Computational Molecular Biology 2025, Vol.15, No.3, 151-159
http://bioscipublisher.com/index.php/cmb

Review Article Open Access

High-Performance Computing Pipelines for NGS Variant Calling

Wenzhong Huang
Biomass Research Center, Hainan Institute of Tropical Agricultural Resouces, Sanya, 572025, Hainan, China
Corresponding author: wenzhong.huang@hitar.org

Computational Molecular Biology, 2025, Vol.15, No.3 doi: 10.5376/cmb.2025.15.0015
Received: 18 Apr., 2025
Accepted: 29 May, 2025
Published: 21 Jun., 2025

Preferred citation for this article:
Huang W.Z., 2025, High-performance computing pipelines for NGS variant calling, Computational Molecular Biology, 15(3): 151-159 (doi: 10.5376/cmb.2025.15.0015)

Abstract With the popularization of high-throughput sequencing (NGS) technology, genomic sequencing data have grown exponentially, posing severe computational challenges for variant detection. Traditional mutation detection processes (such as GATK-based pipelines) are prone to computational bottlenecks and I/O bottlenecks when dealing with large-scale data. This paper reviews the high-performance computing (HPC) processes for NGS mutation detection, introduces the typical workflows and commonly used algorithms of NGS mutation detection, and analyzes the performance bottlenecks of traditional processes. Subsequently, the application of the architecture of HPC and the parallel computing model in bioinformatics was expounded. On this basis, the HPC optimization strategies for the mutation detection process were mainly discussed, including task parallelization, I/O optimization, data locality management, and the methods of workflow orchestration using middleware such as SLURM, Nextflow, and Cromwell. This paper introduces the application of emerging hardware acceleration technologies such as GPU and FPGA in mutation detection, discusses performance evaluation metrics and benchmark testing frameworks, as well as a comparative study of HPC-driven processes and traditional methods.

Keywords High-performance computing; Mutation detection; Next-generation sequencing; Parallel computing; Workflow

1 Introduction

With the decline in sequencing costs, NGS has become the main tool for studying genetic differences, and it is used in humans, animals and plants. It no longer relies on a small number of samples or lengthy experimental steps as it did in the past. Now, it can generate massive amounts of genomic data in a very short time. In this way, various changes - such as SNPS and indels - can be quickly captured. These changes are often related to disease risks or drug responses, which is precisely the key to precision medicine. The general process is actually quite fixed: first, quality control is carried out; then, sequencing data is aligned to the reference genome; next, variations are called up; and finally, annotations are made. Names like GATK, DeepVariant, and FreeBayes often appear in such processes. Although everyone is doing the same thing, their methods vary greatly - some are still using statistical models, while others have shifted to deep learning. The results are also quite interesting. For example, tools based on neural networks like DeepVariant tend to be more accurate than old-fashioned methods when detecting SNPS and indels (Pei et al., 2021).

However, problems also arise: the volume of data is simply too large. The growth rate of NGS data is almost beyond imagination. Once it rises to the scale of tens of thousands of whole genomes, the computing demand snowballes. Ahmedet al. Even compared genomics with astronomy and physics, saying that current research has had to migrate from traditional HPC systems to the cloud in order to survive. After all, using GATK to handle SNP detection for 3,000 crop genomes might take half a year, which clearly no one can afford to wait. Zhou et al. (2023) also noticed this. They developed a new system called HPC-GVCW, which runs on the cluster and is much faster than the traditional process.

So, it actually makes little sense to talk about whether high-performance computing is "needed" now - it is necessary. Distributing tasks across multiple nodes, processing them in parallel, or even leveraging hardware acceleration are all ways to make analysis faster. HPC is designed for speed and scale: as the number of nodes increases, performance can almost grow linearly. Although the cloud platform has different ideas, its purpose is similar - you don't need to worry about the hardware, and resources can be expanded as needed. No matter which

approach is adopted, HPC remains the core pillar of large-scale genomic analysis and the underlying driving force for the true implementation of NGS variant detection (Ahmadet al., 2021).

