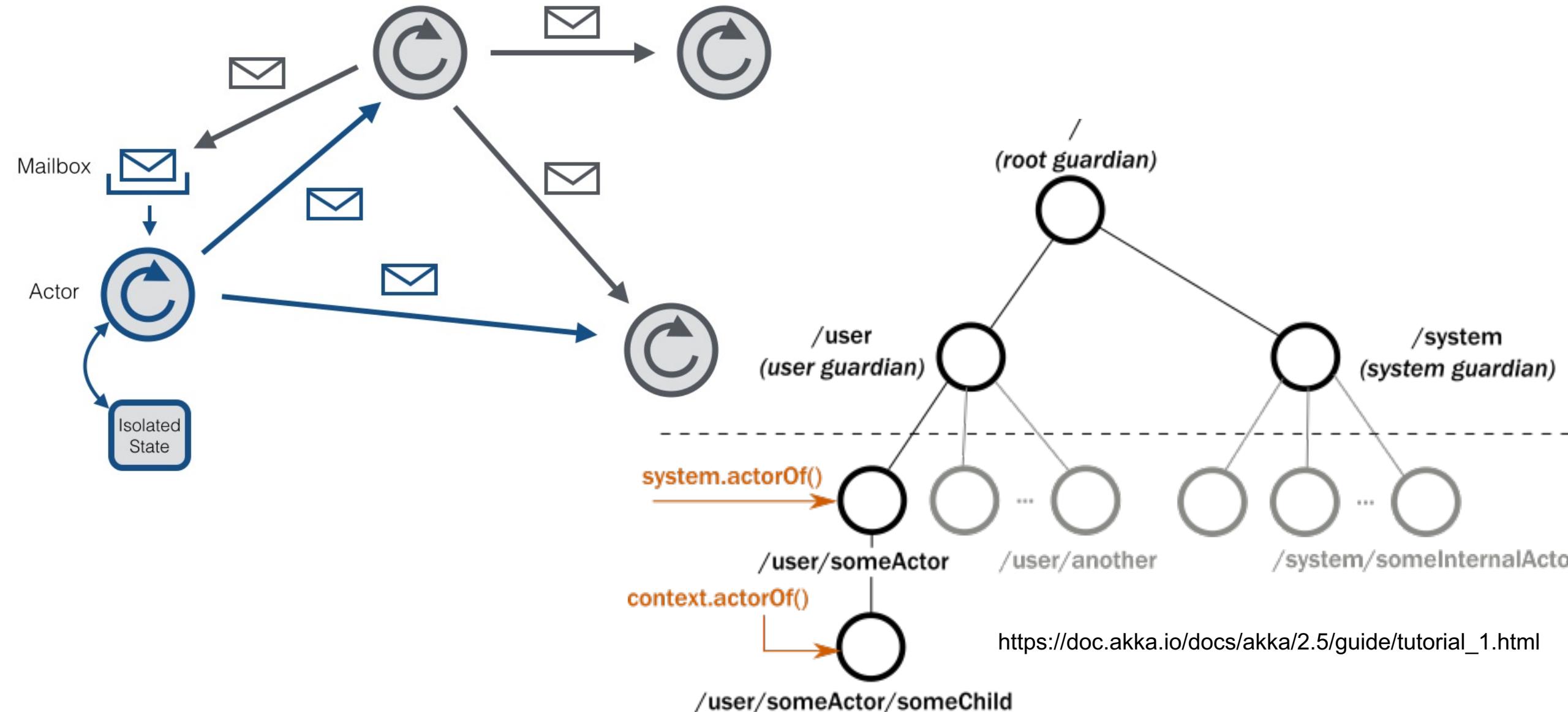
Internal Components of Akka Actor

https://tech401.com/2018/05/31/2018/2018-05-31-introduction_to_akka_model/

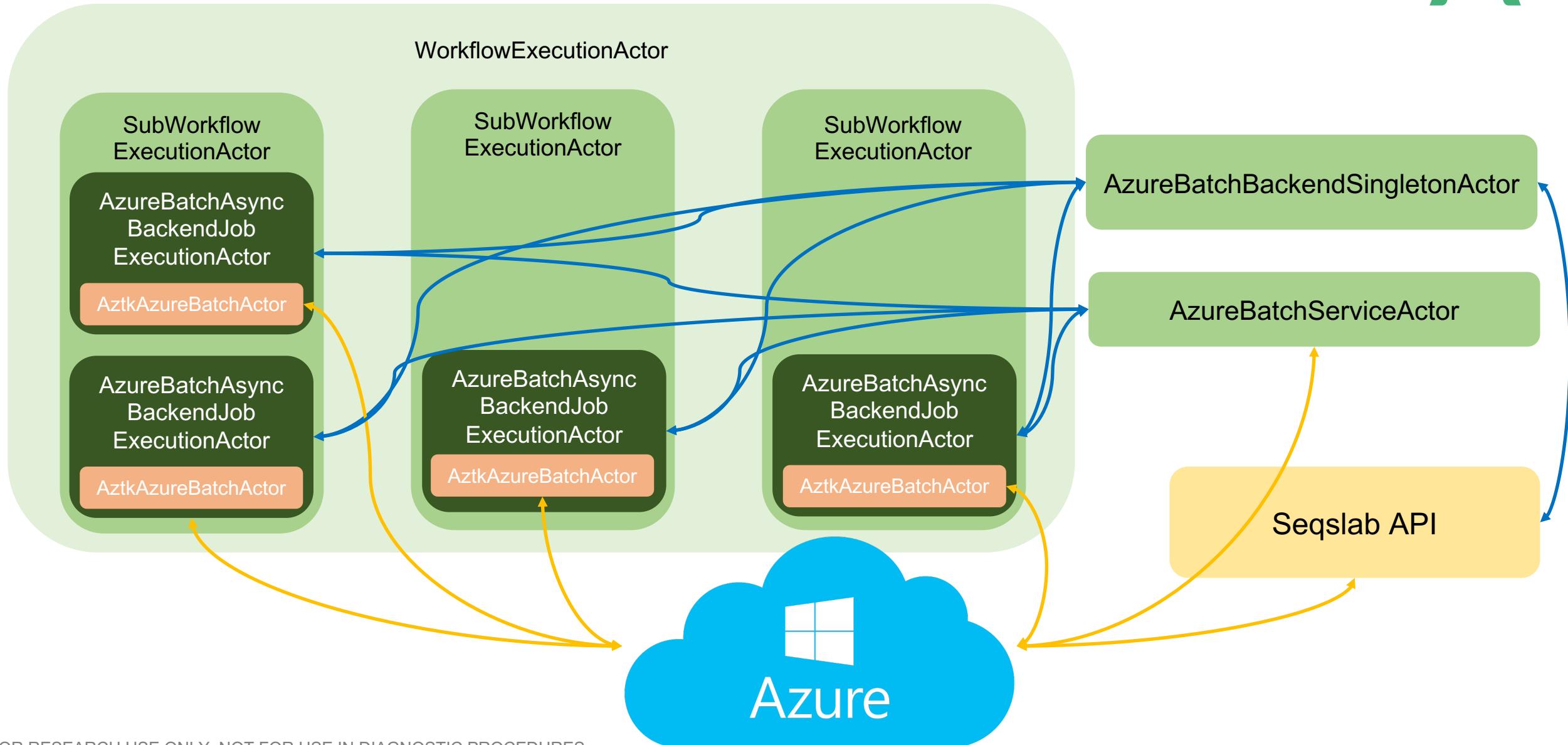
Akka Actor System



<https://cwiki.apache.org/confluence/display/FLINK/Akka+and+Actors>



Akka Actor System in Seqslab WE2 Engine



Create Child Actor



```
class AzureBatchAsyncBackendJobExecutionActor(override val standardParams: StandardAsyncExecutionActorParams)
  extends SeqslabAsyncBackendJobExecutionActor(standardParams) {

  private implicit val timeout: Timeout = Timeout(900.seconds)
  private val initData = backendInitializationDataOption.get.asInstanceOf[SeqslabBackendInitializationData]
  private val pollStatusRetryMax: Int = 5
  var pollStatusRetryCounter: Int = 0
  private val aztkAzureBatchActor = context.actorOf(AztkAzureBatchActor.props(initData, asyncIo))

  override type StandardAsyncRunInfo = SeqslabJob
  override type StandardAsyncRunState = SeqslabRunStatus

  def getNodeState(nodeId: String, seqslabJob: SeqslabJob): Future[String] = {...}
  def getMasterNodeState(seqslabJob: SeqslabJob): Future[String] = {...}
  def allNodeFailed(seqslabJob: SeqslabJob): Future[Boolean] = {...}

  override def executeAsync(): Future[ExecutionHandle] = {

    def deleteAllUpperClusters(pass: Boolean, job: SeqslabJob): Future[Unit] = {...}

    @tailrec
