

Big Seismic Data Visualization on Cloud: A Case Study Collaborating with Industry

Hao Zheng¹, Lei Huang¹, Bidur Bohara², Mike Heck²

***¹Computer Science Department,
Prairie View A&M University,
Prairie View, TX 77446***

***Analysis Division,
Thermo Fisher Scientific,
Houston, TX 77084***

²Materials and Structural

Abstract

Scientific and industrial data have grown too big to efficiently process and visualize using the current desktop solution. The cloud computing becomes a feasible solution to tackle the big data challenge. However, how to efficiently visualize big data sets such as the Oil & Gas industry seismic volume on the cloud remotely remains a challenge. It is not feasible to send big data back to the client side to visualize them. The limitations of Internet latency and bandwidth require all data rendering and processing are carried on the cloud side in order to achieve real-time big data visualization. The paper presents a collaborative work with Thermo Fisher Scientific, our industry partner, to design and implement a cloud-based visualization software system toward to providing a feasible solution for the big data visualization. Our system comprises of four parts including a scalable seismic data processing package, a data communication library, a data visualization module, and a web portal. The work builds on top of open source packages including Apache Spark, Hadoop, and a commercial visualization package named Open Inventor™ 3D Visualization toolkit from the Thermo Fisher Scientific for rendering images. The paper outlines the bricking data format and the level-of-details algorithm used to make it feasible to visualize big data sets near real-time. The work demonstrates the superior performance to the prior work using the same data sets. We implemented this work on the PVAMU big data cloud platform using large seismic volume data to evaluate the performance.

1. Introduction

In our previous efforts [1,2], we built a big data analytics cloud platform with special interests in geophysics data sets. The cloud platform is named Seismic Analytics Cloud (SAC) to process and analyze seismic data with the deep learning capacity [3]. Although the platform is able to process and analyze big seismic data sets with scalable performance, the big data visualization on the cloud remains a challenge to us. In this project, we aim to improve the visualization efficiency of the SAC platform while users conduct the seismic data analytics. At the beginning, SAC was mainly developed as a new computing platform with a balance of both performance and productivity, and featured with big data analytics capability. Soon afterward, we collaborated with Thermo Fisher Scientific to integrate the rendering of seismic slices in SAC

platform. The platform can now support the management of seismic data volumes, attributes processing, seismic analytics model development, workflow execution, and 3D volume visualization on a scalable, distributed computing platform [2]. However, for a big 3D seismic volume takes a long time (over a minute) to display a rendered 3D image on the platform via a web browser. To address this challenge, we collaborated with Thermo Fisher Scientific to bring the high quality and high-performance rendering of seismic volume in SAC using Open Inventor [4].

As a result, the collaboration between Prairie View A&M University (PVAMU) and Thermo Fisher Scientific delivered a successful research platform that combines the power of scalable big data analytics and a close-to real-time big 3D data visualization capability. All of these functionalities are delivered to end users via a cloud platform and accessible via a web-based application. In this paper, we introduce the background of the work, the system overall architecture, the visualization implementation, the discussions and conclusion in the following sections.

2. Background

The objective of SAC is to deliver a scalable and domain-specific cloud platform to facilitate the seismic data analytics research and development in the geophysical areas such as the oil/gas exploration or the earthquake detection. Figure 1 shows the overall system structure used in SAC. The bottom layer is operating systems SAC can build on; next layer is to provide the JAVA and Python runtimes with Hadoop Distributed File System (HDFS for storing the large seismic data files and No-SQL database Cassandra for metadata and data attributes. Standalone, Mesos, and YARN are all supported on our platform for the resource management. We use Apache Spark as the big data parallel processing engine together with the widely-used signal and image processing libraries to provide scalable performance and good productivity. At the very top layer is SAC SDK, workflow, templates, and visualization module. Researchers can build their own seismic data analytics work on top of the cloud platform.