2 Overview of the NGS Mutation Detection Workflow

2.1 Overall architecture of the mutation detection process

In the matter of NGS mutation detection, each step is not actually independent; they are interlinked. Usually, researchers only obtain the original sequencing fragments. The first step is to conduct quality control. For example, use FastQC to check the data and remove those sequences of poor quality or contamination. Next, alignment software like BWA-MEM will be used to align these clean reads onto the reference genome and generate sorted BAM files. True mutation detection comes after this, using GATK HaplotypeCaller or DeepVariant (Poplin et al., 2018) to look for variations such as SNPS and indels. The detection is not over yet. The results still need to be screened, labeled, and finally the variations are displayed through visualization tools. The entire process seems linear, but in fact, each step may in turn affect the quality of the previous ones.

2.2 Common tools and algorithms

When it comes to mutation detection software, GATK is a familiar face. It uses statistical models, such as logistic regression or hidden Markov models, to determine the differences in the sequence. However, AI has now stirred up this field. DeepVariant directly uses deep neural networks to view "stacked images" and identify variations, which is often more accurate than traditional algorithms. Tools such as FreeBayes, Strelka, and Platypus are also frequently used to detect different types of variations. In pursuit of speed, many commercial platforms have begun to implement multi-threading, GPU, and even FPGA acceleration. However, such optimizations are often costly, especially when dealing with big data.

2.3 Performance bottlenecks of traditional processes

Performance issues are almost a persistent problem in all traditional processes. The computational burden is the heaviest during the alignment and mutation invocation phases. For example, when running GATK with a CPU, it may take several hours or even days for one sample. When there are more samples, the time doubles. Moreover, just the I/O operations are troublesome enough - frequent reads and writes during BAM file sorting and VCF generation will seriously slow down the speed (Costa et al., 2018). Although DeepVariant has a high accuracy rate, if it is not processed in parallel, the computing power consumption is also quite astonishing (Yang et al., 2020). To break the deadlock, relying solely on optimizing algorithms is not enough; support at the hardware level must also keep up. Only distributed computing based on Apache Spark, or directly on HPC clusters (Alganmi and Abusamra, 2023), is possible to truly alleviate these performance bottlenecks.

3 Fundamentals of High-Performance Computing in Bioinformatics

3.1 HPC architecture: clusters, grids, and cloud systems

When it comes to high-performance computing, many people's first reaction is "clusters". But in fact, that's just one form. An HPC cluster is composed of many computing nodes connected through a high-speed network and is suitable for tasks that can be split into multiple parts and run simultaneously. Grid computing, on the other hand, takes a different path-it does not rely on a single institution or location, but connects computing resources scattered in different places for use. Then comes cloud computing, which is more flexible. Users don't have to worry about the underlying machines, and resources can be increased or decreased at any time. The overall goal is actually the same: to fully exploit the potential of computing power. For example, Munhoz et al. (2023) compared local HPC clusters and cloud clusters and found that CPU-intensive tasks performed similarly, but applications with frequent communication were still limited by network speed on the cloud. However, the idea of the cloud is quite different - it is more like providing a kind of "service", hiding the hardware behind the scenes and offering users an environment that can be extended and retracted at any time. In reality, these architectures are not clearly distinct. Many systems mix local clusters and cloud resources to strike a balance between cost and performance.

3.2 Parallel computing models: shared memory, distributed memory, and hybrid methods

In parallel computing, no single model can solve all problems. The shared memory model is the most intuitive. Multiple threads share data on the same node, which has low overhead and high speed. However, once the task is

too large, the memory of a single node cannot handle it. Thus, distributed memory models emerged, such as MPI or Spark, which distribute tasks and data to run on multiple nodes. This model is particularly suitable for those extremely large problems that require extensive machine collaboration. There is also a compromise solution - a hybrid model, such as MPI+OpenMP, which combines the parallelism within and between nodes. Gpus and other accelerators are also playing an increasingly important role in this system. They can break tasks down into tens of thousands of small pieces for simultaneous processing. Take NGS analysis as an example. Chromosomes or samples can be allocated to different nodes, and each node can then accelerate internal computing through multi-threading or GPU. This hierarchical parallel approach not only leads to higher throughput, but also enables smoother scaling when the data scale expands.