    def waitAztkClusterStart(pass: Boolean, msg: String, seqslabJob: SeqslabJob, wait: Long = 90000L): Future[(Boolean, String)] = {...}

    @tailrec
    def createCluster(create: Boolean, seqslabJob: SeqslabJob, sparkConf: File, retry: Int=3): Future[(Boolean, String)] = {...}

    for {
      job <- (backendSingletonActor ? TaskPreparation(
        jobDescriptor,
        jobPaths,
        workflowPaths.asInstanceOf[SeqslabWorkflowPaths],
        sparkConf,
        retry,
        pass
      )).mapTo[TaskPreparationResponse]
    } yield {
      val taskFuture = createTask(job, seqslabJob, jobDescriptor, jobPaths, workflowPaths, sparkConf, retry, pass)
      val resultFuture = taskFuture.map { task =>
        if (task.isCompleted) {
          if (task.isFailure) {
            log.error(s"Task $task failed with error ${task.failure}")
            (false, s"Task $task failed with error ${task.failure}")
          } else {
            log.info(s"Task $task completed successfully")
            (true, s"Task $task completed successfully")
          }
        } else {
          log.info(s"Task $task is still in progress")
          (false, s"Task $task is still in progress")
        }
      }
      resultFuture.onComplete { result =>
        if (result.isFailure) {
          log.error(s"Error occurred while waiting for task $task: ${result.failed}")
        } else {
          log.info(s"Task $task completed with result ${result.value}")
        }
      }
    }
  }
}
```

