Biolution

System Design Documentation

Sabina Mammadova

Principles of Distributed Systems

01/03/2021

Contents

Overview

Since 1950s, after when a scientist Rosalind Franklin took a picture of a DNA, it took 40 years to finally poke inside a human cell, take out the crystal from chromosome, unroll, and read it for the first time. The genetic code came out to be simple alphabet letters: A, T, C, G; for building a human, 3 billion of those letters are needed. However, only about 5 million out of this amount makes us different from one another, the rest is absolutely identical.

We believe in the revolution of technology, every newly released product will be the most sophisticated of its generation, but we perceive our biology as fixed. Why? With the help of the genetic revolution tools and Big Data analysis, our biology can be fundamentally transformed. We will see its effect on healthcare (moving from precision to predictive medicine as changes in human genome are highly likely to cause pathological conditions), understanding mostly/partly genetic characteristics (how about knowing the tremendous potential of a child in art beforehand so that the parent would not force him/her to become a doctor), transformation in the way of our species' reproduction (IVF – ranking of embryos from most likely genetic component of IQ to least likely).

To analyze and gain reliable scientific insights out of DNA sequence requires not only laboratory investigation, but also precise bioinformatic analysis. The solution is having HTC grid system distributed across the world containing enormous computational power and storage. Suggested infrastructure, processing of datasets, communication between nodes, naming of grids, and coordination are explained below.

Note: How the DNA sequence are brought into the ready state for processing using which functional genome analysis techniques in the field of epigenomics, transcriptomics, proteomics, interactomics, model systems (chemical mutagenesis, CRISPR-Cas9), and variants detection is in the scope of the project but out of system design documentation.

Infrastructure

Since we are stepping into genome analysis, at first phase, it might be convenient to collect one blood sample from different locations of the world for category – climate change (to evaluate how immune system is established according to weather conditions, viruses) and subcategories: gender, age scale, marital status, eating habits, lifestyle and retrieve useful information after preprocessing. However, it will bring challenges over time because more blood samples will be needed for each subcategory to obtain more accurate results and the system having insufficient CPU and storage will become a bottleneck for high throughput computing; therefore, infrastructure of the system is designed as a distributed grid computing system where it allows for the secure integration of computer systems over high speed networks to provide on-demand access to large chunks of data processing and analysis capabilities.

Coordination

As grids used for complex applications increase from 10s to 1000s of nodes, their functionalities should be decentralized to avoid congestions. The P2P model can support dynamic grid scalability as well as be particularly useful to manage the most significant services that characterize grid computing environment: membership and resource coordination. The objective of membership covers 2 functionalities: adding a new node to the network and assigning this node a set of neighbor nodes. The proposed style to maintain a balance between inherent efficiency of centralized structure, and the autonomy, load balancing and fault-tolerant features offered by distributed structure is adjusting super-peer networks. A super-peer node acts as a centralized resource for several weak peers, while super peers connect to each other to form overlay P2P mechanism on high level.