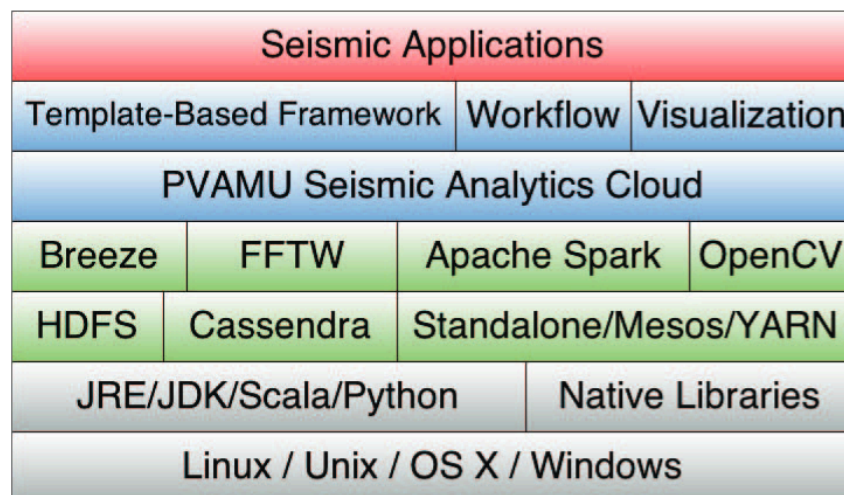


Figure 1. The Software Stack of Seismic Analytics Cloud Platform

To efficiently process the large seismic data, we need to build SAC on a scalable computing platform. We chose Apache Hadoop, a collection of open-source software utilities, and Apache Spark, an open-source distributed general-purpose cluster-computing framework, for processing seismic data. These two frameworks are widely used for big data analytics as they partition the data into small chunks and distribute these chunks across worker nodes to achieve scalable performance.

2.1. Apache Spark and HDFS

Apache Spark is an open-source distributed cluster computing framework developed by AMPLab at the University of California, Berkeley. Compared to the MapReduce technique in Hadoop, Spark provides a resilient distributed dataset (RDD) that keeps data processing in memory to reduce the data IO. To use Spark efficiently, it requires a cluster resource manager and a distributed file system. Spark supports standalone (native Spark cluster), Hadoop YARN, or Apache Mesos. For the distributed storage system, it provides an interface for Hadoop Distributed File System (HDFS), Cassandra, OpenStack Swift, Amazon S3, or a custom solution [5]. We use HDFS [6] for the distributed fault-tolerant file system in SAC.

2.2. Communication

In this project, we use ZeroMQ[7] to transfer messages among the data service and rendering service. ZeroMQ allocates sockets to users to load their message and transfer the message across different types of transportation like in-process, inter-process, TCP, and multicast. The advantage of ZeroMQ is that it can swiftly transfer messages in a cluster. Its asynchronous I/O model allow users building a scalable multicore application.

We use ProtoBuf [8] to define the communication protocol between these services. ProtoBuf is a flexible, efficient, automated mechanism for serializing structured data. Users can define the communication protocol and data structures themselves. Then, they can use the ProtoBuf generated source code to read and write that specific structured data from a variety of data streams. When there is something new to update, programmers can directly update the data protocol and data structures without changing the programs. Also, it can be used in many language codes, includes Java, C++, and Python.

2.3. OpenInventor

Open Inventor is a 3d-visualization toolkit offering a high-level object-oriented graphics libraries (API) for creating advanced 3d-visualization applications [9]. It provides a set of dedicated extensions for developing visualization solution for various data types, such as geometries, volume, mesh, and images, and for implementing remote/cloud-based rendering capabilities.

In addition to hardware-accelerated volume rendering, VolumeViz extension of Open Inventor

provides a large data management (LDM) technology to manage out-of-core loading of large volumetric data that do not fit in the available system memory and far exceeds the video memory capacity even on high-end graphics cards. Since Open Inventor uses GPU for volume rendering, it can only render seismic data that fits in the limited video memory. To address this limitation, LDM creates a hierarchical, multi-resolution bricked representation of seismic volume to allow out-of-core loading of data as needed for rendering.

Open Inventor avoids loading whole seismic volume in full-resolution, instead only loads bricks from different resolution levels depending upon the camera position and available system and video memory. Another advantage of creating hierarchical representation of seismic volume is that it allows Open Inventor for quick rendering of volume in low resolution, and progressively render to higher resolution, without impacting interactive quality of the visualization tool.