Messages Handling



```
final class AztkAzureBatchActor(initializationData: SeqslabBackendInitializationData,  
                                azureBatchClient: AzureBatchServiceClient,  
                                azureStorageClient: AzureStorageClient,  
                                asyncIo: AsyncIo) extends Actor with ActorLogging {  
  
    val aztkPath: String = initializationData.seqslabConfiguration.getAztkPath  
    val mountPath: String = initializationData.seqslabConfiguration.getMountPath  
    val coreSite: File = initializationData.coreSite.get  
    val builders: PathFactory.PathBuilders = initializationData.pathBuilders  
  
    override def receive: Receive = {  
        // Aztk Related Actions  
        case CreateSparkConf(job) => sender ! createSparkConf(job, asyncIo, builders)  
        case CreateAztkCluster(mainWorkflowPath, job, secret, sparkConf, vSubnet) =>  
            ...  
        case SubmitJob2AztkCluster(initData, secret, sparkConf, job, taskFQN) =>  
            ...  
  
        // Azure Batch API Related Actions  
        case GetAztkPool(poolId) => sender ! azureBatchClient.getAztkPool(poolId)  
        case GetAztkTask(jobId, taskId) => sender ! azureBatchClient.getAztkTask(jobId, taskId)  
        case GetAztkMasterNodeId(poolId) => sender ! azureBatchClient.getAztkMasterNodeId(poolId)  
        case GetAztkComputeNode(poolId, nodeId) => sender ! azureBatchClient.getAztkComputeNode(poolId, nodeId)  