3.3 High-performance computing environment

In an HPC environment, who decides when and where jobs run? This depends on the resource management system and the workflow management system. WfMS such as Nextflow, Snakemake, and Cromwell (in conjunction with WDL) can break down complex tasks, automatically handle dependencies, and then hand them over to the scheduler for scheduling (Jha et al., 2022). In large-scale genomic projects, these systems can operate across clusters, grids, and the cloud, keeping cumbersome processes efficient and consistent. Nowadays, containerization and version control are also often integrated into WfMS, making it easier to reproduce scientific research work and facilitating migration to different computing platforms.

4 Optimization Strategies for the HPC Mutation Detection Process

4.1 Parallelization of workflow components and task decomposition

When conducting mutation detection on an HPC system, if the task is not broken down first, it often leads to the predicament of "slow calculation and long waiting time". The HPC-GVCW pipeline is a typical example - instead of processing the entire genome as usual, they first used the "genome index splitter" to slice the reference genome into several large blocks, and each block was handed over to an independent GATK instance to run. In this way, different genomic regions and even different chromosomes can be analyzed simultaneously, and the speed increases immediately. Later, many studies also verified the effectiveness of this train of thought. The Humming Bird pipeline designed by Liu et al. (2020) adopts this decomposition strategy. The result is much faster than the traditional BWA-GATK without losing accuracy. Mulone et al. (2023) used Stream Flow to divide NGS tasks in hybrid cloud and HPC environments and also achieved nearly linear scaling effects. In fact, the principle is straightforward: those time-consuming stages such as alignment, marking repetition, and variant invocation, as long as they can be broken down into small tasks that can be executed independently, can be done in parallel, and the efficiency will naturally be high. Modern workflow management systems like Nextflow and Snakemake inherently support sample-level and chromosomal task parallelism. As long as data dependencies are sorted out, a considerable amount of overall time can be saved and the consistency of results can also be guaranteed.

4.2 I/O optimization and data locality management

No matter how strong the computing power is, if I/O cannot keep up, the process still cannot run smoothly. The bottleneck often does not lie in the CPU, but in storage and network. When data traffic exceeds the limit of the shared file system, alignment and mutation calls are most likely to encounter problems. So the researchers began to think of ways to optimize from the file system and cache. Parallel file systems like Lustre and GPFS have begun to be widely adopted. Some people even directly use local SSDS of nodes for caching to minimize data transmission latency. The CloudGT framework (Xiao et al., 2018) is an example. It implements parallel multi-node execution with Apache Spark and stores data in parquet format. The I/O overhead is reduced by more than half and the speed is increased by 74.9%. Another study found that using local caching or in-memory transfer can significantly reduce intermediate file writes, alleviate disk contention, and make the cluster as a whole more efficient. Moreover, the old principle of "computing data nearby" still works well - task scheduling should be as close as possible to the data source, so that the network burden is much lighter. The Hadoop and Spark systems have long proved this point. Overall, only by integrating the file system, memory buffering and scheduling strategies can the HPC cluster maintain stable I/O performance in the parallel processing of tens of thousands of samples.

4.3 Use HPC middleware (such as SLURM, nextflow, cromwell) for workflow orchestration

To run thousands of tasks simultaneously and still finish them in an orderly manner, coordination is no simpler than calculation. Workflow middleware comes in handy precisely in this regard. Systems like Nextflow and Cromwell (with WDL) enable researchers to write the script for each step first and then hand over the scheduling to the underlying system, such as SLURM or PBS. Users don't have to keep an eye on the job themselves, manually retry or log. The workflow engine will automatically handle these cumbersome tasks. Cullen et al. once reported that WAGS pipelines combined with SLURM can efficiently manage thousands of NGS samples; Mulone et al. (2023) found that StreamFlow performs more flexibly in a hybrid HPC and cloud environment, and its performance is no worse than that of Snakemake. Recently, automated assessment tools such as RecallME and Benchmarker have also been added to continuously monitor pipeline performance, making the process more stable and reproducible. What is more valuable is that this type of middleware has high versatility. The workflow defined declaratively can be directly run regardless of whether it is placed in a local cluster, a national HPC center, or the cloud. With the containerization support of Docker or Singularity, the environment is unified and the versions are consistent, making the analysis results more reliable. It can be said that with the help of these tools, NGS analysis can finally be smoothly expanded from small laboratories to large-scale, cross-institutional genomic projects.