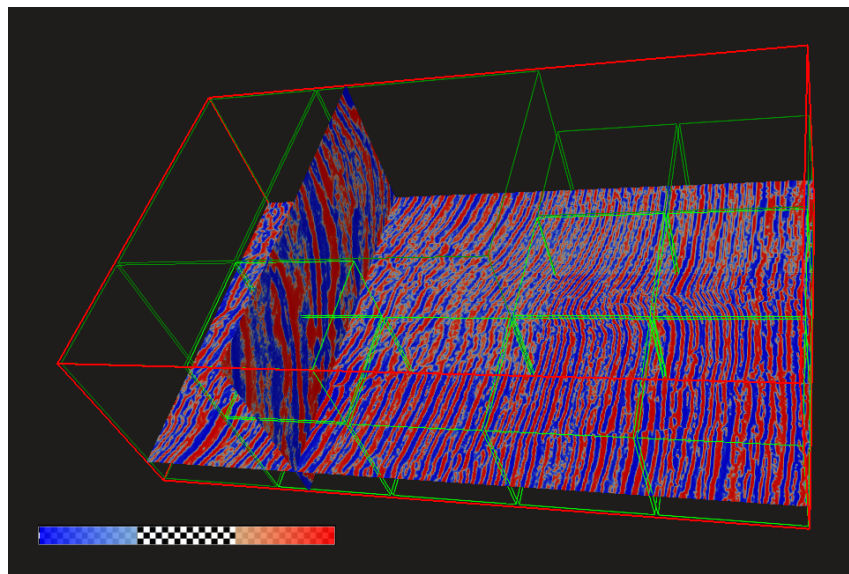


Figure 2. Slice rendering with a set of bricks from different resolution levels. The bright green boxes represent full-resolution bricks that are closer to the camera; and dark green boxes represent low resolutions bricks for volume region away from the camera for available memory.

The bricked representation of seismic volume also makes it possible to store bricks (chunks of volume) in a distributed file system in the cluster. Open Inventor will request data server for individual bricks, as needed, for the volume rendering.

Open Inventor also provides remote visualization technology, RemoteViz, that makes it possible to perform remote rendering of large data in a dedicated high capacity GPU-enabled server **node**. The rendered images are transmitted over web to a web application in a client's devices, which could be a tablet, a smart phone, a laptop or a desktop. PVAMU's SAC platform uses RemoteViz technology to render seismic volume on a remote rendering server and display the rendered image on the web-page to be accessible from anywhere. RemoteViz also manages 3D interaction from users, performs bandwidth calibration to adjust quality vs interactivity, and supports VP9 and H.264 encoders for streaming images.

3. Implementation

Although significant improvements have been made in the Internet speed and quality, the cloud-based big data visualization remains a challenge to researchers since it is not feasible to transfer a few GBs from the cloud to any users at real-time. The data transferring latency requires us to implement a big data rendering service on the cloud so that only the rendered images are transferred between the users and the cloud, which make the real-time visualization feasible.

SAC implements a cloud-based big data analytics and visualization platform by providing seismic data accessing and visualization services. Figure 4 shows the user interaction with SAC through a web portal, the visualization module detects the user's interaction with data, and then communicates with the data service and the rendering service. Data is sent from the data service to the render service to render them into 3D images, which are pushed to the web portal to display.

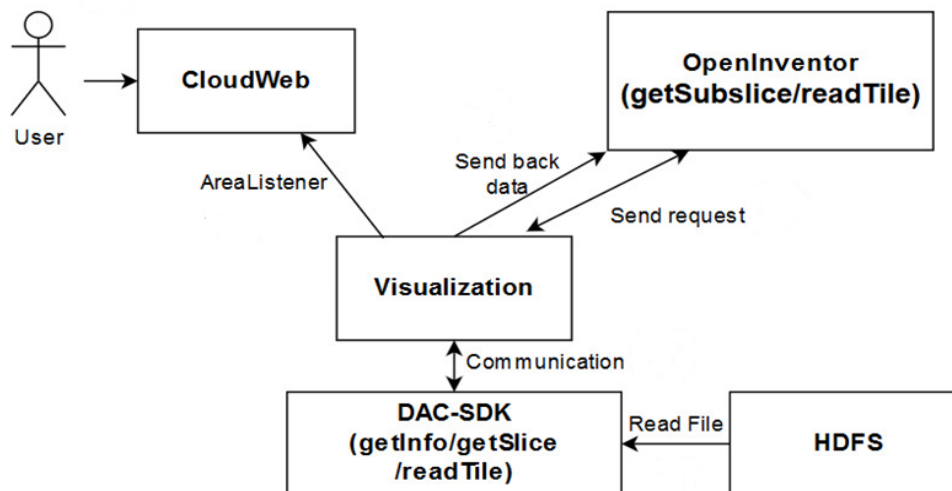


Figure 3. SAC data access and visualization services

The first implementation of the cloud-based visualization takes over a minute to load a large seismic volume. The reason for the long-time visualization is that the 3D seismic volume is stored in memory aligned with one direction, either inline, crossline, or Z (or X, Y, Z). When the current display shows a seismic volume in 3D, whole volume is needed to render the image. To display a slice of data, SAC may need to transpose the data if the slice is not aligned with the internal data structure in memory, which takes significant time. To overcome these shortcomings, we implemented the brick format and the Level of Detail (LOD) to allow partial data transferring and data cache. Using LOD, SAC can cache different resolution of data on different levels. When users try to view the seismic data, the platform can show the low-resolution image to them at first. And then the higher-resolution data is gradually loaded into the memory in the background. The brick format removes the transposing task since a brick can be used for any directions of a slice.