        case GetAztkTaskOutputLog(jobId, taskId) => sender ! azureBatchClient.getAztkTaskOutputLog(jobId, taskId)  
        case CopyBlob(src, dst) => azureStorageClient.copyBlob(src, dst)  
    }  
}
```



```
package cromwell.engine.workflow.lifecycle.execution

import ...

case class WorkflowExecutionActor(params: WorkflowExecutionActorParams)
  extends LoggingFSM[WorkflowExecutionActorState, WorkflowExecutionActorData]
    with WorkflowLogging
    with CallMetadataHelper
    with StopAndLogSupervisor
    with Timers
    with CromwellInstrumentation {

  when(WorkflowExecutionPendingState) (...)

  /* **** */
  /* ***** Running ***** */
  /* **** */

  when(WorkflowExecutionInProgressState) {
    // If we're done, the workflow is successful
    case Event(ExecutionHeartBeat, data) if data.done => handleWorkflowSuccessful(data)
  }

  /* **** */
  /* ***** Failing ***** */
  /* **** */

  when(WorkflowExecutionFailingState) (...)

  /* **** */
  /* ***** Aborting ***** */
  /* **** */

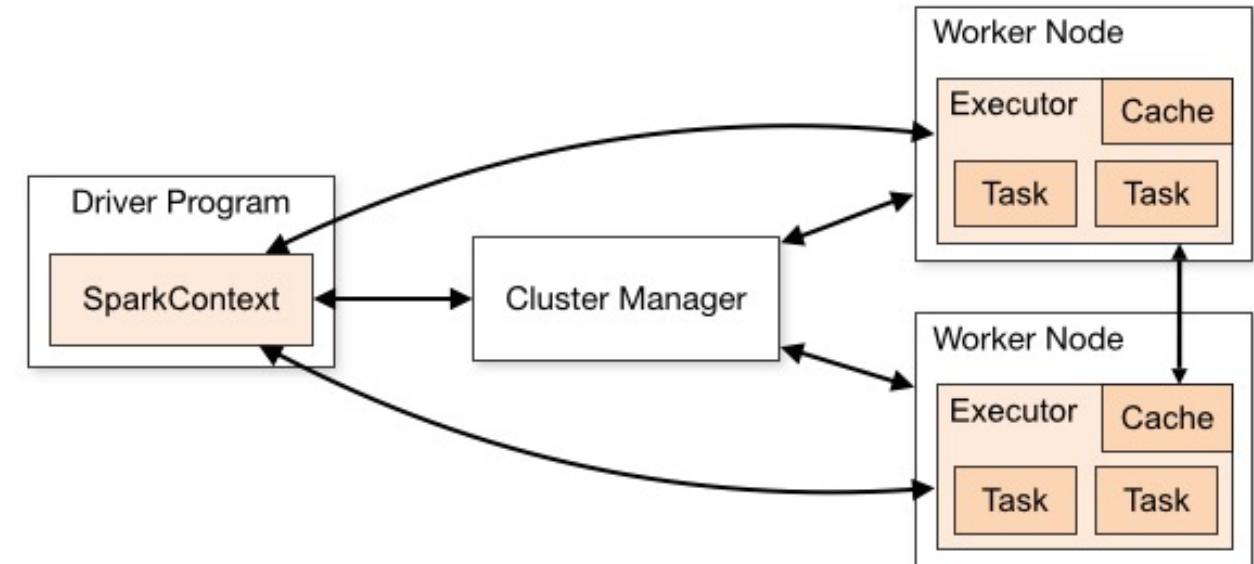
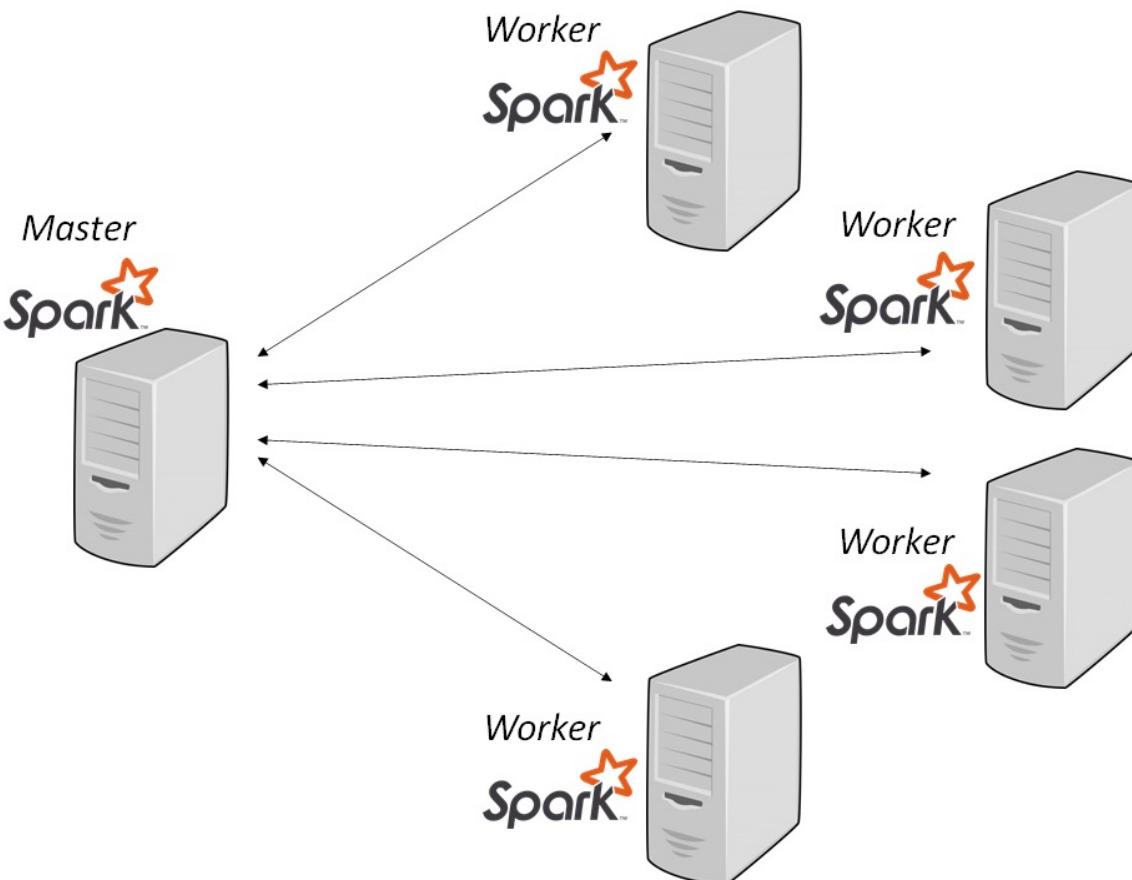
  when(WorkflowExecutionAbortingState) {
    // If we're done, the workflow is aborted
    case Event(ExecutionHeartBeat, data) if data.done =>
      context.parent ! WorkflowExecutionAbortedResponse(data.jobExecutionMap)
      goto(WorkflowExecutionAbortedState)
  }
}
```