5 Emerging Technologies for Enhancing Performance

5.1 Accelerated application of GPU and FPGA in mutation detection

Hardware acceleration has become a hot topic in NGS analysis in recent years. Illumina's DRAGEN platform is a typical example, which uses FPGA chips to run genome alignment and variant invocation. The test results show that its speed is ten to thirty times faster than that of traditional CPU tools (Arram et al., 2017). NVIDIA took a different route - Clara Parabricks directly moved the GATK workflow onto the GPU cluster. Early tests showed that it could increase the speed by about 35 times. Later, with the release of more powerful Gpus, this figure rose to 65 times. Some studies have also found that GPU-accelerated mutation detection can achieve an overall speed increase of 10 to 66 times, and the accuracy rate can still remain above 99% (Vats et al., 2022). The performance of the FPGA platform is also not weak. It can be about 28 times faster than the CPU in comparison and mutation analysis (Guo et al., 2019). Of course, hardware is not omnipotent. Costs, compatibility and development difficulties still exist, but it has indeed brought about a qualitative leap in the running speed of traditional processes.

5.2 Containerization and repeatability: docker, singularity and CWL

Installing bioinformatics tools on different HPC systems is a headache-inducing task. The emergence of containerization technology has somewhat alleviated this chaos. Docker, Singularity, and standardized description methods like the Common Workflow Language (CWL) have made environment configuration and tool deployment much simpler. For large-scale projects like WARP and Terra, directly binding WDL workflows with Docker images results in smooth switching between local clusters and cloud platforms. Singularity has become the "default choice" in HPC because it does not require administrator privileges and can ensure consistent operation across nodes. Research shows that whether using Nextflow + Docker or CWL + Singularity, portability and repeatability can be enhanced without sacrificing performance - even in large-scale tasks involving exome or whole genome. These standardized approaches make cross-platform workflows less prone to "errors", which is particularly important for clinical-level process validation.

5.3 Integration with cloud high-performance computing and hybrid computing models

Cloud computing has completely rewritten the concept of scalability. In the past, when people talked about HPC, they always thought that they had to rely on expensive local clusters. Nowadays, many research teams tend to "combine" - using local HPC and cloud resources together rather than completely replacing them. Mulone et al. (2023) presented an example where they used StreamFlow to integrate local clusters with the cloud, optimizing both cost and flexibility while maintaining performance. The same team also proposed a heterogeneous cloud model integrating GPU, TPU and FPGA to dynamically allocate tasks through performance prediction. Today's cloud service providers - AWS, GCP, Azure - all offer accelerated instances of Gpus and FPgas, often equipped

with parallel file systems or object storage to facilitate data transmission. In addition to features such as automatic scaling and serverless orchestration, bioinformatics pipelines can freely switch between the cloud and HPC. It can be said that hybrid high-performance computing is no longer just an experimental attempt, but is gradually becoming the norm in large-scale genomic research.

6 Performance Evaluation and Benchmarking

6.1 HPC process evaluation indicators

When evaluating high-performance computing pipelines, what people care about is not just "whether they run fast or not". Speed is of course important, but metrics such as throughput, scalability and cost often need to be considered together. Throughput refers to the number of samples that can be processed within a certain period of time, while scalability reflects the extent of performance improvement after adding resources. The problem is that resources double but the speed does not necessarily double accordingly. This situation is quite common in tests (Carrier et al., 2015). Ahmed et al. Compared the performance of different workflow management systems in the mutation detection task, and the results showed that there were significant differences among the systems in terms of scalability and stability. When it comes to cost, sometimes a lot of computing power and time are spent, but the performance improvement is not proportional. At this point, it is necessary to consider whether it is cost-effective. In an HPC environment, people will also look at detailed indicators such as scheduling efficiency, resource occupancy rate, and I/O bandwidth. When it comes to the cloud, the accounts become even more complicated - apart from computing power, the rental duration, data transmission and storage costs also need to be taken into account. Only by comparing all these dimensions together can we truly see which solution is "more worthwhile", rather than simply focusing on the speed.

