A Web Interface for Running Parallel Image Processing Algorithms in AWS Cloud

Toaha Siddique
Department of Mathematics and Computer Science
CUNY Queensborough Community College
New York, USA
toaha.siddique18@student.qcc.cuny.edu

Dr. Esma Yildirim
Department of Mathematics and Computer Science
CUNY Queensborough Community College
New York, USA
eyildirim@qcc.cuny.edu

Abstract— Whole scale images produced from medical scanners of digital pathology technology are very difficult to be processed and read by a computer in normal circumstances. The whole scale images after being scanned are turned into datasets. The datasets are very large and not readable by the general computer program. A system must be implemented with the help of other systems to make a fluid and easy way for the datasets to be converted into a compatible format. Extreme scale image parallel processing algorithms are needed to be run to convert the datasets into a readable format. A cloud storage and processor are needed for wide access of data and dataset processing. Input datasets are needed in the cloud storage of Amazon S3 and they must be identified. A cluster of compute nodes is needed to be instantiated for processing by using Amazon EMR and EC2. Parallel image processing application code is needed to be moved to the instantiated cluster. Cluster parameters are needed to be configured and the application is needed to be run. In this research, a web-interface is developed to automate these tasks for the user and test it with extreme-scale image processing algorithms such as content-based image retrieval and basic transformations. The web interface is an initial version of an image processing portal running on the cloud. The interface is designed by interacting with the AWS cloud through its Java API and uses technologies such as JSP, HTML and CSS.

Keywords— Whole slide image; parallel processing; algorithms; cluster; cloud computing; Java; HTML; CSS; JSP; distributed file systems; Hadoop; HDFS; Amazon EMR; Amazon S3; Amazon EC2

I. Introduction

Whole slide images are large whole-body images of a physical structure with multiple resolution (e.g. multi-giga-pixel) microscopical data sets produced by digital scanning devices. This new technology is used by many leading institutions around the world for education, research and diagnostic purposes. Whole-slide image scanning is very advantageous over other microscopy methods in aspects like long-term keeping, easiness in reusability and analysis. Whole slide scanner technology makes it possible to scan a single glass slide in approximately 3 minutes. This results in large amounts of data in a short time. The amount of big data created by the scanners has become a great source for large-scale data analytics applications. The software developed for whole slide imaging focuses on visualization and it has fewer applications which concentrate on quantitative analysis where it is useful to provide reproducible and objective measures to support diagnostic decisions. Analytics applications like classification of cancer types and automated content-based image retrieval need expensive computational operations like filtering and sub-sampling, feature computation, low-level transformations and object segmentation. Several studies have investigated the use of deep learning strategies on WSIs in an effort to resolve complex problems in diagnostic pathology. In studies, algorithms and workflow depended on pre-processing the WSI data into tiles of smaller size because of the high cost of the computation in neural networks. Whole Slide Image applications have very limited use and the access to the storage system is not a part of the parallelization process. Because of this fact, there is no performance analysis on the data access of whole slide images. Access to the storage system by the server is provided. There are several image processing algorithms which include pixel data access tracking, classification, feature extraction, and object segmenting which are analyzed by the CPU and GPU and MIC processor architectures. When whole slide images are sub-divided into smaller size images during pre-processing steps and moved to a storage system of the host node within the cluster, the working nodes access the disk system of the host node of the cluster to process the image. The transfers between multiple disk systems impede overall performance.

The need of a web interface is shown in this paper for the ease of the overall process of converting whole slide image datasets. By using Amazon EC2 processor which is used by EMR for running applications the storage of the datasets is also accessed through Amazon S3. These technologies ae part of the Amazon web services system which makes the process easier. The master application assigns a number to the worker application which is then able to access the storage directly. But in this application parallel processing works only for one whole slide image dataset at a time. The data size inserted into the memory of the worker application changes according to the memory size of the node. The scalability of the application is affected and cannot be scaled to the proper size in regards, to the data analytics processes and algorithms. Fixing the size of the tile which was brought into the memory of the node based on the size of the WSI and memory limit of the node creates algorithm processing difficulties. A fixed number of processors can only scale the memory. In issues related to large-scale of spatial data (e.g. satellite imaging and whole slide images) and high computational complexity of spatial queries are addressed using a solution based on the Hadoop ecosystem. In this approach, the WSI's are converted into a Hadoop file system compatible format as a pre-processing step, however the paper provides only limited performance results. The issue of readability and compatibility arises when converting a whole slide image data using a file system. But the Hadoop file system faces this issue by having a pre-processing step by converting the file into a compatible format. The compressed whole slide images in the Hadoop file system are converted into binary format from multiple inputted datasets. In overall the challenges are the large size of the whole size image datasets and their scalability. The file formats of the scanned data sets of the whole slide images. The libraries for the programming language of the parallel processing algorithm application being foreign to the computer system. The challenges presented by the whole slide images are solved by a web interface to automate the tasks after parameters are inputted into the Amazon EMR cluster settings. The parameters and settings create a system step by step so that the whole slide datasets can go through the proper processing of the parallel processing algorithms through Amazon EC2 and the Hadoop distributed storage system on Amazon S3.