Finite State Machine

Apache Spark

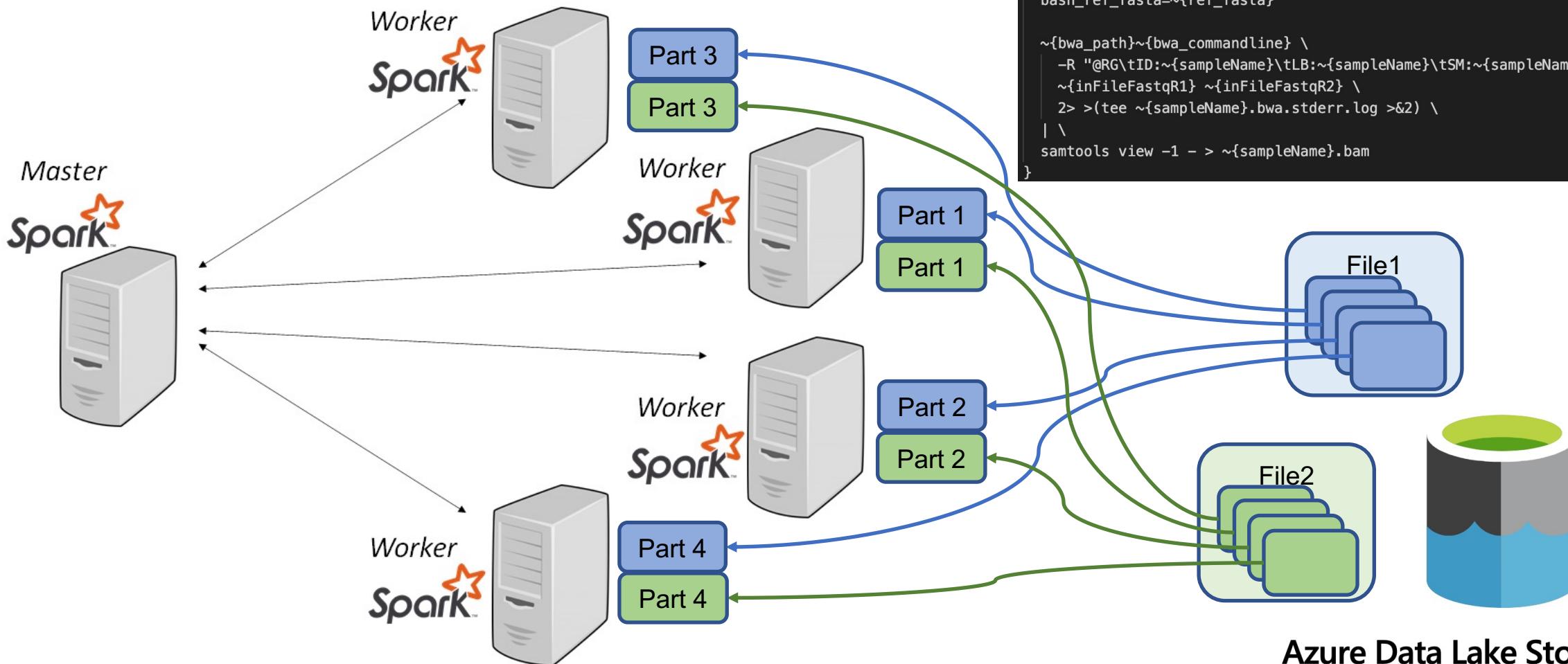


- Distributed Clusters Management for Parallel Computing



<https://spark.apache.org/docs/latest/cluster-overview.html>

RDD + repartitionAndSortWithinPartitions + mapPartitions





The Future of Healthcare Is in the Cloud

Secure and reliable virtual access to healthcare professionals and data has become table stakes for us to meet our 21st century challenges and goals.