With the collaboration with Thermal Fisher Scientific, the student works in the project is able to understand the internal data structures and APIs of the Open Inventor toolkit. Data in Open Inventor is stored as an Octree structure in one file. In order to reduce the data loading time, we pre-computes the LOD of any seismic files in SAC and convert the previous slice-based data storage into the brick format. We define the size of bricks and the index for each brick referring to the rules given by the Thermo Fisher Scientific.

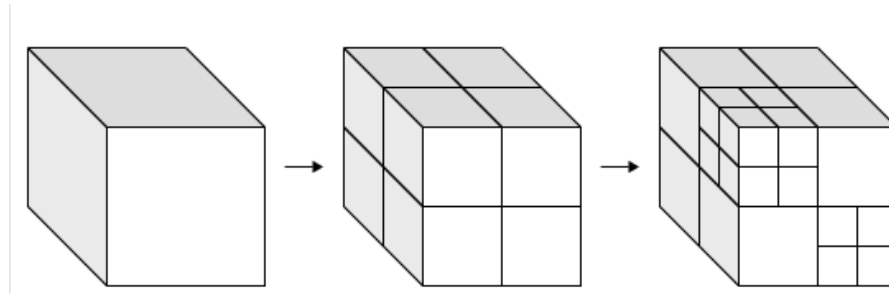


Figure 4. Octree structure used for the level of details

In a standard seismic data file, the data is stored slice by slice. In order to transform these slices to bricks, we aggregate several slices at first according to the defined brick size. In this way, we get multiple large cuboids, which size in the inline direction is the same as the defined brick size. After that, we cut each cuboid along the other two directions to get the final bricks. At last, we define a 3D index as the start position for each brick. After creating the brick format, we create several levels of bricks in different resolutions. We subsample one level of bricks to generate a lower resolution brick. Figure 4 shows the Octree structure and level of details.

After we transform data stored in slices to bricks, we need to give a unique index to each brick following the rule given by the Thermo Fisher Scientific so that the Open Inventor can query each brick using its unique index. And all the bricks in each brick volume need to follow the order in Figure 5 to define their indices.

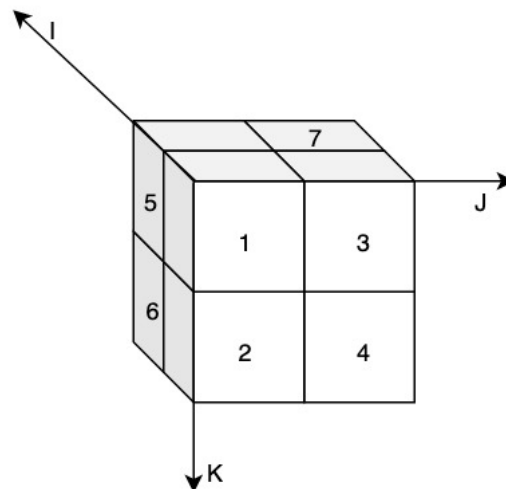


Figure 5. index order in Octree structure

To further improve the visualization efficiency, we cache the LOD data in each file into memory at the beginning when a user accesses it. When the users try to view the seismic data next time, the system can directly query required bricks from the memory without invoking any calculation. Because the resource on each device is limited, we set up a limited number for cached data files. When there are more than five files cached in this array, the system is going to drop the first cached data and cache the new data instead.

After the brick format and LOD implementation, the data service and rendering service communication are improved. As we described before, SAC uses ZeroMQ and ProtoBuf to implement the data communication. When users interact with data view through the web portal, only partial bricks at a certain level of details are needed for the rendering service to complete its job. The rendered images are transferred to the user's web browser to achieve real-time data visualization and interaction.

Table 1 below shows the performance after we change the Open Inventor configuration from reading slices to reading bricks. In this test, we use a 198.7MB seismic volume to compare the performance. From this table, we can find using LOD and bricks helps a lot to display image faster. Figure 6 shows how images looked in different resolutions.

