2013 IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum

Proactive support for large-scale data exploration

Mark Hereld
Computation Institute
Argonne National Laboratory
Email: hereld@anl.gov

Tanu Malik
Computation Institute
The University of Chicago
Email: tanum@ci.uchicago.edu

Venkatram Vishwanath
Argonne National Laboratory
Email: venkatv@mcs.anl.gov

***Abstract*—Computational science is generating increasingly unwieldy datasets created by complex and high-resolution simulations of physical, social, and economic systems. Traditional post processing of such large datasets requires high bandwidth to large storage resources. In situ processing approaches can reduce I/O requirements but steal processing cycles from the simulation and forsake interactive data exploration. The Fusion project aims to develop a new approach for exploring large-scale scientific datasets wherein the system actively assists the user in the data exploration process. A key component of the system is a software assistant that evaluates the stated and implied analysis goals of the scientist, observes the environment, models and proposes actions to be taken, and orchestrates the generation of analysis and visualization products for the user. These products are managed and made available to the scientist through an interactive space consisting of a database and a visual interface. The scientist sifts through and explores the available analysis products while indicating preferences that are translated into goals, completing the feedback loop that steers the assistant in its future actions.**

I. INTRODUCTION

Datasets in fields as wide ranging as cosmology, biology, high energy physics, materials science, economics, uncertainty quantification, and chemistry are critical catalysts of progress, innovation, and our general welfare. They enable us to better care for our citizenry through health and social services, understand the complex forces that shape our lives, and improve our productivity and overall well-being. But for a variety of technical reasons, including the distance between data sources and the relative gains in computing compared with storage performance, it is becoming increasingly impractical to subject datasets of interest to brute-force analyses.

Exploring data by traditional means is fast becoming intractable. It takes far too long to make an analysis pass over today's large datasets to support exploration in any practical way. Our strategy? Enable analysis on subsets, navigable summary data, and guided analytic dives. This will require means for effectively scheduling compute, storage, and communications resources against targeted analysis goals in order to produce, capture, and present stream-of-analysis products for the scientist to explore. We view this as a critical data capture and management dimension of the big data problem. At the same time, we will need to develop new scalable data visualization techniques and tools to handle this stream and allow the scientist to guide exploration and production of the elements in the stream.

The Fusion project aims to develop a new method for exploring large-scale scientific datasets. It will include a software assistant that evaluates the stated and implicit analysis goals of the scientist and provides analysis and visualization products in response. The interactive visualization space to be developed will enable the scientist to engage these available data products by selecting, arranging, and manipulating them.

We have begun to design a prototype framework to help us develop and test this new Fusion paradigm. The Fusion Assistant will respond to (1) goals set explicitly and implicitly by the user, (2) models of the analytics environment and the analysis algorithms, and (3) live monitoring of the analytics environment. It will produce a stream of analyses chosen to maximize cost-benefit based on these input streams. Because the assistant operates autonomously and can adapt to the changing requirements placed on it by the scientist and the environment, it will provide proactive support for the exploration process, enabling the scientist to focus on the data rather than the details of producing and analyzing it. The analysis products will be managed in a database and available to the scientist through an interactive interface. Using this Fusion VisSpace, the scientist will be able to sift through and explore the available analysis products while indicating preferences that will be translated into goals to steer the Assistant's future actions. Principal capabilities built into the VisSpace will be interactive query, autonomous summary generation, and visual navigation. Queries will draw on data from the results already produced and will spur the assistant to generate data where lacking, contributing to a smooth exploration experience.

The planned system will provide for a collaborative experience of data exploration wherein the scientist evaluates the meaning of the analysis while steering the software assistant's production of further analyses. By favoring piecemeal analysis and summary activities over very time-consuming wholesale analyses, we believe that Fusion will enable the scientist to develop understanding while building informed criteria for later offline analysis by more traditional means.

The Fusion paradigm that we are designing can be thought of as a twist on the traditional idea of computational steering. In that paradigm the human is added to the loop in order to supply the evaluative capacity, using it to control the course of the simulation or analysis under way. Fusion shifts the balance, giving the Fusion assistant significant latitude for evaluation and adaptation. The human in this loop is nudging rather than

978-0-7695-4979-8/13 $26.00 © 2013 IEEE
DOI 10.1109/IPDPSW.2013.48

steering the course of the analysis.

II. The Case for Proactive Support

Fusion does not target problems where the path to the answer is known in advance and can be addressed by predetermined analyses. For big data applications, these require a long time to crunch. Instead, we are interested in problems where exploration of the data is required in order to build understanding, develop strategies for discovering unknown features, or determine the parameters of a fixed wholesale analysis (which would then take a long time). For these problems it is impractical to interleave time-consuming analysis and visualization steps with engaged reading of the results. Instead, we pursue development of means for rapidly exploring big data despite its fundamentally unwieldy scale.