[Add to Queue](#)**Morris Panner**

ENTREPRENEUR LEADERSHIP NETWORK CONTRIBUTOR

CEO of Ambra Health



February 6, 2021

5 min read

We will look back on 2020 as a pivotal moment for the use of [cloud computing](#) in [healthcare](#). As the pandemic swept away old constraints, digital [health](#) innovators rushed in. In the face of a major crisis, providers and technologists worked tirelessly to make healthcare better, pushing change to save lives. Innovation and [entrepreneurship](#) don't come without risk, but they also can provide enormous benefits. Collecting and sharing data via the cloud will enable a healthcare system fit for the 21st century.
<https://www.entrepreneur.com/article/363124>

The case for cloud

[NEXT ARTICLE](#)

Easy access to and sharing of data is an essential foundation for building a healthcare system that works for today's on-demand needs. Hybrid cloud deployment among healthcare providers is [expected](#) to reach just 37% this year, up from 19% in 2019. Most hospitals still rely on outdated software systems that have been repeatedly patched. Building atop shaky foundations like this leads to major inefficiency and frequent errors. Healthcare professionals lose a great deal of time that could be better spent on patient care to these inflexible and unreliable systems.

The cloud provides unprecedented scaling, data integration, and access advantages. Doctors with access to complete information on a patient's electronic health record (EHR), prescriptions, test results, and imaging, are better equipped to find the right diagnosis and identify the best course of treatment. Data-driven decisions, based on huge information sets, can help healthcare professionals and researchers to spot patterns, uncover insights, and deliver a higher standard of care.

