Visual Analytics Ecology for Complex System Testing

Simon Su*, Michael Barton[β], Michael An[β], Vincent Perry[β], Chen Li[μ], Jianfeng Jia[μ], Brian Panneton*

US Army Research Laboratory* Secure Mission Solutions[β] University of California, Irvine[μ]

ABSTRACT

Diverse visual analytics requirements of our heterogeneous data community have pushed the limit of a single visualization capability. We developed a composable system of loosely coupled visualization hardware with various software to address some of our visual analytics challenges. Visual Analytics Ecology (VAE) allows users to perform visual analytics of their data using the most suitable visualization tools to accomplish their data analysis goal. We described a scenario using different visualization tools in our VAE to analyze multiple aspects of simulation data.

Keywords: Data Visualization, Immersive Display, Tiled Display.

Index Terms: Human-centered computing ~ Visualization systems and tools

1 INTRODUCTION

Test & Evaluation (T&E) is the single largest producer of data in the Department of Defense Research, Development, Test & Evaluation community and the big data challenges are diverse. The Army T&E community tests everything the Soldier touches and everything that touches the Soldier, for example, every network, application, vehicle, weapon, piece of equipment, communication device, data link, etc. and measures everything conceivable to assess its effectiveness, suitability, survivability and safety. These requirements produce massive, heterogeneous, distributed data sets requiring new approaches for analysis and exploitation. A larger challenge still is the growing number of requirements for real-time or time-critical analysis and how to use high performance computing (HPC) resources for them. The Army Test and Evaluation Command (ATEC) has made great progress in utilizing HPC for test and evaluation applications, but now requires concerted HPC focus, especially for machine learning and real-time analysis, to integrate the disparate sources of test data, visualize the results, and automatically identify anomalies and validate the data. These are common requirements throughout the larger Department of Defense T&E community [1].

2 BACKGROUND

We are creating a capability for interacting with heterogeneous big data through a composable and scalable human-computer interface. Endert et. al. discussed the importance of supporting the human in the visual analytics loop [2]. On the hardware side, one requirement is to flexibly perform analytics on devices of large size, such as a multiple monitor display wall, Figure 1. Large screen real estate offers the means to simultaneously visualize and analyze more data, and increases the potential for interactivity in a collaborative visualization environment. However, it also increases the computational demands and cost. Together with the general trend of ever increasing size of datasets means that we require a framework that can move data to and from remote HPCs, as well as tools for server and special clients, like the display wall. Another requirement is to visualize 3D spatial data which are more suited for virtual reality (zSpace and Oculus Rift) and augmented reality (Microsoft HoloLens) display examples are shown in Figure 1. Regardless of the type of display, from handheld to the tiled display wall, the goal is not images or animations; the goal is communication and interaction, among researchers and between researchers and the data.

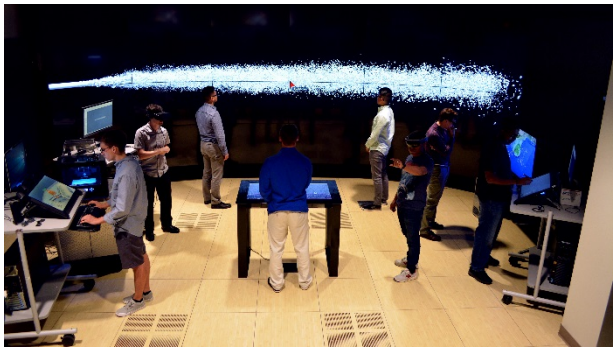

Figure 1: Visualization Ecology with various display systems.

There are countless ways to create visual analytics tools when one considers different software platforms, APIs, and hardware combinations. The appropriate combination of tools depends on the specific use case. In terms of software libraries and frameworks, the open source community constitutes a rich ecosystem of tools, many of which are the gold standard. Balancing the need for creating specific applications vs. making an extensible and general purpose tool produces challenging design decisions. We adopt the analogy of a biological ecosystem – a community of living organisms in conjunction with nonliving components that interact as a system – as inspiration for what we term a visual analytics ecology (VAE) as shown in Figure 2. The primary goal is to provide a user centric, data centric, and visualization algorithm and hardware agnostic visual analytics capability.