To achieve this, we consider the value of guided spot evaluations, analyses, and visualizations to build understanding of the dataset under observation. Arguably, the scale of the data and the computational resources required to generate a valuable stream of reductions will require a good deal of attention to the work of orchestrating the processes involved. The required attention threatens to overwhelm the scientist interested in the results. We therefore propose to design proactive software to attend to these parts of the exploration process, leaving the higher-level aspects to the scientist-in-the-loop.

A. Driven by Large-Scale Science

We have been working with application domain scientists to develop requirements for computational infrastructure in order to simplify their data management needs and analysis processes. Two high-priority examples for this project are sketched in the following paragraphs. Both will produce extremely large and rich datasets.

Extreme-scale simulations of the universe. Astronomy has excelled in developing comprehensive sky surveys. As survey coverage and depth have increased, so has the simulation requirement for interpreting the observations and extracting the underlying science. Data from cosmological simulation continues to increase exponentially. A typical astronomical survey data management pipeline takes raw sensor output and produces data products, in the multi-petabyte regime, that are then sent to the analysis pipeline, the results of which are then used for scientific inference. Corresponding large-scale simulations must work with much larger datasets, much of which never make it to the storage system.

In order to analyze the available and anticipated wealth of observational data, sophisticated simulations of the formation of structure in the universe are essential. These simulations have many uses: (1) to advance our understanding of structure formation, (2) to test out new analysis methods on simulated skies, (3) to explore the effects of different cosmological models on the large scale distribution of galaxies, (4) to extract cosmological parameters from the observations and build precision prediction tools, and (5) to develop robust error estimates. A large suite of simulations has to be carried out to address these tasks.

Measuring the human connectome. Recent years have seen incredible progress in neuroscience. Researchers trying to understand how the brain is laid out are developing the technologies needed to directly determine the neural connectivity of increasingly large volumetric samples. Starting with precise, automated high-throughput slicing and image digitization, stacks of images are analyzed in detail to reconstruct 3D models down to synaptic connections. The human brain has some 100 billion neurons and 250 trillion synapses. Resolving 20-nanometer features might be practical with five-nanometer imaging resolution on 30-nanometer thick slices. But Jeong et al. [1] note that at this resolution, digitizing only a cubic millimeter would produce a petabyte of raw data.

The currently practical approach to analyzing the images includes segmenting them into classifiable object fragments, aligning neighboring slices and correcting for distortion of the slice images, and fitting features through the volume to build three dimensional models of the thicket of dendritic fibers, synaptic connections, and neuron bodies. It is computationally intensive and requires, for the present, human intervention to ensure correctness of the automated findings.

B. Related Work

A number of tools have been developed to explore data managed in database or spreadsheet forms (Polaris [2] now commercialized as Tableau [3]–[5], XdmvTool [6], VisTrails [7]). None provide for proactive generation of an analysis stream, although Polaris/Tableau support automatic generation of visual views. A notion of zoom described in connection with Polaris considers lattices of parameterized visualizations on nodes in a cubic graph [2]. This language of visual abstractions as nodes is analogous to the Fusion notion of a configuration and related analysis product, but again it includes no notion of continuous autonomous generation of these nodes.

Algorithms for layout of and interaction with collections of data instances have been encapsulated in a number of available toolkits and packages. Among the most widely used are D3 [8], Graphviz [9], Information Visualization Cyberinfrastructure software framework (IVC) [10], Piccolo Toolkit [11], Prefuse [12], Titan [13], and Tulip [14]. We will be leveraging the investment in these in implementing Fusion.

In the high-performance computing arena, refactoring the traditional graphics pipeline in order to better utilize computational resources is one approach being considered for ensuring performance of large-scale analyses [15]–[17]. The focus here is on wholesale analysis rather than exploration.

Beaudouin-Lafon [18] divides the relationships between human and computer into three classes: computer as tool, computer as partner, and computer as medium. Fusion adopts the computer-as-partner perspective. Moere [19] has proposed using agent-based methods to autonomously create emergent visual representations that reflect underlying priorities built into the agents. This approach is not incompatible with the Fusion VisSpace but is not our current focus.

Rimey and Bolme [20] have studied the value of saving and reusing millions of visualizations compared with the cost of keeping them. The visualization instances they collect and reuse are similar to our notion of a configuration. Whereas their instances are manually generated by scores of intelligence analysts, Fusion significantly focuses on autonomous generation of these instances.

A theoretical foundation is provided by Purchase et al. [21] that may help us in our early design phase of the Fusion visual interface. The authors attempt to enable early evaluation of information visualization methods in order to avoid the effort of costly implementation in the event of an ineffective concept. Rimey and Bolme [20] also enumerate a set of principles for effective spatial organization of large numbers of visualization instances. There is also an extensive literature on human factors that we will not directly engage with in this paper. Instead we will focus on evaluations of user feedback and performance metrics where possible, in order to provide a useful gauge of the success of our work.