Hidden Technical Debt in Machine Learning Systems

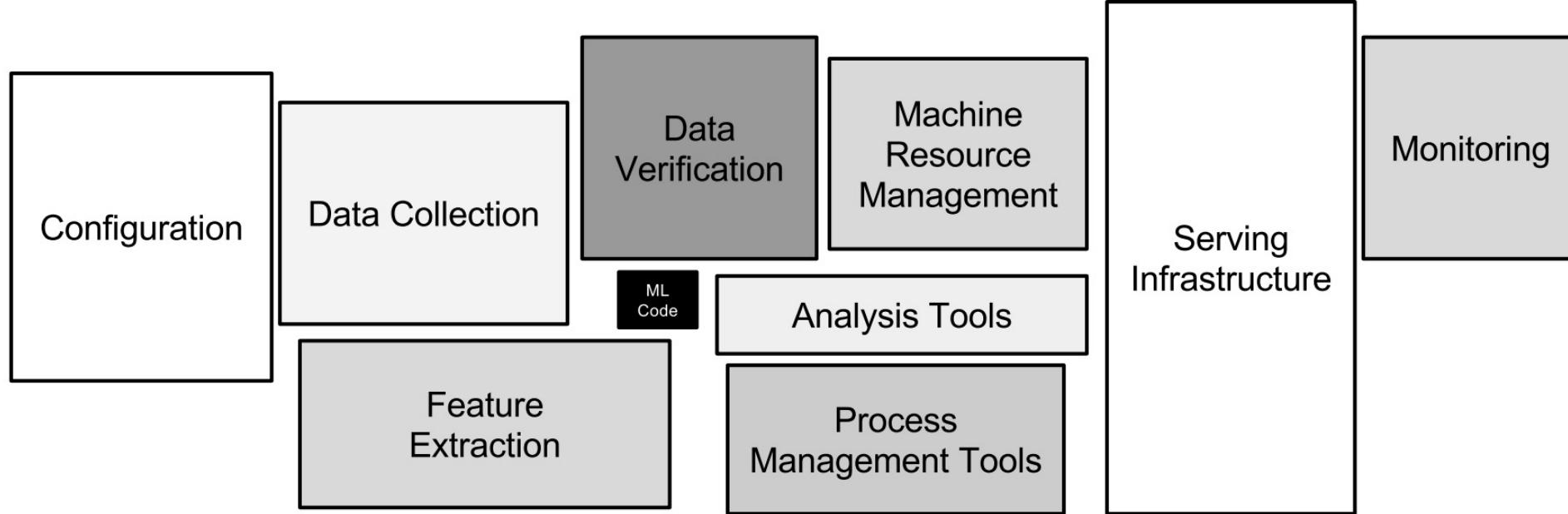
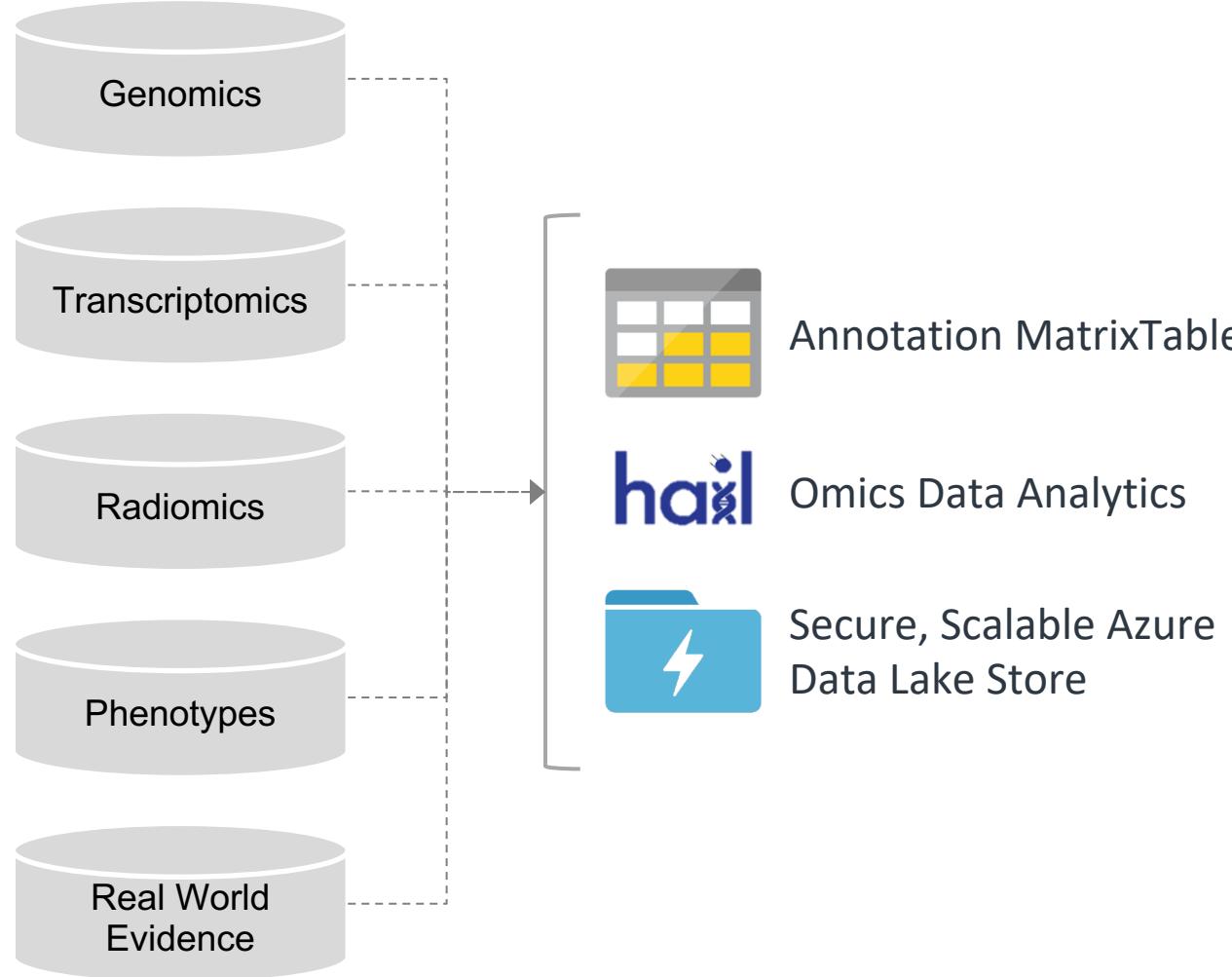
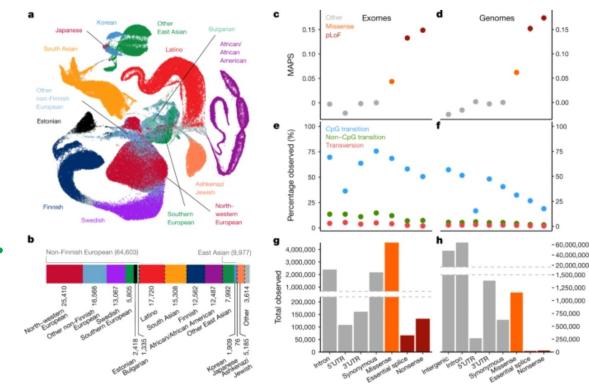