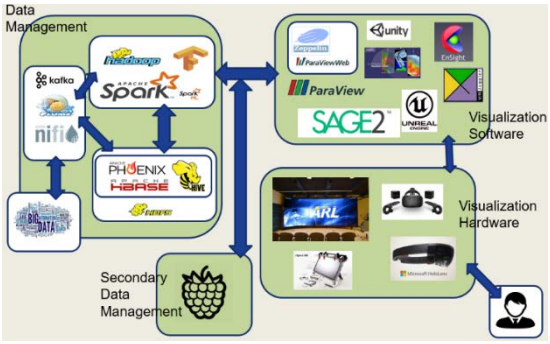

Figure 2: Visual Analytics Ecology.

*simon.m.su.civ@mail.mil

2017 IEEE Workshop on Visualization in Practice (VIP)
October 1–6, Phoenix, Arizona, USA
978-1-5386-1403-7/17/$31.00 ©2017 IEEE

3 VISUAL ANALYTICS ECOLOGY

One of the issues to be resolved is the need to uncover complex relationships among the large numbers of high-dimensional variables, for example, instrumentation data from the system under test, video data, meteorological data from the test environment, observer log data, test article maintenance data, etc. Traditional data exploration using column or structured data formats can be limiting in power, i.e., with spreadsheets and SQL databases, or complex and accessible only to trained data scientists and statisticians, i.e., data frames in R / Python. Our goal is the development of visual analytics tools that meaningfully augment the discovery of relationships in data for a larger pool of users across many domains. We work directly with users, to understand both their technical requirements, and also how the capability will be used, by whom, and under what conditions, employing a systems engineering requirements analysis method. This is showcased in our relationship with the US Army Aberdeen Test Center (ATC).

3.1 Data Management

With the monotonic growth of data, in fact exponential growth, and the requirement to utilize legacy test data to resolve issues in deployed systems, the ATC has taken steps to improve the test data flow, depicted in Figure 3. There is a total of four 'tent poles' that need to be addressed to increase the performance of the data flow: storage, transfer, reduction and analysis. Previous efforts have identified and addressed these 'tent poles' in an isolated manner. For instance, in the early 2000s, ATC consolidated multiple data stores into a single HPC SQL database. Later, when the time to reduce terabyte-size datasets was the long pole in the data flow, ATC and ARL refactored and rewrote the data reduction software and implemented it on a dedicated HPC, decreasing data reduction time from 60 hours per terabyte to 5 hours per terabyte, Renard et al. [3]. Though the time was reduced by an order of magnitude this setup incurred the additional cost of having to transfer data to the HPC and then into the HPC SQL database. As data sizes increase the data transfer, analytics and visualization become the next long poles in the tent. ATC and ARL now look at the problem in a holistic way. The new approach, depicted in the Data Management bubble in Figure 2, implements a Hadoop-based ecosystem which addresses all four of the 'tent poles' at the same time.

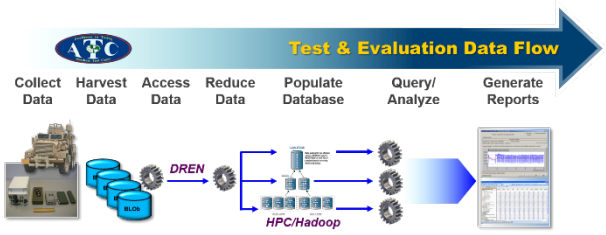

Figure 3: Test Data Workflow.

