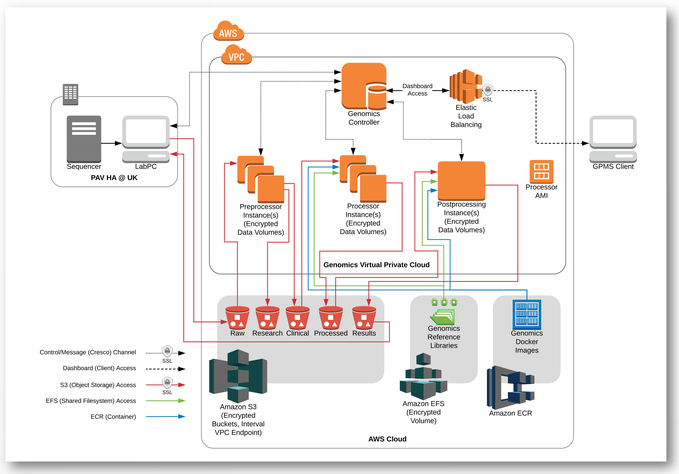
Distributed Automated Processing of Clinical Genomic Cancer Panels Leveraging Cloud Infrastructure

Caylin Hickey, V.K Cody Bumgardner, PhD, Kannabiran Nandakumar, PhD
Department of Pathology and Laboratory Medicine,
University of Kentucky, Lexington, KY

Genomics Processing at University of Kentucky Healthcare

With the recent development of next-generation sequencing (NGS,) more institutions are looking to leverage genomic sequencing for both academic research purposes as well as clinical cancer diagnostic assistance. In clinical applications, this can take the form of pipelines which translate some or all of the process by which raw base call images are used to generate annotated variant caller files for bioinformaticists to provide diagnostic to the medical community. For this work, we describe the control framework used to manage this workflow for our clinical operations.

The complex, multifaceted computational requirements of NGS require large amounts of processing power while also working on tightly controlled, mostly contained processes. As such, this system lends itself well to cloud computing infrastructure in which resources can be allocated as required. For our work, this has taken the form of both a private cloud, backed by OpenStack and Ceph, as well as the public cloud (Amazon Web Services) using encrypted architecture. Additionally, as the work must be reproducible, we leverage Docker **[1]** application containerization to precisely control the processing environment. Finally, we also leverage the BagIt **[2]** specification for archiving data developed by the Library of Congress. We collectively refer to this system as the *Genomic Processing Management System (GPMS.)*



Services utilized in AWS as well as the data paths used to transfer data between stages of processing

Bioinformatic Processing Components

Even before considering the computational requirements needed to store, transmit, and process genomic sample data, one must realize said processing in NGS requires a complex set of tools incorporating data from different reference sets and processing tools which must be customized for each individual panel being analyzed. At UK Healthcare, we have validated a number of panels for clinical analysis, including:

Solid Tumor Panel – 198 genes to identify somatic variants in solid tumors

Hematologic Panel – 97 genes to identify somatic variants in myeloid malignancies

Cardiology Panel – 203 genes with known associations to a number of heart-related malformations such as cardiomyopathies, arrhythmias, aortopathies, etc.

To process these panels, we leverage a number of commercial and open-source tools which have been tuned to minimize turn-around time while guaranteeing accurate results, including the **Broad Institute's Genomic Analysis Toolkit (GATK,) Novocraft's NovoAlign** and **NovoSort**, **Pindel**, and **Ensembl's Variant Effect Predictor (VEP.)** Additionally, a number of variant annotation databases are curated by our department such that custom annotations can be provided in the detailed analysis. In order to ensure accurate and repeatable results. These tools are packaged into scripted **Docker** containers which guarantee the same execution layer on each machine used for processing. Docker provides operating-system-level virtualization which isolates the running code inside a thinly-provisioned container which delivers applications, not hardware like virtual machines.

Leveraged Cloud Infrastructure

Identity and Access Management (IAM) – IAM allows for the generation of account-linked AWS keys (key and secret pairs) which allow for specific access to one or more services in AWS for the account to which they are linked. Individual keys have been generated to tightly control access to the subsequent systems from both GPMS agents and external users/programs.

Virtual Private Cloud (VPC) – A system which allows for virtual networks to be established in the AWS cloud environment to isolate networks of instances which require tightly controlled ingress/egress. This allows for the tight control of access to the virtual machines.

Elastic Compute Cloud (EC2) – EC2 comprises the virtual machines, or instances, on which the processing system runs. This involves two types of virtual machines. The first, the central genomic controller, handles the coordination of genomic processing, storage of historical audit data, and visual presentation to the users. The second, the genomic processing image, is a repeatable *Amazon Machine Image (AMI)* used to create isolated instances for individual processing steps.

Elastic Block Storage (EBS) – These are the storage volumes attached to the EC2 instances. The root drives, which house the operating system and GPMS orchestration code, are normal volumes while the drives used to store and process the actual data are encrypted.

Elastic Container Registry (ECR) – A hosted Docker registry service that allows for the hosting of Docker images used in processing, with access controlled by IAM configured keys.

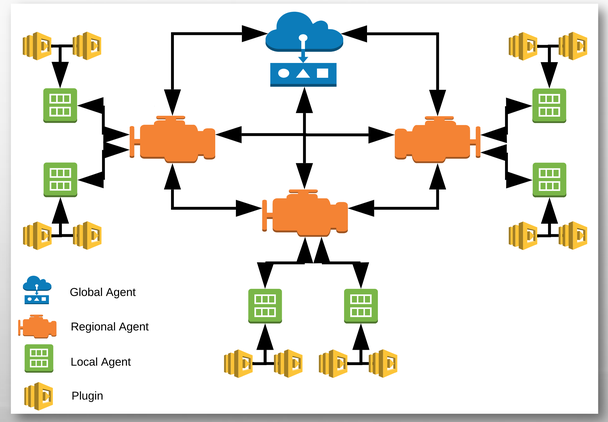
Elastic File System (EFS) – This is a shared file system which houses the reference data the bioinformatic Docker containers use in processing of flow cells and samples. Access to this system is controlled by the security groups and residing VPC.