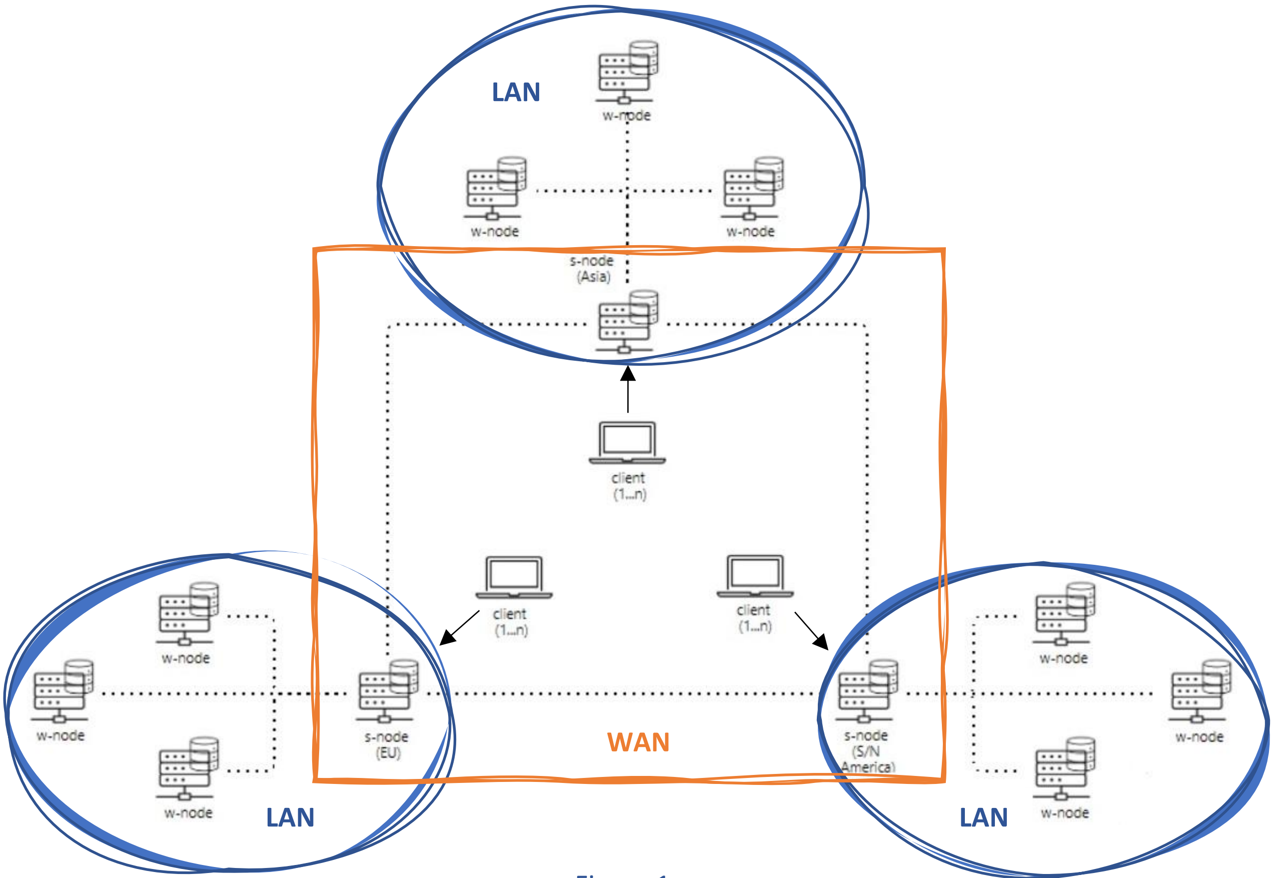

Figure 1.

Clients submit the preprocessed dataset (calling as input from now) to super peers and receive results from them; super peers act on behalf of their clients and themselves by distributing the load of processing among weak-peer nodes, forwarding response messages back to client and answering name resolution queries with the help of index server (see Naming section for more detail) coming on overlay network.

Although centralized manner between weak and super peers is efficient, super peer is a single point of failure for its cluster. To prevent the negative effect on reliability and preserve continuous availability, having secondary super-peer (SSP) which is synchronized with primary super-peer (PSP) can be adopted. Replica configuration between databases will follow master-replica approach where asynchronous propagation scheme – PSP DB sends the metadata of clients to SSP DB and continues its job instead of waiting for the acknowledgement message on the application of CRUD operation - will be fulfilled.   During the repairment of primary super peer, secondary super peer can handle the incoming inputs. This solution is called super-peer redundancy.

Another challenge driven from the nature of the network is the election process of a super-peer node. *Distributed decision-making ability* fulfils the need of a fully distributed algorithm that operates under dynamic condition – ad hoc network. It also attempts to utilize the heterogeneous capacities like bandwidth, processing power, uptime, neighborhood size of the participating peers to improve performance and reliability for the whole network. It is possible to elect the best-fit peer to be promoted as super-peer node based on mentioned capacities. The protocol implemented for election is Hierarchical 2-level Overlay (H2O) - a distributed negotiation protocol. More precisely, during the election, each node broadcasts a message to other super-peer candidates by which it makes relevant local observations and by taking these observations into account, makes a judgement on which node reconfiguration action will be taken. The message can be security certificate since only peers that hold the certificate are eligible to be elected as super peers.

Naming

Names play a crucial role in all computing environments to access resources, differentiate entities (super peers in our case), or refer to correct location(s). An entity may have multiple access points and they are not stable; therefore, it is essential to name the entity with location-independent property.