6.2 Benchmark datasets and standard test frameworks

When it comes to testing, the approaches of different teams vary greatly, but the most commonly used ones are still those reference samples with "true value". The "Genome in a Bottle" (GIAB) project is a recognized standard that is most suitable for evaluating the accuracy of SNP and Indel detection. NA12878 is one of the most beloved samples and has almost become a "touchstone" for detection accuracy. In addition, the data from the Illumina Platinum Genome and 1000 Genome projects are also often used as controls. Some researchers also use simulated data or manually sliced datasets to test the performance of the algorithm at different scales. The key is not where the data comes from, but whether the output results can be compared with the standard true value to calculate the missed detection rate and false detection rate. Only in this way can we clearly see where the shortcomings of the algorithm lie.

6.3 Comparative research process of HPC empowerment and traditional methods

When it comes to the advantages of HPC, many comparative experimental results are actually quite intuitive. For instance, in the HPC-GVCW experiment by Zhou et al. (2021) (Figure 1), they conducted SNP detection on 25 rice genomes simultaneously in just 120 hours, while the traditional process would take half a year. The GPU-accelerated NGS pipeline has also undergone similar benchmark tests, and its speed is 65 times faster than that of CPU tools. The research results further indicate that both FPGA and GPU far exceed traditional CPU in terms of speed and scalability. Although hardware investment is indeed not cheap, the trend is already quite clear - high-performance computing is turning mutation detection from a "months-long project" into a "hours-long task". This performance gap has also made HPC an almost indispensable core technology in large-scale genomic research.

7 Case Study

7.1 Description of computing infrastructure and datasets

In this case, we did not build the computing environment from scratch but directly used two mature HPC platforms. The data was selected from the high-coverage samples of the "Thousand Genomes Project", totaling 98 individuals. The depth of the whole genome sequencing was approximately 30×, and after compression, it was about 632GB. The CloudLab cluster configuration is relatively higher, with 8 nodes. Each node is equipped with 40-core CPU, 192GB memory and NVIDIA Tesla P100 GPU. Although the Fabric environment also has 8 nodes,

each node only has a 24-core CPU and 128GB of memory, but it has two additional NVIDIA T4 Gpus. Both are equipped with parallel file systems and SSD storage to cope with frequent read and write operations (Dongarra et al., 2020). Although the GPU models and storage performance are different, both systems can run multiple mutation detection tasks simultaneously. The addition of GPU also enables us to embed deep learning modules into the analysis process, which is not common in traditional HPC environments.