II. Cloud Computing

A. How it works

Cloud is a collective term for various services on the internet. The cloud is basically remote services which can be accessed through the internet and those services are not readily needed to be available by one's own computer. The services are software as a service, platform as a service and infrastructure as a service. Software as a service are readily available software on the internet to do different tasks like Gmail. Platform as a service are readily available software and hardware tools for use like Heroku. Infrastructure as a service are readily available infrastructure services like networking, storage and virtualization as seen in Amazon Web Services and Microsoft Azure. The equipment we normally have with us, that is the hardware and also the software installed on the computer are sometimes difficult to manage and also there are portability issues, they can be expensive for work heavy tasks too. When hardware and software are readily available on the internet for use and are managed by the owners of the respective services and are frequently updated and they are comparatively cheap as physical material it makes tasks more affordable and easier to complete. So, the cloud is a lot like a portal on the internet where the tasks are done, or the services are being streamed to the user. The files also involved are transferred to the user. Every task or process done, and the services used are all happening on servers or other computers which are located in some other place besides the users.

B. How it is used for Parallel Applications

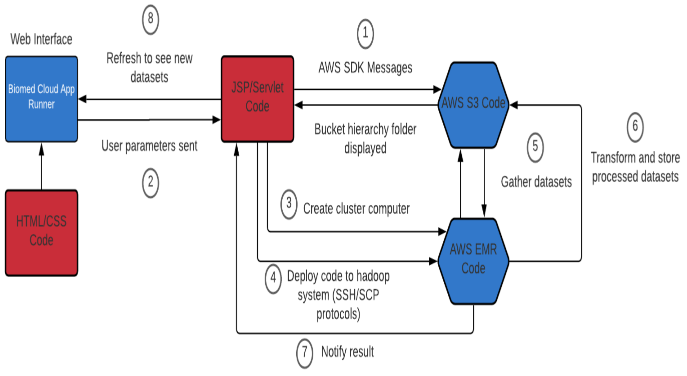
The cloud uses a huge and safe role when it comes to distributed computing and parallel processing. Normally when computers are used as a single system, if an issue occurs like the system going down it causes a huge problem the data gone cannot be retrieved. Distributed systems were introduced to tackle issues faced by a huge server. A distributed system is a group or cluster of computers with a master computer and working computers. The working computers follows the commands of the master computer. In a distributed system when there is a file system managing the clusters, the files are duplicated in the different nodes (working computers). If a file is lost in a node, a copy of it is available in another node. The cluster of computers in the system work in a parallel manner. Chunks of a file are divided amongst the cluster and they are processing the file at the same time. Less time is taken than a single huge server or computer to process the files. Distributed systems can be difficult to manage without an administrator. The cloud plays a great role in the service of distributed systems. The full distributed system services are available on the cloud. Amazon Web Services, the top cloud platform, gives different cloud services. Amazon EMR works as the parallel processing application runner and Hadoop Distributed file system is the distributed file system. Different services of Hadoop are run through Amazon EMR, while Amazon S3 is the storage and Amazon EC2 gives the compute capacity.

C. Research Purpose and Accomplishment

This research was done to make a system which would make the task of making whole slide image datasets scalable and readable. A web interface serves as that system which would give ease to the user to gain the results needed. So, a basic web interface which could interact with AWS was developed with a programming language which had libraries which could read the datasets and interact with the cloud which runs the parallel processing algorithms. As the datasets were inputted and the algorithms were run through the web interface after a cluster were created through selection and parameters, new datasets were created in the output path which are readable and scalable.

III. System Architecture

A. Model-View-Controller Model



The web interface system is designed using Java code, HTML, CSS and JSP code. The system interacts with

Amazon web services. Through Java API the AWS code is made to interact with the web interface.

B. Tools

- JSP/Servlet:
  Java Server Pages lets the user use Java programming language in accordance with HTML and CSS on the same page. JSP takes user input.
- HTML/CSS:
  HTML is used to display text on the webpage in various ways. CSS is used to style the text and webpage.
- AWS SDK:
  AWS Software Development Kit is used to give all the available libraries to the integrated development environment to code for different processes and put parameters for the system.