	Full resolution	1/2 Resolution	1/4 Resolution	1/8 Resolution
Read in Slices	19.11 second	12.48 second	6.52 second	5.03 second
Read in Brick	10.18 second	2.08 second	1.15 second	0.65 second

Table 1. The image loading time for the same seismic data file

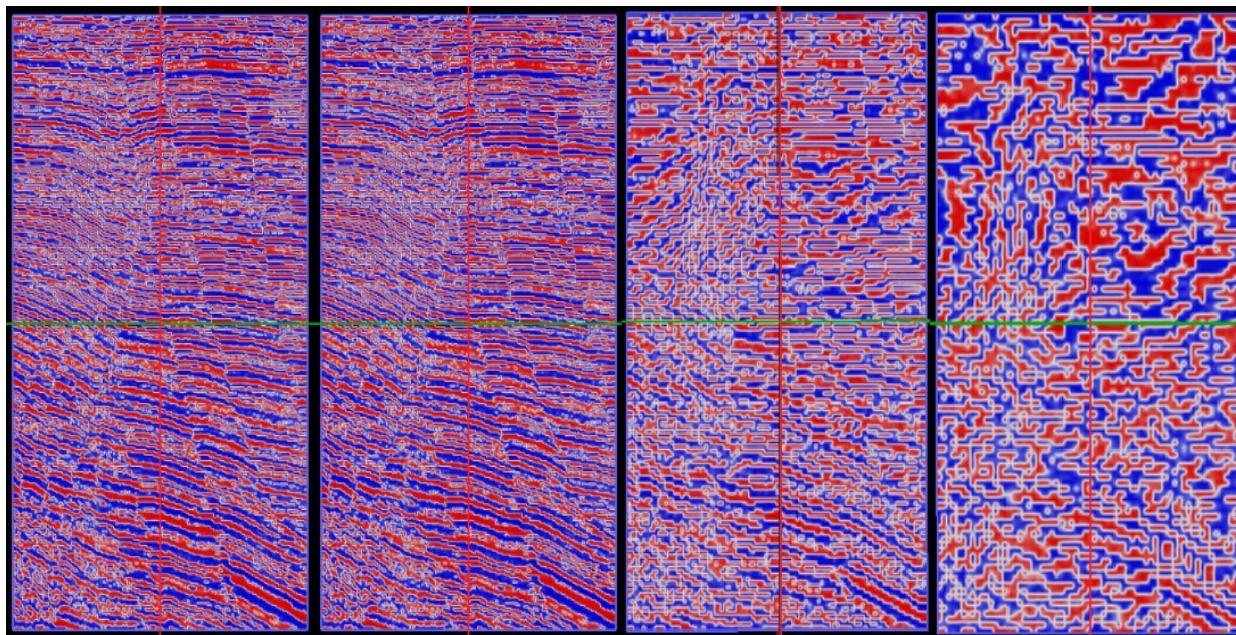


Figure 6. image in different resolution (From left to right is full, 1/2, 1/4, 1/8 resolution)

4. Discussion

4.1. Open Source vs. Proprietary Software

SAC was built mostly on top of open source software packages. For visualization, we have a choice of either selecting an open source visualization package or using the Open Inventor toolkit, which is proprietary software from Thermo Fisher Scientific. There are obvious pros and cons for open source and proprietary software. For our case, we collaborate with the Thermo Fisher Scientific since it is a good opportunity to our students to learn the industry standards. The Open Inventor toolkit has advantages of robustness and high-performance. By consulting with the experts in Thermo Fisher, we are able to understand Open Inventor toolkit and build the visualization module quickly. The academic team spent more time in the data analytics and scalability research, which has been the focus of the research team.

4.2. Next steps

Although this project allows user to visually analyze their seismic data in a short time, there are multiple functionalities that can be improved.

- While loading low resolution data, the rendered image fails to display all features of the seismic data. We need to explore better subsampling algorithm to build the LOD.
- We will provide an option to blend/composite together multiple attribute volumes. For that we need to implement simultaneous loading of multiple seismic volumes.

5. Conclusion

From this project, we demonstrate that it is feasible to establish a real-time big data visualization system on the cloud. By using LOD and bricked format, we can quickly display a rendering of big 3D seismic volume. Comparing with the previous method that read data in slices, we observed a significant improvement in the rendering performance.