When a new client joins the WAN network, it needs to connect to the super-peer node to proceed with the input processing. As soon as client connects, it receives the metadata (may include IP address, attribute-value pair (ex. (country-field) where field identifies the functional genome analysis field that the super peer is responsible for) of the closest logical super-peer nodes sent by index server according to the GPS location of client. Then client chooses the needed node and connects by including appropriate metadata. Number of index servers may be limited to the number of super-peer nodes in the network or can be kept independent of the super-peer nodes and amount can be decided upon the network traffic analysis. In the first case, both super-peer node and index server should send an "alive" message in a mutual manner every 5 seconds to preserve the consistency of metadata and to be aware of the downtime happening on any of them to take appropriate action (refreshing to metadata of SSP by server or handling the metadata sharing to clients by node). In the second case, metadatas are requested from nodes by each index server every 5 seconds. In both cases, if the response does not come within 30 seconds, index server/super-peer node accepts the second party as stale, within 5 minutes, as died.

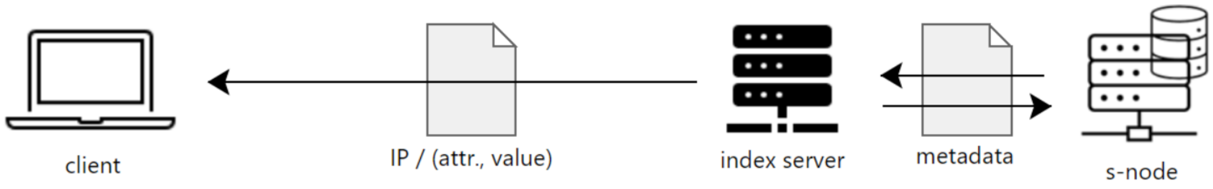

Figure 2.

To process the input, super peer itself keeps the index over its client which holds the information (continent code + ID given by network administrator) to uniquely identify the client. When existing client makes changes on identification information, it will send metadata to its super-peer and the super-peer will add/update this metadata on its index. In case of leave, the removal of the metadata will be implemented by super-peer. If the dependency is kept between index server and super-peer node, both clients' and node's metadata can be stored in index server.

Communication

As it was mentioned, the inputs coming from clients are received by super-peer nodes and they deliver the partitions of the input to weak-peer nodes to process the purpose of the request.

Request types include analysis and processing where analysis is a form of processing with a minor difference such that datasets generated from 2 or more blood samples are put together to analyze the various scenarios. For instance, what positive/negative changes on genome may be observed when a sequence of DNA letters of a person's lung (living in Asia) who suffers from cystic fibrosis – genetic disease causing persistent lung infections and limiting to breathe over time – are replaced with another person's healthy lung DNA encoding who lives in North America or experimenting genetic characteristics of 2 politicians.

The communication between client – super-peer and super-peer – weak-peer follows queue-based persistent mode. In both request types, input should meet the recommended specifications such as size, field to be eligible for processing step. After validation, super-peer node measures the required CPU and storage, compares available capacity of weak peers for processing, partitions the input, and assigns tasks accordingly. Each weak-peer node gets the task and either starts the processing or puts the task into local queue. At the end of isolated processing, the results are sent to super-peer node for aggregation and delivery to client.

In analysis request, since the processing scales geographically, super-peer node may need to connect to other super-peer nodes' databases to acquire DNA sequence(s) from other regions and start partitioning both the input and dataset(s) obtained from other super-peer node(s). To ensure consistent exchange of information among peers, causal ordering delivery of messages (CODM) carries substantial functionality in such environments. It provides message synchronization and reduces the indeterminism produced by asynchronous execution of the peers, or by random and unpredictable delays of messages within the communication channels. In the proposed protocol – Immediate Dependency Relation (IDR), the idea is based on sending only identifier of immediate predecessors (super-peer nodes) of a message. Regarding the storage requirements, this protocol stores the vector clock and the message control information, thus the resulting storage overhead is twice the number of super peers in the system. Nevertheless, since weak peers are responsible for the massive processing, overhead issue can be tolerated for the current situation of topology.

Processes

If we zoom in to super-peer node, main physical space is given to storage, whereas processing cores occupy huge portion of space in the weak-peer node.