The proposed Hadoop ecosystem consists of tightly integrated projects from the Apache Software Foundation. A major motivation for this path is the huge support of the open-sourced community that has greatly contributed to these projects, the foundation and its collection of around 350 other projects [17]. Having such a large supporting group lowers the amount of framework based maintenance required. The projects selected for the Data Management ecosystem can be separated into 3 groups: ingestion, storage and manipulation. Ingestion is made up of NiFi[18], Kafka[19] and Flume[20]. Each of these projects allows data to be efficiently pulled from or pushed into the storage or manipulation group. The storage group is built upon the Hadoop Distributed File System (HDFS) [21] and includes Hive [22], HBase [23] and Phoenix [24]. Hive is geared towards analytics on historic data, like finding long term trends, and it provides a SQL style interface. HBase is used for near real-time analytics. Phoenix resides on top of HBase to give analysts a familiar SQL style interface. Both storage paradigms include a JDBC interface allowing external applications to have direct access to the storage. The manipulation group is made up of Hadoop MapReduce, Spark [25] and TensorFlow [26].

The manipulation group interacts with each of the other groups. For example, Hive calls Hadoop MapReduce for its queries. Analysts can interface with Spark via SQL; they can also design more complex and efficient queries using Python or Scala through Spark than they could with SQL. Analysts write the queries in a web-based notebook called Zeppelin [27], another Apache Software Foundation project designed for interactive queries and which ties together the analytical and visualization components.

The Hadoop ecosystem enables benefits not seen with previously used data solutions. All data types – text, network, weather, geospatial, observational, instrumentational, audio, and video – are stored within the same system, allowing automated rather than manual correlation among the data sets using the same tools. For example, when an anomaly is observed in vehicle instrumentation data, human intervention is currently required. Analysts must examine the maintenance records to determine if a maintenance event caused the anomaly and scan video files to determine if an observable event caused the anomaly. After analysts rule out the aforementioned causes, then they can focus on what went wrong on the vehicle. Automating this process helps eliminate human intervention as a source of delay and error, and it helps elucidate hidden anomalies identified by machine learning algorithms.

Currently, the data management ecosystem runs on a few small test systems, including a dynamic approach using Lawrence Livermore National Laboratory's Magpie[28]. Magpie is a collection of scripts that enable running Hadoop within HPC batch environments. This allows us to scale the number of nodes depending on is the size of the dataset; however, due to overhead and architecture differences, running Hadoop on HPC batch systems is slower and not persistently online for queries. Regardless, having the ability to run on HPC makes the connection to ParaViewWeb and other visualization software easier, as shown in Figure 2.

3.2 Secondary Data Management

Cloudberry [5] can be described as middleware between a data management system and the visualization side of the ecology, as illustrated in the Secondary Data Management bubble in Figure 2. The middleware allows building fast, real-time analytics tools that support interactive visualization with large datasets. It does so by requesting data from the client application more intelligently. Specifically, it completes query requests faster by caching results of previous queries and storing aggregate results, providing potential data reuse to accelerate future related queries and by more quickly ingesting data into the data store on the back end for progressive updates to the data. From the front-end application, interfacing with Cloudberry is simple. One passes a query to the Cloudberry layer in a JavaScript Object Notation (JSON) format as a message via a web socket. The application listens for messages and responds. There is no more complexity on the client side compared to sending a query directly to a database, as the Cloudberry middleware resolves query optimization.

3.3 Visualization Hardware

The visualization hardware bubble in Figure 2 includes several distributed collaborative visualization systems. It ranges from the large high-resolution tiled display as shown in Figure 1, to multi-touch displays, to personal workstations, to tablets, to head mounted complete immersive virtual reality systems (Oculus Rift

and HTC Vive – Figure 5), to semi-immersive virtual reality system (zSpace – Figure 4), to augmented reality display system (Microsoft HoloLens – Figure 7), to the users themselves.