C. *What's Missing?*

Big data analysis and visualization require efficient use of computational resources. Exploration requires interaction. Enabling exploration of big data will require careful scheduling of resources against changing goals and conditions on relatively short timescales, a juggling act that begs for a software solution. What is missing from the currently available range of software tools and ideas is a method for building support for these activities into a framework that enables a scientist to interactively explore big data. Proactive support would entail adaptive response to changing circumstances and provide meaningful assistance to the research scientist. This is the idea that defines Fusion.

III. Design Overview

Exploration requires interaction. Figure 1 provides a top-level view of the approach Fusion takes in integrating interactive visual analysis with autonomous decision-making in order to create an interactive environment for exploring big data. This architecture creates two feedback loops designed to improve performance and experience of the exploration process for big data. The `exploration loop` finds the researcher interactively setting goals and priorities that are interpreted by the Fusion Assistant in the context of other input it uses. The Assistant orchestrates a stream of responsive analyses that are collected in the Results Database. From there, the Fusion VisSpace environment presents them to the scientist who continuously and interactively explores the currently available analysis results while directing the course of future analytics activity.

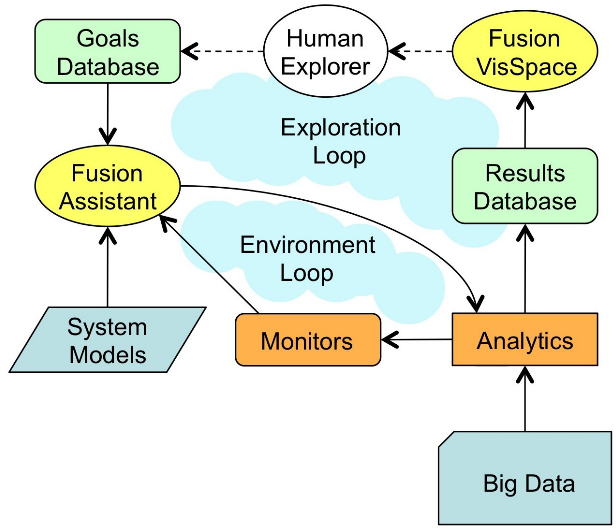

Fig. 1. Overview of the Fusion architecture.

Analytics monitoring in the `environment loop` provides information to the Assistant about the performance of processes in flight so that it may optimally schedule future work to keep the system running efficiently.

A. *Cooperative Data Exploration Interface*

Some of the Fusion analysis products will clearly involve scientific visualization: volume renderings of sub-volumes, coarse-grain overview volume renderings, and so on. But the primary function of the Fusion model for data exploration is to interactively build a collection of views on aspects of the data that will include not only these scientific visualization products but also a wide range of graphical data products such as frequency analyses, data coverage representations, projections, and cumulative summaries. In this way, the Fusion model changes the focus from scientific to *information visualization*. It therefore makes sense to draw on the literature of information visualization as much as or more than on the scientific visualization counterpart.

Interactions. Yi et al. [22] have proposed a taxonomy of seven interaction techniques organized around a user's intent. The authors draw from smaller and differently focused taxonomies developed over the years by other researchers. Some characterized interactions by user goal but were not as comprehensive as this classification scheme. It is a model for high-level description of user goals that will inform the design of the Fusion goals database and interactive goal-setting interface.

They distill interactions in the realm of information visualization into a set of seven classes, summarized in Table I. Such a decomposition is inherently subject to inconsistencies, as they note. We find it to be a suitable starting place, and in fact appropriate for use in describing scientific visualization interactions. We note that Fusion visualization includes elements

Interaction Class†	Description†	Exploration	Goal-Setting
Select	Mark something as interesting	Highlight	Add new goal
Explore	Show me something else	Pan	Extend result set
Reconfigure	Show me a different arrangement	Rotate	Crossmix goals
Encode	Show me a different representation	Histogram	Change layout
Abstract/Elaborate	Show me more or less detail	Zoom	Change LOD
Filter	Show me something conditionally	Query	Adjust priorities
Connect	Show me related items	Correlate	Parameter sweep

TABLE I
GOAL-CENTRIC INTERACTION CLASSES. †TAKEN FROM YI ET AL. [22]

of both information and scientific visualization. The Fusion VisSpace is distinguished by the need to map interactions into two separate but related kinds of actions.