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Integrative Medical Data Repository Enabling Multi-Omics



- Genotype-phenotype / drug-response association study
- Interpretation-focused machine learning
- Variant filtering and annotation
- Plotting and visualization



<https://www.nature.com/articles/s41586-020-2308-7>



ABOUT HOW WE WORK TOOLKIT NEWS & EVENTS COMMUNITY IMPLEMENTATION CONTACT



Join Us



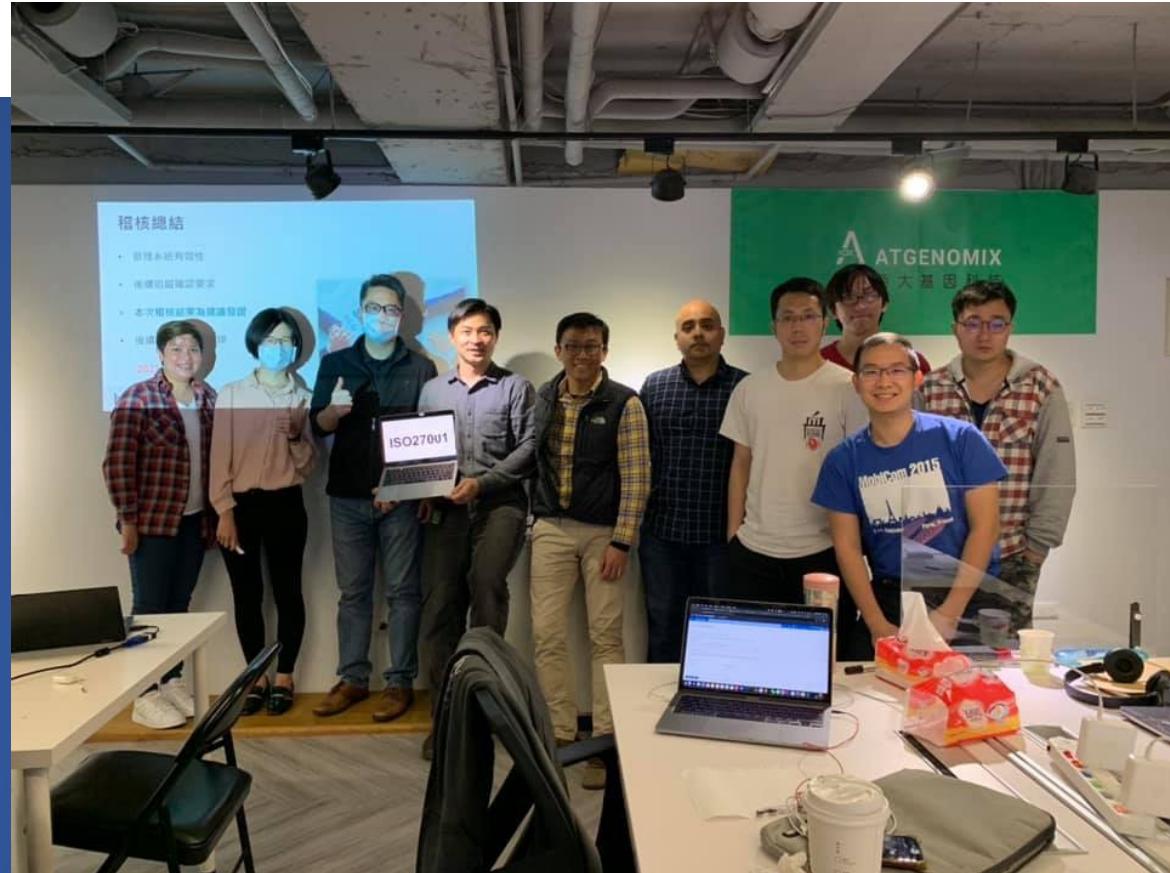
亞大基因科技

鎖定精準醫療基因體分析

運用智慧高效能基因體學大數據運算，讓全世界的研究人員與組織成就更多，以實現精準醫療的承諾。

亞大基因立足台灣，放眼國際，矢志成為亞洲最大的基因大數據分析軟體公司。

歡迎有志之士一起加入！



Sr. Data Engineer

Gold
Microsoft
Partner


2020
臺灣最
10家
TTA COOLEST STARTUPS



yuting.lin@atgenomix.com