3.4 Visualization Software

The visualization software bubble in Figure 2 shows scientific and none scientific visualization software available in our ecologyA hybrid visualization system [6] capable of combining the benefits of both immersive and non-immersive visualization for a seamless 2D and 3D environment supporting information-rich analysis would overcome some of the challenges. Figure 4 shows ParaView running on zSpace.

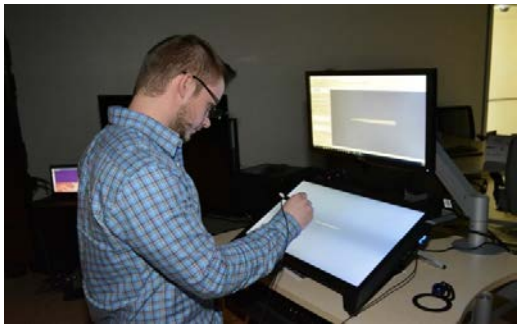

Figure 4: ParaView Running on zSpace.

Figure 5: ParaView Running on HTC Vive.

VisIt [7] scientific visualization software supports client/server mode to take advantage of the HPC system for large scale datasets. ParaView [8] and EnSight [9] software are configured for high resolution on the tiled display and in full VR mode with HTC Vive, Oculus Rift, and zSpace. Figure 5 shows ParaView running on HTC Vive; fully immersive and 3D interactive data visualization. In addition, ParaSAGE [10] extends ParaView to a native SAGE2 [11] application, taking advantage of SAGE2 scalable, high resolution capability. Figure 6 shows ParaSAGE running on the high resolution display wall with keyboard and mouse interaction.

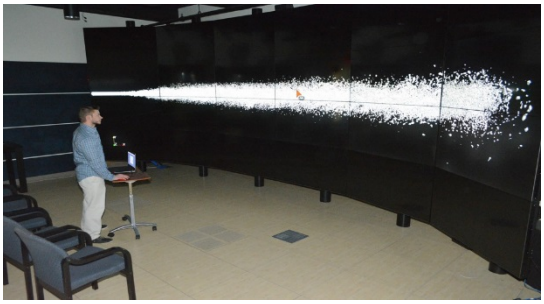

Figure 6: ParaSAGE Running on High Resolution Display Wall.

Unity [12] and Unreal Engine [13] also support virtual reality and augmented reality data visualization application development. Figure 7 shows our augmented reality data visualization application developed using Unity running on Microsoft HoloLens [14]. Presenting the internal combustion engine fuel injection data using different 2D and 3D visualization systems provided the domain scientist with new insight into their computational fluid dynamics model.

Figure 7: Augmented Reality Data Visualization with MS HoloLens.

Figure 8: Information Visualization Using ParaViewWeb Running on SAGE2 Framework.

We further developed our collaborative high resolution data visualization framework [15] on top of the ParaViewWeb framework [16] supporting both scientific and none scientific data within a single framework. Figure 8 shows our SyncVis data visualization application running on a high resolution tiled display within the SAGE2 framework. SyncVis allows the user to display multiple variables using multiple diagrams with one streamlined view. The ParaViewWeb framework allows synchronizing data displayed using different visualization views; selecting a variable in one view highlights the variable in all the views where the same variable appears.

4 Discussion and Case Study

We are applying this software with Army test data for automotive systems, communications systems, and logistics data. Automotive and communications system tests produce 10s of terabytes of data for a single system. ParaViewWeb allows building an automated workflow, from data ingestion to visualization and interrogation of the data. Our VAE is still a work in progress with the eventual goal to build a work flow for customer applications and handoff the tool to the customer while we continue development of improved capabilities and expand the tools to encompass additional customer requirements. Having ATC as part of the requirements and development team provides a natural two-way communication for ARL to understand user requirements, for the customer to understand the visualization capability, and to naturally transition the capability to the customer already trained in its use. The

ongoing relationship facilitates sustaining the capability – with bug fixes, updates, new features, and port to new platforms.