- **Exploration:** First, and most typical, it will enable the user to interact with data drawn from the Fusion Results Database. These actions support exploration of the available data products generated by the Fusion Assistant. For example, the *Select* interaction might cause an item or set of items to be highlighted so as to facilitate visual tracking of the selection set through other manipulations.
- **Goal-Setting:** Second, it will enable the user to identify, modify, and prioritize the analysis goals that drive the Fusion Assistant. In the context of guidance, the *Select* interaction might add a new goal to the goal set.

The final columns in the table indicate some of the context sensitive mappings that might be assigned. Contextual cues include the gesture or event generated by the user (mouse click) and the setting of that event (screen location and frontmost object).

Representations. The visual interface and the interactions it will mediate must fulfill three objectives for Fusion.

- **Browsing and exploring in the Results Database:** Data drawn from the Results Database in response to interactive queries will be presented in a variety of layouts, exposing the user to different possible ways to correlate the information in each visualization instance.
- **Summary and overview of available results:** Visual representation of individual analyses with respect to the data volume, available data variables, and other individual analysis results will provide the user with needed context. For example, if small subvolumes are being explored, a map of what has already been rendered and what is still uncharted would give the user information about morphology, coverage, and relationships.
- **Activities and goals of the Fusion Assistant:** A linked visual representation of the goals in effect, the target mix of analyses, and the currently executing mix of analyses will indicate level and kind of activity that characterizes the Assistant at any time.

B. Proactive Assistant Design

Fusion takes a principled approach in providing assistance. The bases for this principled approach has its roots in control theory [23], [24], autonomic computing in software systems [25], [26], and database administration [27]–[32], where the cost of ownership of the system is high and the system must self-reflect and self-adapt in order to provide assistance to the human. We believe that a comprehensive application of this approach for analysis and visualization is relatively unexplored but is critical for solving big data science problems.

To apply this approach practically, Fusion includes the Fusion Assistant (FA) a decision-making assistant that realizes causality of events, enables prediction, and recommends action. Figure 2 shows the assistance provided by FA to cosmology scientists in their regular work of processing large amounts of simulation data, through multiple analysis paths, and finally viewing analysis and visualizations. The analysis itself is a sequence of steps consisting of transforming data into an appropriate form, organizing for efficient analysis, applying data mining algorithms, building an appropriate visualization model, and refining parameters. Each step (shown by dotted line) consists of several choices (the circles). For instance, there can be several methods to transform data based on mathematical expressions, range expressions, or scale and orientation. The job of Fusion is to help the human in the loop make these decisions effectively, based on feedback, goals, constraints, and hardware configurations.

The principled approach as it applies to the Fusion Assistant comprises three phases.

1) **Observe:** The objective of the observation phase is to determine which quantities to examine at a given time and detect significant shifts in them. It entails reconnaissance of the static VisSpace environment and a continuous monitoring of the dynamic quantities, such as user interactions, data request workloads, feedback, goals, performance meters such as throughput and response time, and data distributions.
2) **Model:** The objective of the model phase is to predict the best configuration for the system among various candidate configurations and the observed state. The objective is to assess quantitatively the impact of adjustments made to various candidates, based on observation. The candidate knobs are evaluated *hypothetically*, and thus it needs utility/cost functions and a mathematical model for the specific function:

$$observations \times configs \leftarrow performance\ metrics. \quad (1)$$

It is important that the model choose a configuration

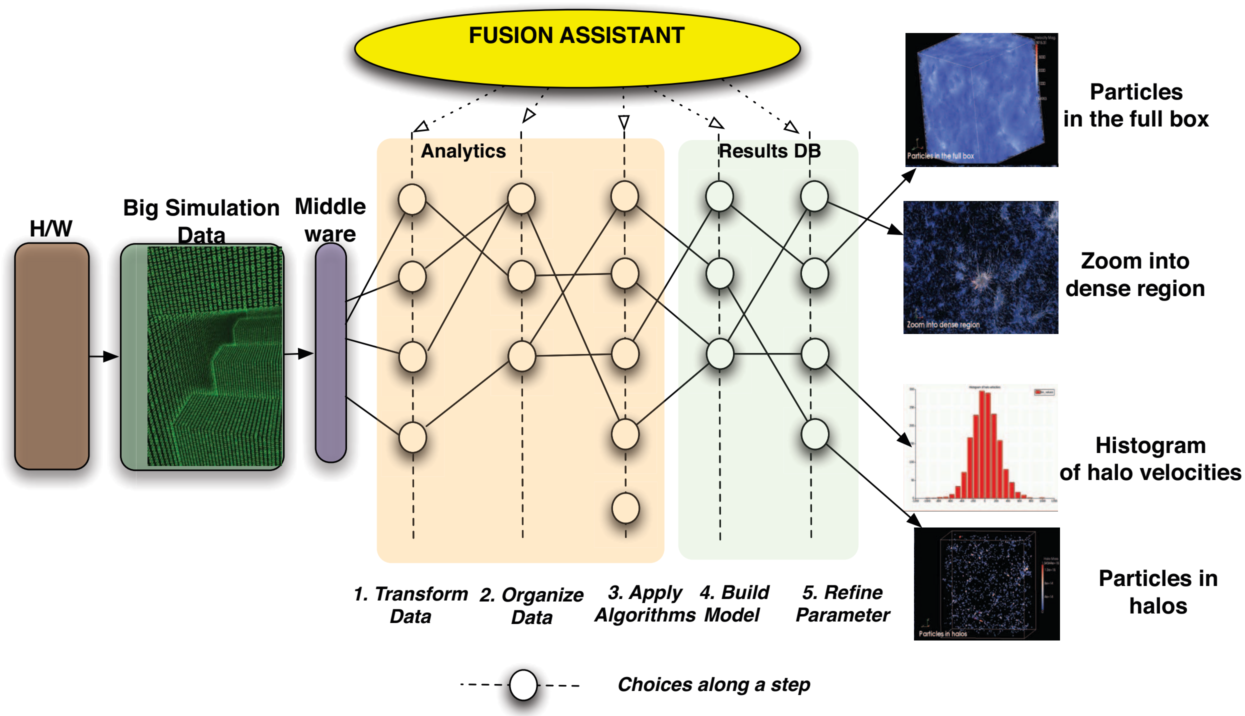

Fig. 2. Fusion Assistant assists the user in choosing the best choices in steps that lead from big data to its analysis and visualization.

when we predict a significant improvement and that the system can be expected to remain operationally stable. The stability check is important, for we might otherwise improve only certain aspects while introducing a risk of degrading the system in other regards. Thus a robust consequence evaluator is essential, which evaluates on the cost scale all possible choices and the cost of transitioning between choices.

3) **React:** As an abstraction, the react phase is the simplest of the three phases: when the evaluation step gives us a clear recommendation on which parameter should be adjusted by how much, one merely has to turn the knob. However, this online adjustment may create engineering problems: it is not easy to build a system where all tuning parameters can be dynamically varied while the system is serving its workload. We believe that a robust translation of these phases within a visualization-analysis environment depends on careful interpretation and implementation of these steps. In the above we have outlined our broad understanding of these principles in the context of the visual analysis environment. We now describe our implementation choices, which have been carved after careful consideration based on specificity of the environment and the needs of the applications.

IV. Main Implementation Issues

In the following subsections we discuss what we see as the main implementation issues to address in each of six areas in the design.

A. *Generate Analytics and Monitor*

Analytics consists of a set of data products, including a collection of views on portions of the data, scientific visualization products, and a range of graphical data products. Monitor determines what statistics to generate and will provide a better view of the internal state of the analytics area.

Managing statistics on data. In general, selecting statistics to generate statistical information over petabytes of data is a difficult task [29]. We will explore the problem by creating a feature similar to the auto-create-statistics feature present in databases, which causes the database server to automatically generate all single-column histograms via sampling and multi-column histograms via classification and regression functions. Note that we cannot afford to load data into the database systems because of the sheer size of the data, which in simulation environments resides predominantly in files. Therefore, other mechanisms will have to be explored, such as creating a shell database that contains all the statistics but not the data itself.

Selecting statistics to auto-create also implies being able to evaluate the usefulness of a statistic without creating it. Database servers use syntactic criteria. For a wider class of statistics (such as multicolumn), however, syntactic criteria alone are not sufficient and more powerful pruning of candidate set is desirable. We will explore magic number sensitivity analysis (MNSA) [33], a technique proposed to address this problem. The key idea is to impose a necessary condition before a syntactically relevant statistic materializes. Although these were proposed for static query workloads, we will explore them for ad hoc queries more typical of visual analytics workloads.

We will also explore how to tune statistics based on execution feedback, an area in which we have previous experience [34]–[36] in the case of histograms. Self-tuning histograms use execution feedback to bias the structure of the histogram so that frequently queried portions of data are represented in more detail compared with data that is infrequently queried. Since our techniques scale to whether the queries are range queries or user-defined functions, we believe that the techniques will provide a good basis for ad hoc visual analysis. More work is needed, however, to understand the applicability of these techniques when combined with visual feedback and analysis.

Improving monitoring infrastructure. To be able to get a deeper insight into the workings of analytics, we will need an event monitoring infrastructure. These events can currently be obtained from visualization tools. The objective is to monitor events accurately and in a timely fashion. Our primary approach will be to utilize provenance systems for monitoring to monitor the internal state of the server, diagnose problems, and tune solution,. We will utilize systems such as SPADE [37]–[39]. that provide fine-grained provenance over kernel events and a efficient mechanism for querying. The latter is vital for monitoring. Most monitoring systems limit the selection of attributes to monitor and their thresholding. But by integrating detailed provenance, it will become easy to pose questions such as "Identify instances of a goal-centric interaction classes that execute more than twice as slow as the average instance over a window of the last 10 executions." Of course, the challenge in supporting such monitoring queries is to ensure that the overhead is not high, which is currently a problem in provenance systems. We need a tighter integration of provenance into the monitoring infrastructure.