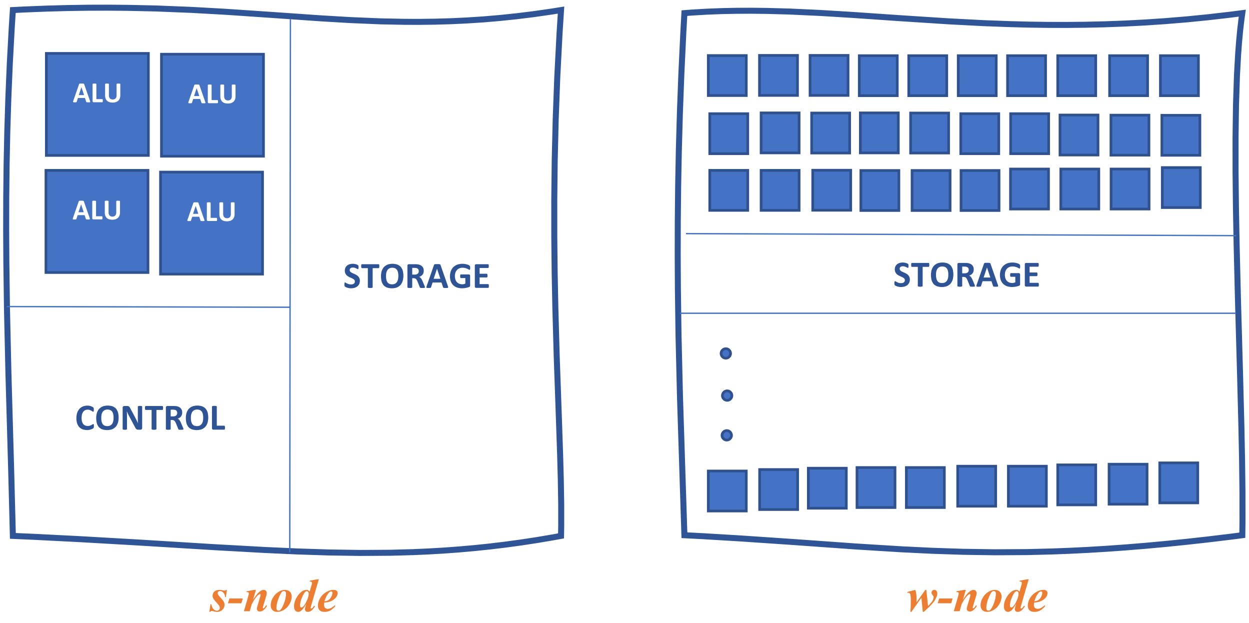

Figure 3.

When super-peer node receives the request, it first determines the type of request. After the validation and/or communication with other super-peers mentioned above, it allocates memory and space in MMU for the aggregated result. Next, job of super-peer node becomes monotonic such that it controls weak-peer nodes throughout the processing.

For weak-peer nodes, massive-core technology – GPUs are utilized which has its own device memory to easily increase the performance of the node without raising the processor clock speed. Each NVIDIA GPU has +1500s parallel processing cores where cores are managed by thread manager having 0 overhead in thread switching. GPU uses parallel computing platform and programming model, named CUDA, running kernels (sets of computing instructions designed for GPU) in 1000s of lightweight concurrent threads. A kernel is executed by a grid containing blocks where each block is a batch of threads that can cooperate via on-chip shared memory (user-managed data cache) and synchronization.

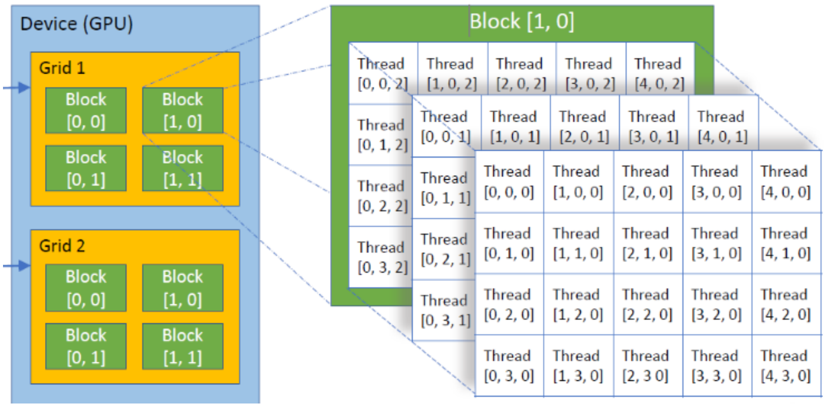

Figure 4.

References

Bote-Lorenzo, M. L., Dimitriadis, Y. A., & Gomez-Sanchez, E. (2004). *Springer-Verlag LNCS 2970*. pp. 291-298.

Evropeytsev, G., et. *al.* (2017). An efficient casual group communication protocol for P2P hierarchical overlay networks. *Journal of Parallel and Distributed Computing*. pp. 149-162.

Manuel, P. (2011). A lightweight distributed super peer election algorithm for unstructured dynamic P2P systems. *Universidade Nova de Lisboa*.

Yang, B. & Garcia-Molina, H. (n.a.). Designing a Super-Peer Network. *Stanford University*.