In a typical use case, the user uploads simulation and sensor data and creates visualizations using the chosen visualization technique. Once created, the user has the option to run the visualization on the most suitable display. For example, during a test, ATEC collects sensor data in the field. ATEC also employs physics-based simulation to correlate with the measurements. In one example, communication system test data is incorporated in a large spreadsheet. The data include radio ID, lat and long coordinates, if the radio was transmitting, the start and stop times of transmission, if the radio was receiving, start and stop times of message, waveform used, and other data. Spreadsheets are limited in the amount of data per tab, restricting analysis to small windows of time and number of interacting systems. We ingested the test data into ParaViewWeb and were able to quickly isolate outlier anomalies, visualize the data, and provide analysts access to datasets not restricted to small numbers of systems or periods of time. SyncVis data visualization, shown in Figure 8, is used to analyze large amounts of sensor data.

Figure 9: EnSight Running on High Resolution Display Wall showing data from an internal combustion simulation

For 3D temporal and spatial data generated by simulations, the user employs a fully-immersive approach to analyze the data, after which the user can use the virtual reality display for additional immersion in the simulation data. Figure 9 shows a user employing the high resolution display wall to discover overall structure of a large internal combustion simulation. The figure captures a single snapshot of a simulation that consists of 80 million grid nodes distributed over 5,000 processors and run for 340 wall-clock hours for thousands of time steps; it generated terabytes of data. The image illustrates the ability to interact with the combustion chamber flow and fuel spray in the presence of valve and piston dynamic motion and chemical reactions. The cyclic process is animated on the display wall and the user controls the progress and resolution, revealing the complex interacting physical processes in this high resolution multi-physics simulation.

The VAE is the realization of an interactive ecosystem of devices, humans, software and data that provides a framework for which a renewed study of the meaning of interaction and computation is achieved that redefines visual analytics. The applicability of such a system provides new understanding for data science.

5 Conclusion

We do not believe one size fits all in visualization. Our VAE gives the user the flexibility to display the data using the software and hardware best suited for the specific analysis and exploration task. Depending on the computational requirements, the user can tap into HPC resources to shorten the processing time required. As interactive visualization requires near real time processing of the data, we further improve the overall performance of the ecology using middleware like Cloudberry. Moving forward, we plan to streamline the process by making available additional visualization techniques, by automating the data ingest step, by expanding the use on additional visualization devices, and by building automated work flows for our customer applications.

References

[1] J. Michael Barton and Raju Namburu, “Data-Intensive Computing for Test and Evaluation,” ITEA Journal, 38 (2), June 2017, pp. 144-152.

[2] A. Endert, M. Hossain, N. Ramakrishnan, Chris North, P. Fiaux, and C. Andrews, "The human is the loop: new directions for visual analytics," Journal of Intelligent Information Systems, pp. 1-25, 2014

[3] Ken Renard, Brian Panneton, Gregory Bessack, James Adametz and Brian Tauras, “Processing and Analysis of Large Data Sets from High Bandwidth Tactical Networking Experiments Using High Performance Computing,” The ITEA Journal 36 3, pp. 220-225, 2015.

[4] Brian Panneton, Brian Henz, P. Patel, and James Adametz, "Processing and Analysis of Large Data Sets Using High Performance Computing: Beyond Experimental Data,” Int’l Conf. on Advances in Big Data Analysis. ISBN 1-60132-427-8, July 2016

[5] Jianfeng Jia, Chen Li, Xi Zhang, Chen Li, Michael Carey and Simon Su, "Towards Interactive Analytics and Visualization on One Billion Tweets," 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2016), Monday October 31 - Thursday November 3, 2016 — San Francisco Airport Marriott Waterfront, California, USA

[6] Simon Su, Aashish Chaudhary, Patrick O'Leary, Berk Geveci, William Sherman, Heri Neito, and Luis Francisco- Revilla, "Virtual reality enabled scientific visualization workflow," 2015 IEEE 1st workshop on Everyday Virtual Reality (WEVR), 23 March 2015.