Monitoring of system variables and workloads. We will determine hardware and network configurations and measure usage of the network, I/O reads and writes, throughput, and response times. Most data generation will be on HPC systems, which have fairly detailed performance characteristics and in some cases require appropriate benchmarks to be run. Our objective will be to determine when to benchmark and place priority on the performance characteristics.

Our monitoring infrastructure also includes feedbacks and workloads. Most visualizations systems provide profiling tools; we will use the tools to gather our workload of analysis request sequences and user feedback on each request. We will also monitor workloads to derive useful metrics or detect shifts in data access or analysis patterns through change in access syntax. If our workload becomes very large, we will explore compressing workloads or creating representative workloads over a period of time using sampling and mining techniques.

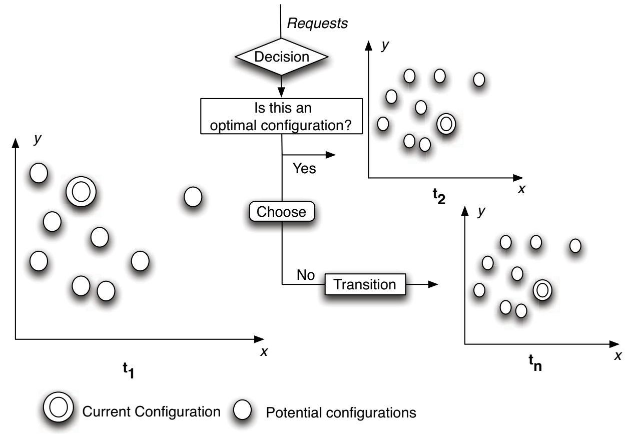

Fig. 3. How system models are used to inform decision-making of Fusion Assistant.

B. System Models and Design of Algorithms

Models will guide the Fusion Assistant to the best decision available in order to satisfy the goals.

A generic system model is described by (i) an enumeration of candidate configurations, (ii) an evaluation of cost of executing visual analytics requests in a current configuration and hypothetically in nonrealized configurations, and (iii) a determination of the cost of moving between configurations. Using such a generic system model, Fusion Assistant decides when the current configuration choice is no longer optimal and chooses a new configuration and transition to it such that excessive costs are not incurred in moving relative to the benefit. Figure 3 shows an interaction between system model and Fusion Assistant. Tasks are divided based on the system model description.

Enumerating the configuration space. Even for a simple self-adaptive task such as selecting the right combination of data variables to choose to visualize, the space of all potential configurations is exponential. Therefore, it is necessary to choose a configuration subspace that is tractable in size. Often the pruning of the configuration space is domain specific. The task will be to quickly find some domain specific methods that prune space, such as by removing configurations that are too costly to realize or do not serve the predominant workload or by merging the configuration space.

Develop adaptation algorithms. We will develop three variations of algorithms offline, time-bound, and online based on the level of response/interactivity required.

In *offline adaptation*, snapshots of the analysis environment, including the state of the hardware, data, and user request, are compared with preanalyzed environment snapshots in order to determine optimal action. Creating good representations may require considering events within a snapshot as a sequence or a sequence of sets. The first approach has been previously explored by us in the context of database systems [40], but the latter representation is indeed a better representation of the real-world situation in which the state of the system varies significantly over time.

Time-bound adaptation will act when system time-to-adapt constraints are in effect. Fusion will need to make judicious tradeoffs especially concerning how much input to take from the analysis environment and what aspects of the system to adapt. In order to find optimal configuration, offline and time-bound adaptation will consist of a greedy search over the configuration space. This could be bottom-up or top-down, each of which has different merits. The bottom-up strategy begins with the empty (or pre-existing configuration) and adds configurations in a greedy manner. This approach, as demonstrated previously [41]–[43], can be efficient when there are strict constraints of storage or time, since the best configuration is likely to consist of only a few variables that have been adapted. In contrast, the top-down approach [44] with a globally optimal configuration could be infeasible if constraints are violated. The search strategy then progressively refines the configuration until it meets the constraints. The latter can be more suitable for visualization that is strictly goal-based.