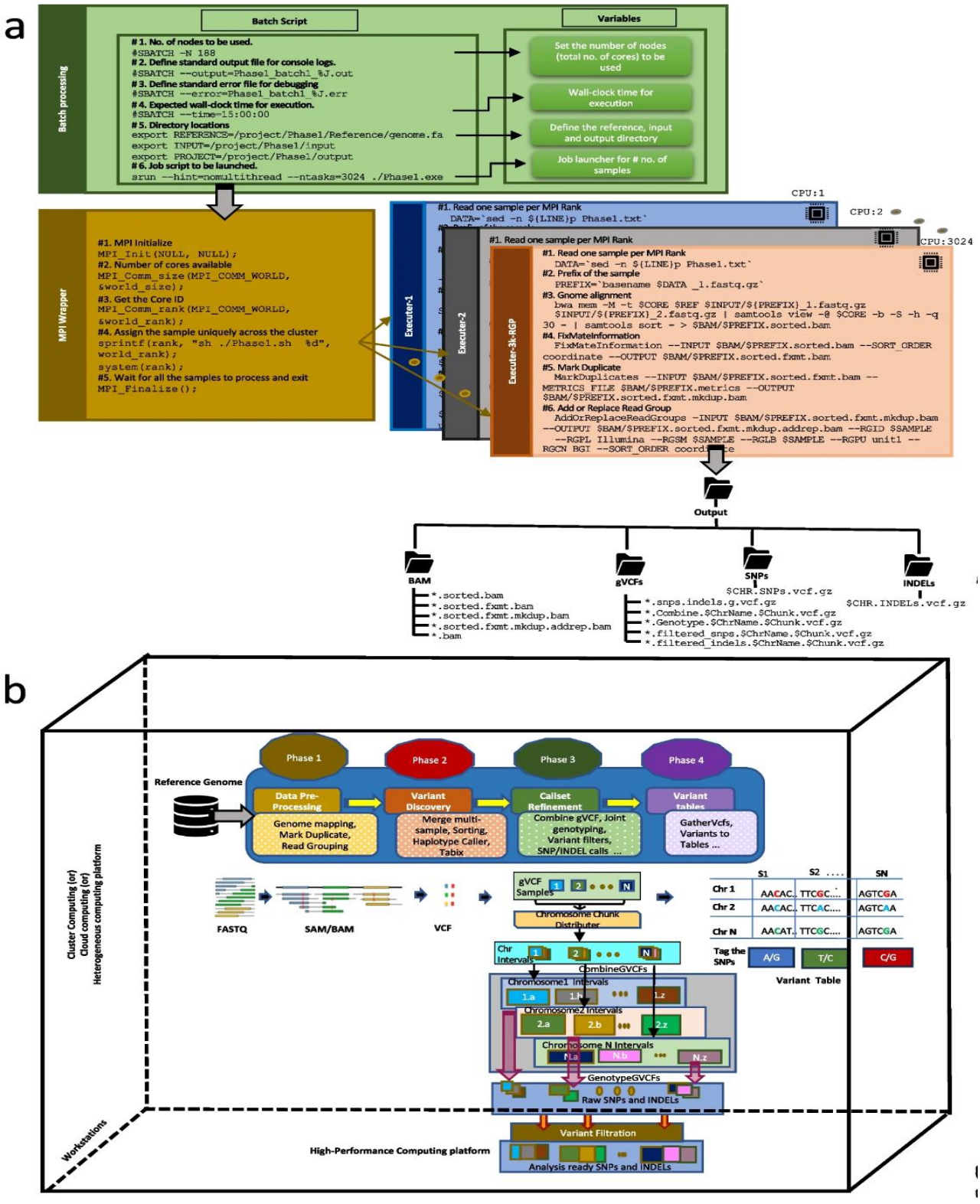

Figure 1 Automated and flexible genome variant calling workflow (GVCW) design for a HPC systems and b diversified system architectures

7.2 Process design and optimization technology

The construction of the workflow is based on the avah★ framework, which is a heterogeneous parallel architecture where both the CPU and GPU are involved in the computing. Instead of stringing all the steps into a straight line, we designed it as an asynchronous scheduling mode, allowing tasks such as comparison, sorting, and mutation calls to run separately. Tasks are allocated through queues to avoid wasting resources due to waiting. When executed concurrently, mutual exclusion control is also added to prevent I/O conflicts caused by multiple nodes accessing the same file simultaneously. Before entering the process, data is split into chromosome or specific region levels, with each node responsible for a part of it. This sharding strategy significantly improves the utilization rate of the cluster. The GPU is mainly responsible for computationally intensive tasks such as GVCF compression and deep learning prediction. The Slurm scheduler is responsible for resource allocation. It automatically detects idle nodes and assigns tasks. Although the entire system seems complex, it is much smoother than the traditional serial process, with significantly reduced waiting time and I/O contention.

7.3 Performance results, lessons learned and future improvement directions

Judging from the results, the acceleration effect brought by the GPU version is quite obvious: the speed is increased by approximately 3.67 times in the CloudLab environment, and even reaches 5.05 times in the Fabric environment. The most obvious improvement is in the comparison and initial mutation invocation stages, but not all parts can benefit equally. In CloudLab, the read and write speeds of storage are relatively slow, and the I/O bottleneck still exists; But after Fabric was replaced with a faster NVMe SSD, this problem almost disappeared. The simultaneous operation of multiple Gpus can maintain a high utilization rate, which further verifies the effectiveness of the parallel pipeline design. Overall experience shows that while computing speed is important, the balance between computing and I/O is even more crucial. Asynchronous scheduling can make resource allocation more reasonable, while GPU hybrid acceleration significantly improves the processing efficiency of the deep learning process. Next, we are considering trying AI-driven automatic scheduling to make task allocation smarter. Meanwhile, explore the dynamic expansion of the cluster to enable the number of nodes to adjust according to the load changes. With the gradual popularization of long-read sequencing technology, how to further optimize its performance on heterogeneous clusters will also be the problem to be solved in the next step.