[7] E. Wes Bethel, Hank Childs and Charles Hansen, High Performance Visualization: Enabling Extreme-Scale Scientific Insight, Chapman & Hall/CRC, 2012

[8] KitWare. Paraview. http://www.paraview.org/

[9] CEI. EnSight. https://www.ensight.com/

[10] Dylan Kobayashi, Simon Su, Luis Bravo, Jason Leigh, Dale Shires, ParaSAGE: Scalable Web-based Scientific Visualization for Ultra Resolution Display Environment, IEEE Visualization 2016, Poster, 23-28 October 2016, Baltimore, Maryland, USA

[11] T. Marrinan, J. Aurisano, A. Nishimoto, K. Bharadwaj, V. Mateevitsi, L. Renambot, L. Long, A. Johnson, and J. Leigh, “SAGE2: A New Approach for Data Intensive Collaboration Using Scalable Resolution Shared Displays” (best paper award), 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing. 2014.

[12] Unity. https://unity3d.com/

[13] Unreal Engine. Epic Games, Inc. https://www.unrealengine.com/

[14] Microsoft HoloLens. Microsoft. https://www.microsoft.com/en-us/hololens

[15] Simon Su, Vincent Perry, Nicholas Cantner, Dylan Kobayashi and Jason Leigh, "High-resolution interactive and collaborative data visualization framework for large-scale data analysis," International Workshop on Visualization and Collaboration (VisualCol 2016), October 31 - November 04, 2016, Orlando, Florida, USA

[16] ParaViewWeb. ParaView. https://www.paraview.org/web/

[17] "Projects Directory." Apache Projects Directory. N.p., n.d. Web. 29 June 2017. <https://projects.apache.org/>.

[18] "Apache NiFi." Apache NiFi. N.p., n.d. Web. 29 June 2017. <https://nifi.apache.org/>.

[19] "Apache Kafka." Apache Kafka. N.p., n.d. Web. 29 June 2017. <http://kafka.apache.org/>.

[20] "Welcome to Apache Flume." Welcome to Apache Flume — Apache Flume. N.p., n.d. Web. 29 June 2017. <http://flume.apache.org/>.

[21] "Welcome to Apache™ Hadoop®!" Hadoop. N.p., n.d. Web. 29 June 2017. <http://hadoop.apache.org/>.

[22] "General." Apache Hive TM. N.p., n.d. Web. 29 June 2017. <https://hive.apache.org/>.

[23] "Apache HBase – Apache HBase™ Home." Apache HBase – Apache HBase™ Home. N.p., n.d. Web. 29 June 2017. <https://hbase.apache.org/>.
[24] "Overview | Apache Phoenix." Overview | Apache Phoenix. N.p., n.d. Web. 29 June 2017. <https://phoenix.apache.org/>.
[25] "Apache Spark™ - Lightning-Fast Cluster Computing." Apache Spark™ - Lightning-Fast Cluster Computing. N.p., n.d. Web. 29 June 2017. <https://spark.apache.org/>.
[26] "TensorFlow." TensorFlow. N.p., n.d. Web. 29 June 2017. <https://www.tensorflow.org/>.
[27] The Apache Software Foundation. "Apache Zeppelin." I'm Zeppelin. N.p., n.d. Web. 29 June 2017. <https://zeppelin.apache.org/>.
[28] Chu, Albert. "LLNL/magpie." GitHub. N.p., 15 June 2017. Web. 29 June 2017. <https://github.com/LLNL/magpie>.
[29] "Introduction." Matplotlib: Python Plotting — Matplotlib 2.0.2 Documentation. N.p., n.d. Web. 29 June 2017. <https://matplotlib.org/>.
[30] "VTK - The Visualization Toolkit." VTK. Kitware, n.d. Web. 05 July 2017. <http://www.vtk.org/>.