Online adaptation will support dynamically response-based, continuous tracking of the analysis environment. This type of adaptation might be the only option for situations where the variables in the analysis environment change too unpredictably. For example, when the system consists of a large number of users and the analysis requirements vary significantly, it is difficult to predict the best response that satisfies all users. We will model online adaptation following the task systems developed by Borodin and El-Yaniv [45]. Task systems have been researched extensively in theory [45]–[49]. This was first demonstrated for database systems by developing online algorithms for task systems that search the space of physical design alternatives over a continuously incoming workload of database queries [50]. A significant goal will be to develop online algorithms that take $O(N)$ time at each step in the worst case, instead of $O(N^2)$ algorithms, which are the norm in task systems. N refers to the number of configurations in the configuration space.

Estimating costs. In our generic model we consider two primary kinds of costs: cost for performing visual analysis in a current system configuration and cost for transitioning from current configuration to a different more desirable configuration. In general, costs are assumed to be positive and asymmetric when transitioning between configurations. We will explore different cost functions including scalar, linear, linear over time, and exponential. In [50], [51] a novel approach was introduced using caching and query optimizers.

C. Recommendation and Conducting

In orchestrating a recommendation from the System Model, the most important system metric is the time to adapt, which is the time taken between the identification that a change is required until the change has been effected safely and the system moves to a continue state.[1]

Two approaches exist for orchestrating a recommendation: automatically and manually by a person monitoring the system. Two problems are of particular interest.

Translating recommendations. An important implementation issue with respect to realizing a recommendation is that Fusion will be working on a test server initially. Any recommendation that it gives will need to be translated to the requirements of the production server. While the analysis environment in Fusion will be simulating the hardware characteristics of the production server, the result will still be a function of tuning done on the test server, whose actual hardware characteristics are different. We therefore will have to ensure that the recommendations obtained are identical, that is, the production server was tuned directly.

Improving recommendations through Hot Configurations. Another important implementation issue is realizing some hot configurations of the system. Prerealizing some configurations that seem to be gaining advantage in the configuration space but will take too much time if the system is supposed to suddenly switch to that particular configuration can be of significant advantage. This technique has potential when there is more than one way of organizing the data, since physical data organization can take a lot of time. The technique is also potentially useful when data needs to be moved from a remote location to a local location. Keeping a cache of objects close to the local location with some desirable objects provides advantages [51]. We also need an organized feedback loop to gather input from the user and translate it to appropriate system requirements.

D. Support for Goal Setting

Some of the user's actions in the interactive VisSpace will set the analysis agenda to be followed by the Fusion Assistant.

Describing goals. The vocabulary for describing goals will be designed to cover interactions laid out in Table I. We will design the Goals Database (GD) to be populated with goal types that include configuration settings used by the Fusion Assistant and priorities to influence the analysis work mix. A goal such as "more of analysis objects like this" would use configuration settings found in the metadata for selected objects to construct a template. The goal with this descriptive template would be entered into the GD for the Assistant to respond to.

Designing the interface to the Goals Database. We will need to provide a stable interface to the GD in order to facilitate concurrent development on both the Assistant and the VisSpace components of Fusion. The actions to be implemented include adding goals to the list and priority

[1] Adaptation time is time to transition to the new state and is different from reaction time, which is time when an environmental element changes and the system recognizes the change, decides what reconfiguration is necessary.

setting and changing. In practice, a modest number of goals will be in effect at any one time, but the interface will include more comprehensive tracking of goal history, motivating our plan to use database machinery.

Implementing goal-setting interactions. Goal-setting interactions will be embedded in several different views on the data and state of the analysis as appropriate. We will need to ensure that interactions with objects in the interface will unambiguously trigger goal-setting or exploration actions. Views on the goal stack will support interactive manipulation of relative weighting of goals. Views on data products will support addition of goals to support a "more like this" interaction style. Views on summary data will enable creation of goals that explore new regions or variables.

E. Enabling Exploration

The user will be engaged primarily in exploring data, which in the Fusion paradigm means the set of analysis results already captured in the Results Database.

Designing the Results Database. Constructing a usable interface to the Results Database will be an early goal, again to facilitate concurrent development of the publisher (Assistant) and the consumer (VisSpace) components. Analysis results will be captured in the database so as to maximize utility of the data. Arrays and vectors of computed results with metadata to enable flexible rendering by the VisSpace processes will be favored where possible. Volume renderings, for example, will capture metadata (units, POV, index of transfer function) alongside the 2D array of pixel values. Tuples of derived quantities will be saved along with metadata to allow display of combinations and subsets as dictated by interactions in the VisSpace.

The Results Database will also include metadata about the big dataset itself: overall size, dimensional information, variables, and format. These will enable the VisSpace to provide the user with variable and analysis options as well as enable contextual summary data to be presented in useful form.

Populating the exploration view with objects from the Results Database. Collections of objects that match user-specified query criteria will be presented for inspection and selection. Queries will be based on attributes of the overall dataset, of individual objects, and of collections of objects, as well as on object fitness functions based on the goals set at any particular time. Objects satisfying the current criteria will be rendered in the exploration view. They will be laid out according to various preferences – object affinity, time order, fitness – using algorithms adapted from other applications.