For this work we collaborated with Open Inventor product team from Thermo Fisher Scientific, who provided us with valuable insights on the concepts of efficiently loading and rendering large 3D volumetric data on the cloud using Open Inventor toolkit. With their help, we could successfully complete this project and achieve the objective. It is also very valuable experience to our students in understanding the data visualization technology as well as the coding standard used in the industry.

6. Acknowledgement

This research work is supported by the US National Science Foundation (NSF) awards #IIP-1543214 and by the Office of the Assistant Secretary of Defense for Research and Engineering (OASD(R&E)) under agreement number FA8750-15-2-0119. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies

or endorsements, either expressed or implied, of the US NSF, or the Office of the Assistant Secretary of Defense for Research and Engineering (OASD(R&E)) or the U.S. Government.

References

1. Y.Yan, L. Huang, L. Yi, "Is Apache Spark scalable to seismic data analytics and computations," 2015 IEEE International Conference on Big Data, 2015
2. Y.Yan, M. Hanifi, L. Yi, L. Huang, "Building a productive domain-specific cloud for big data processing and analytics service," *Journal of computer and communications*, vol. 3, no. 5, pp. 107-117, 2015
3. Lei Huang, Xishuang Dong, and T. Edward Clee, "A scalable deep learning platform for identifying geologic features from seismic attributes." *The Leading Edge*, published in Vol. 36 no. 3 pp. 249-256, April 2017.
4. Open Inventor Toolkit, Retrieved from www.openinventor.com, 2019
5. Apache Spark Introduction, Retrieved from https://en.wikipedia.org/wiki/Apache_Spark, 2019.
6. Hadoop Introduction, Retrieved from <https://hadoop.apache.org/>, 2019.
7. ZeroMQ Introduction, Retrieved from <http://zeromq.org/intro:read-the-manual>, 2019
8. ProtoBuf Introduction, Retrieved from <https://developers.google.com/protocol-buffers/docs/reference/overview>, 2019
9. J. Wernecke et al., *The Inventor mentor: Programming Object-oriented 3D Graphics with Open Inventor*. Addison-Wesley Reading, MA, 1994
10. J peng, C-C. J. Kuo, "Geometry-guided progressive lossless 3D mesh coding with octree decomposition." *Journal ACM Transactions on Graphics (0730-0301)*, 24(3): pp 609-613, 2005
11. Korovin Aleksandr Sergeevich, Abdrashitova Maria Ovseevna, Skirnevskij Igor Petrovich, "Web-application for real-time big data visualization of complex physical experiments," 2015 International Siberian Conference on Control and Communications, 2015.
12. H Yan Li, Junming Ma, Bo An, BDViewer – A web-based big data processing and visualization tool. 2018 42nd IEEE International Conference on Computer Software & Applications, 2018.
13. Fuchao Qi, Guizhou Zheng, Application of Interactive data language in three-dimensional seismic data visualization. 2010 2nd conference on environmental science and information application technology, 2010.
14. David Deibe, Margarita Amor, Ramon Doallo, Big data storage technologies: a case study for web-based LiDAR visualization. 2018 IEEE International Conference on Big Data (Big Data), 2018.
15. I. A. T. Hashem, I. Yaqoob, N.B. Anuar, S.Mokhtar, A. Gani, S. U. Khan, The rise of big data on cloud computing: Review and open research issues. *Information system*, vol. 47, pp. 98-115.

Biographical information

HAO ZHENG is a graduate student in Computer Science Department at the Prairie View A&M University. His interest is focus on the parallel computing.

Dr. LEI HUANG is an Associate Professor in the Department of Computer Science, Prairie View A&M University (PVAMU), a member of Texas A&M University System, where he is leading research at the Computing Research Lab since 2011. His research areas include the Big Data Analytics, Cloud Computing, and High Performance Computing (HPC). He also has several years' industry experiences in the oil and gas software development and seismic exploration research.

Dr. BIDUR BOHARA is Product Applications Specialist at Thermo Fisher Scientific, focusing on 3D Visualization solutions for E&P, Digital Rock Analysis and Medical Imaging applications. He received his PhD in Computer Science with focus on Scientific Visualization from Louisiana State University, Baton Rouge in 2015.

MICHAEL HECK is Product Manager for the Open Inventor™ 3D visualization toolkit at Thermo Fisher

Scientific. Mike has been involved in writing, managing, teaching and applying 3D visualization toolkits for scientific and engineering applications for over 30 years, surviving many different graphics platforms, languages and APIs. Mike is an Electrical Engineering graduate of the University of Pittsburgh. He has been a speaker or instructor at various conferences including GTC, SEG and SIGGRAPH.