8 Future Development Directions and Conclusions

Artificial intelligence is gradually changing the role of high-performance computing in bioinformatics. In the past, people paid more attention to improvements at the algorithmic level, such as how to make comparisons faster and models more stable. Now, the situation is somewhat different - deep learning is beginning to enter the mutation detection process. Tools like DeepVariant have proven that neural networks, after training, can indeed significantly improve detection accuracy. Next, AI may not remain confined to a single model but permeate the entire analysis process, adopting an end-to-end approach from input to output, and leveraging GPU clusters to achieve faster inference. In fact, AI is not only capable of "detection", but is also gradually "managing systems". Some teams have already used reinforcement learning to dynamically adjust task scheduling and cloud resource allocation, enabling the cluster to remain efficient under different loads. Although these practices are not yet mature, the trend is already quite clear - the combination of AI and HPC is making NGS analysis more automatic, smarter and less troublesome. However, new problems brought about by the improvement of computing power have also emerged. Scalability and energy consumption have become realities that cannot be ignored.

The scale of sequencing is constantly expanding, and the petabyt-level data is too much for many systems to handle. The problem is not only in computing power; storage, networking, and heat dissipation have all become new bottlenecks. As clusters grow larger and larger, energy consumption is soaring, and energy conservation and heat dissipation have become unavoidable issues. How to complete more computations without increasing power consumption is one of the current research focuses. On the other hand, data management remains an old problem that slows down progress - compression, transmission and access may all become bottlenecks. In a cloud HPC environment, the situation is more complex. Considerations of security and privacy make system design more challenging. To address these issues, researchers are exploring new directions: using scalable file systems, hardware heterogeneous integration (such as FPgas, ASics), and low-power architectures to balance performance and energy efficiency. But to achieve a truly "green HPC", it may still take more time.

Looking back, the value of high-performance computing in NGS mutation detection is already quite clear. Traditional pipelines are constrained by computing and I/O bottlenecks. Once the volume of data increases, their efficiency drops significantly. The introduction of parallel computing, hardware acceleration and intelligent scheduling has brought about a qualitative change to the entire situation. Strategies such as task decomposition, asynchronous scheduling, and I/O optimization have significantly increased processing speed. The use of Nextflow and SLURM makes the management and migration of processes simpler. GPU, FPGA acceleration, containerized deployment, and cloud integration further enhance the repeatability and scalability of the process. The existence of standard datasets such as GIAB also makes performance comparisons among different studies more grounded. Our practical experience also demonstrates this point - processes that used to take several days to complete can now be finished in just a few hours. Looking ahead, the emergence of AI-driven automatic scheduling and new hardware architectures will make the role of high-performance computing in the field of genomics even more prominent. Perhaps it won't be long before a more efficient and automated NGS analysis system truly becomes the norm.

Acknowledgments

The author extends sincere thanks to two anonymous peer reviewers for their invaluable feedback on the manuscript.

Conflict of Interest Disclosure

The author affirms that this research was conducted without any commercial or financial relationships that could be construed as a potential conflict of interest.

References

Ahmad T., Al Ars Z., and Hofstee H.P., 2021, VC@Scale: scalable and high-performance variant calling on cluster environments, GigaScience, 10(9): giab057.
https://doi.org/10.1093/gigascience/giab057

Alganmi N., and Abusamra H., 2023, Evaluation of an optimized germline exomes pipeline using BWA-MEM2 and Dragen-GATK tools, PLoS One, 18(8): e0288371.
https://doi.org/10.1371/journal.pone.0288371

Arram J., Kaplan T., Luk W., and Jiang P., 2017, Leveraging FPGAs for accelerating short read alignment, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 14(3): 668-677.
https://doi.org/10.1109/TCBB.2016.2535385

Carrier P., Long B., Walsh R., Dawson J., Sosa C., Haas B., Tickle T., and William T., 2015, The impact of high-performance computing best practice applied to next-generation sequencing workflows, bioRxiv, 2015: 017665.
https://doi.org/10.1101/017665