C. *Interaction of the AWS Software development kit with JSP Servlet code*

Through Java code a program is created to access AWS S3, EC2 and EMR through the AWS SDK. Through JSP using Java code, a hierarchy of the buckets and objects of Amazon S3 is created in the webpages. Algorithm selection is given on the web page. Input and output path readers for the datasets is set in the webpage. The option for selecting the node type and node number is given. The tile width and height selection are also given in the web interface.

D. *Users parameters being sent from the Web Interface to the JSP Servlet code*

Through the form of the webpage the inputted options of algorithm, input destination, output destination, node type and number, are sent to the Java code for input. Through Java code on JSP and the AWS SDK the inputs are able to connect with AWS EMR.

E. *JSP Servlet creates a cluster using AWS EMR code*

The JSP Servlet after sending the inputs to AWS EMR, the cluster parameters already being set by Java code in JSP, an AWS EMR cluster is created according to the node type, node number.

F. *JSP Servlet deploys code to the Hadoop System*

According to the selected Algorithm, the JSP servlet moves Java algorithm code for running parallel image processing algorithms in the Hadoop distributed file system in AWS S3.

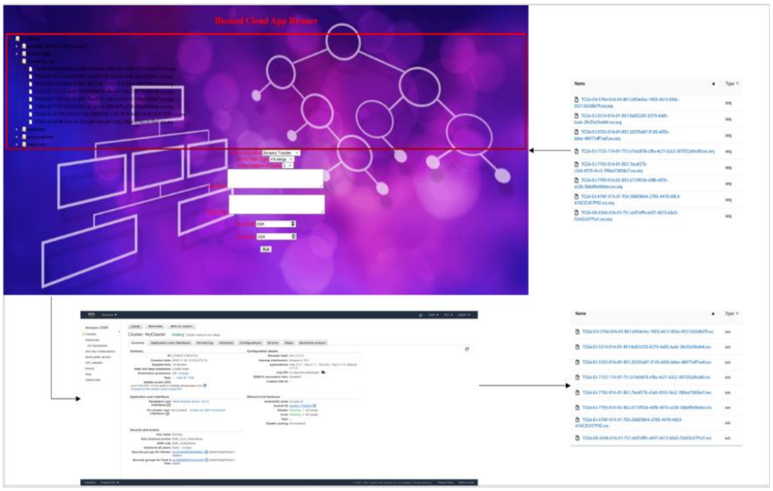
G. *AWS EMR code gathers datasets in AWS S3*

Though AWS SDK, AWS EMR gathers the inputted whole slide image datasets in AWS S3.

H. *AWS EMR transforms and stores the processed datasets*

The parallel image processing algorithms are run on the inputted datasets. The inputted datasets are transformed into datasets which are scalable and readable. The new datasets are stored in the selected output path. The system notifies the results. After the web interface is refreshed the new datasets are seen.

IV. System Design



The system was designed using Java, HTML and CSS on JSP. The web interface displays AWS S3 buckets and folders on the interface. The web interface is designed for selection of algorithm, node number, node type, dataset input destination, dataset output destination, tile width and tile height.

V. System Features

- The system has the ability to contact AWS S3 storage system and display a hierarchical representation of buckets and folders.
- The system has the ability to set transform algorithm parameters.
- The system has the ability to instantiate and configure a parallel AWS EMR cluster computer based on user parameters.
- The system has the ability to deploy algorithm codes on the EMR cluster using SSH and SCP protocols.
- The system has the ability to run the algorithms remotely on the EMR cluster.

VI. performance Results

The EC2 instance selected was t2.micro.The time taken for the cluster to start is 10 minutes. Three node types m4.large, m4.xlarge and m5.xlarge were used with different number of nodes. The input datasets were of 1.6 GB in size. The time taken is the time excluding the time the cluster takes to start. It was seen that the better the node type and more the number of nodes the time taken was less.

Node Type	Node Number	Time Taken
m4.large	3	30 minutes
m4.large	4	26 minutes
m4.xlarge	3	25 minutes

m4.xlarge	4	20 minutes
m5.xlarge	3	20 minutes
m5.xlarge	4	15 minutes

Conclusion

Whole Slide Images present challenges for data analytics algorithms to be performed in terms of scalability and compatibility. Using a distributed system on the cloud makes the task easier, faster, less risk prone and more efficient. Many steps are involved to do the task. A web interface acts as the medium for doing the steps very easily. Different options for selection are there to be chosen easily. The tasks are then automated for the user to see the results of new transformed datasets which are scalable, compatible and readable.

Future Work

This research opens the path for researchers and medical professionals alike to successfully scan whole slide images and perform data analytics algorithms on them using a web interface and AWS as their platform.

Acknowledgment

This research is funded by PSC-CUNY Award # 62177-00 50.