Simple Storage Service (S3) – S3 is used to house the genomic data between stages of processing. All data stored in S3 is stored in buckets encrypted to the main AWS account. All access to S3 from GPMS agents occurs over SSL and/or via internal endpoints inside the VPC. The data is stored in an archival format known as BagIt developed by the Library of Congress which keeps as part of its packaging the MD5 hashes of all files stored within as well as hashes of the inventory/hash lists themselves to ensure no corruption has occurred. This is then packaged in a tape archive (TAR) tarball archive file format for easier storage management.

Elastic Load Balancing (ELB) – Load balancing allows for the control of secure access to visual components of the controller. In this system, SSL access as well as port control is provided by utilizing this load balancing layer.

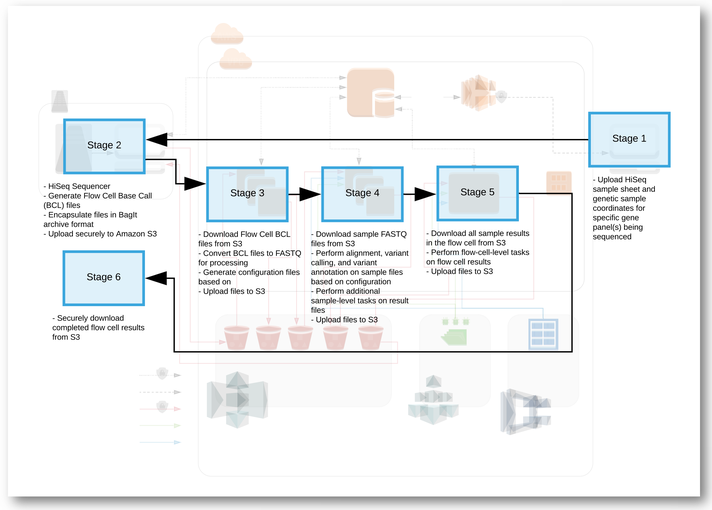
Cresco Edge-Computing Control Framework

Orchestration of distributed tasks is a deep field in informatics that has resulted in many control frameworks by which control of information flow and processing can be managed. One such framework, the Cresco framework **[3]**, is an actor-model system leveraging the edge-computing paradigm resulting from separate research by a number of the contributors to this system. In it, agents organize themselves in a hierarchy used to monitor system health and performance as well as the status of distributed tasks being run. Agents provide secure, low-latency communication and managed task execution, optimal task placement and scheduling, and overall system performance and health monitoring.



Brief outline of the steps involved in the processing of genomic panels and how that information flows to the cloud infrastructure and back

Brief Overview of Processing Stages



Brief outline of the steps involved in the processing of genomic panels and how that information flows to the cloud infrastructure and back

Computational Processing Components

Controller - The GPMS controller combines an administrative dashboard with a central process controller used to coordinate the activities of the other agents used in delivery and processing of genomic data. Agents report their activities to the controller via a secure communications channel provided by the Cresco framework. From there, various stages of work are performed by AWS instances that are generated by, and disposed of by, the controller using established AMIs and configuration parameters specific to the processing required. This information is stored in a local databased on a long-running instance in the same VPC in which the files are processed. Administrators can view the progress of the processing of genomic data and perform some limited operations in the event of unforeseen malformations in the data or processing, with systems to record the reason for such deviations as well as who performed them. There is also limited reporting via email of the progress of processing, which will continue to be improved, along with many other reporting and summarization systems in subsequent updates.

Raw File Uploader - The agent watches for new completed flow cells to arrive off the sequencer, uploads them to AWS, and informs the GPMS controller of its progress.

Processor - This agent resides on AWS instances in the AWS cloud environment. These processor agents run specific tasks depending on which stage of processing they are working on, which they are informed of by the controller at their genesis. The tasks generally involve downloading and restoring the previous stage's files from S3, running a specified Docker container on those files, possibly some additional tasks, and finally archiving the results and uploading those back to S3 for the next stage.

Results

Original, manual processing on hematology panels took on the order of days to weeks, due to the grouping of samples groups into flow cells, then serialized for further processing and annotation. Initial automation efforts described in **[4]** reduced by parallelizing sample processing on a local OpenStack cluster, but scales to only a single flow cell at a time. Moving to AWS reduces both the individual processing time, via variable instances size, as well as parallelizing entire flow cell processing such that results can be return on the order of hours.

References

1. Boettiger, C. (2015). An introduction to Docker for reproducible research. ACM SIGOPS Operating Systems Review, 49(1), 71-79.
2. Kunze, J., Littman, J., Madden, E., Scancella, J., & Adams, C. (2018). The bagIt file packaging format (v1. 0) (No. RFC 8493).
3. Bumgardner, V. C., Marek, V. W., & Hickey, C. D. (2016, October). Cresco: A distributed agent-based edge computing framework. In 2016 12th International Conference on Network and Service Management (CNSM) (pp. 400-405). IEEE.
4. Bumgardner, V. C., Marek, V. W., Hickey, C. D., & Nandakumar, K. (2016, September). Constellation: A secure self-optimizing framework for genomic processing. In 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom) (pp. 1-6). IEEE.