Costa C.H.A., Misale C., Liu F., Silva M., Franke H., Crumley P., and D'Amora B.D., 2018, Optimization of genomics analysis pipeline for scalable performance in a cloud environment, In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, pp.1147-1154.
https://doi.org/10.1109/BIBM.2018.8621208

Dongarra J., Luszczek P., and Petitet A., 2003, The LINPACK benchmark: past, present, and future, Concurrency and Computation: Practice and Experience, 15(9): 803-820.
https://doi.org/10.1002/cpe.728

Guo L., Lau J., Ruan Z., Wei P., and Cong J., 2019, Hardware acceleration of long read pairwise overlapping in genome sequencing: a race between FPGA and GPU, In: 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), IEEE, pp.127-135.
https://doi.org/10.1109/FCCM.2019.00027

Jha S., Pascuzzi V.R., and Turilli M., 2022, AI-coupled HPC workflows, arXiv Preprint, 2208: 11745.
https://doi.org/10.48550/arXiv.2208.11745

Liu J., Wu X., Zhang K., Liu B., Bao R., Chen X., Cai Y., Shen Y., He X., Yan J., and Ji W., 2020, Computational performance of a germline variant calling pipeline for next generation sequencing, arXiv Preprint, 2004: 991.

Mulone A., Awad S., Chiarugi D., and Aldinucci M., 2023, Porting the variant calling pipeline for NGS data in cloud-HPC environment, In: 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), IEEE, pp.1858-1863.
https://doi.org/10.1109/COMPSAC57700.2023.00288

Munhoz V., Bonfils A., Castro M., and Mendizabal O., 2023, A performance comparison of HPC workloads on traditional and cloud-based HPC clusters, In: 2023 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW), IEEE, pp.108-114.
https://doi.org/10.1109/SBAC-PADW60351.2023.00026

Pei S., Liu T., Ren X., Li W., Chen C., and Xie Z., 2021, Benchmarking variant callers in next-generation and third-generation sequencing analysis, Briefings in Bioinformatics, 22(3): bbaa148.
https://doi.org/10.1093/bib/bbaa148

Poplin R., Chang P., Alexander D., Schwartz S., Colthurst T., Ku A., Newburger D., Dijamco J., Nguyen N., Afshar P., Gross S., Dorfman L., McLean C., and DePristo M., 2018, A universal SNP and small-indel variant caller using deep neural networks, Nature Biotechnology, 36(10): 983-987.
https://doi.org/10.1038/nbt.4235

Vats P., Sethia A., Samadi M., and Harkins T., 2022, Rapid variant detection and annotations from next-generation sequencing data using a GPU-accelerated framework, Cancer Research, 82(12_Supplement): 1900-1900.
https://doi.org/10.1158/1538-7445.AM2022-1900

Xiao A., Dong S., Liu C., Zhang L., and Wu Z., 2018, CloudGT: a high-performance genome analysis toolkit leveraging pipeline optimization on Spark, In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, pp.1343-1350.
https://doi.org/10.1109/BIBM.2018.8621495

Yang C.H., Zeng J.W., Liu C., and Hung S.H., 2020, Accelerating variant calling with parallelized DeepVariant, In: Proceedings of the International Conference on Research in Adaptive and Convergent Systems, pp.13-18.
https://doi.org/10.1145/3400286.3418243

Zhou Y., Kathiresan N., Yu Z., Rivera L.F., Thimma M., Manickam K., Chebotarov D., Mauleon R., Chougule K., Wei S., Gao T., Green C., Zuccolo A., Ware D., Zhang J., McNally K., and Wing R., 2023, HPC-based genome variant calling workflow (HPC-GVCW), bioRxiv, 25: 546520.
https://doi.org/10.1101/2023.06.25.546420

Zhou Y., Wu Z., and Dong S., 2021, ADS-HCSpark: a scalable HaplotypeCaller leveraging adaptive data segmentation to accelerate variant calling on Spark, BMC Bioinformatics, 20(1): 76.
https://doi.org/10.1186/s12859-019-2665-0