Developing exploration interactions. We expect that most of the exploration activity will focus on interactions with the objects presented in the exploration view. The principal mode of exploration in Fusion is forming new queries to extract collections of results from the Results Database for review. Selections of objects will support construction of queries for "more like this": same variable different region, same region different variable, same variable and region different resolution, available statistical analyses on objects like this, and so on. These variations are to be distinguished by what portion of the object metadata the user chooses to hold fixed in the query.

F. Developing the Visual Environment

The visual presentation of analysis results, state of the Results Database, and state of the Fusion Assistant will provide the user with insight into the data.

Accommodating exploration, summary, and monitoring with views. The environment will need to allocate limited pixel real estate to three main interactive visual functions. A useful mechanism will render one functional view in its entirety and the other two in a small form that expresses only summary indications of state. This indicator mode for the Assistant monitoring view will let summary statistics or gauges describe the state in thumbnail form. The user will cycle through views according to need. Each view will, for example, be useful in setting or changing goals for the Fusion Assistant.

Building and manipulating summaries and overviews. For volumetric datasets the user will need to know how much and in what distribution analysis products cover the space under study. This data is available from the Results Database. We will implement a simple widget that presents a navigable cube with volume rendering of coverage density for the quantity queried. The visualization code will leverage existing software already available to us and repurposed here to draw from the Results Database.

Visualizing the Fusion Assistant at work. To support the user in setting goals while exploring a dataset, we will include visual status of the Assistant, analysis tasks in progress, and the goal stack currently in force. The presentation will be designed to convey at a glance (1) level of activity in Assistant, (2) computational resource load, and (3) goal priority and relation to Assistant activities and current computational processes. This indicator will provide natural feedback for troubleshooting and training.

V. CONCLUSIONS

In this paper we have introduced the ideas behind our design of Fusion, a framework for proactive support in exploring large-scale datasets such as are generated by scientific simulations on supercomputers. The system aims to provide a man-machine collaboration wherein the user guides the process of automatically generating modest scale analyses and visualizations.

The user navigates through the available data through the agency of a visualization environment backed by a database containing and describing currently available analysis results.

Throughout the interactive process, the user specifies by his actions and focus what parts and aspects of the dataset are of immediate interest. These are captured as a set of shifting goals that drive an automated assistant. It is this Fusion Assistant that orchestrates a continuous stream of analysis tasks whose results pour into the results database. These activities are also governed by the availability of resources as indicated by monitor processes designed to inform the Fusion Assistant.

By favoring piecemeal analysis and summary activities over time-consuming wholesale analysis, we believe that Fusion will enable the scientist to develop understanding while building informed criteria for later offline analysis by more traditional means.

Acknowledgements

We thank Mike Papka, Ian Foster and Salman Habib for input. This work was supported through in part by the U.S. Department of Energy under contract DE-AC02-06CH11357.

References

[1] W.-K. Jeong, J. Beyer, M. Hadwiger, R. Blue, C. Law, A. Vazquez-Reina, R. Reid, J. Lichtman, and H. Pfister, "SSECRETT and NeuroTrace: Interactive visualization and analysis tools for large-scale neuroscience data sets," *Computer Graphics and Applications, IEEE*, vol. 30, no. 3, pp. 58–70, May-June 2010.

[2] C. Stolte, D. Tang, and P. Hanrahan, "Polaris: a system for query, analysis, and visualization of multidimensional databases," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 75–84, 2002.

[3] "Tableau," `http://www.tableausoftware.com`.

[4] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1137–1144, Nov. 2007.

[5] P. Hanrahan, "VizQL: A language for query, analysis and visualization," in *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '06. New York: ACM, 2006, pp. 721–721.

[6] M. O. Ward, "XmdvTool: Integrating multiple methods for visualizing multivariate data," in *Proceedings of the conference on Visualization '94*, ser. VIS '94. Los Alamitos, CA: IEEE Computer Society Press, 1994, pp. 326–333.

[7] L. Bavoil, S. Callahan, P. Crossno, J. Freire, C. Scheidegger, C. Silva, and H. Vo, "VisTrails: enabling interactive multiple-view visualizations," in *Visualization, 2005. VIS 05. IEEE*, Oct. 2005, pp. 135–142.

[8] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

[9] J. Ellson, E. Gansner, L. Koutsofios, S. North, and G. Woodhull, "Graphviz open source graph drawing tools," in *Lecture Notes in Computer Science*. Springer-Verlag, 2001, pp. 483–484.

[10] "Information Visualization Cyberinfrastructure software framework (IVC)," `http://iv.slis.indiana.edu/sw/`.

[11] B. B. Bederson, J. Grosjean, and J. Meyer, "Toolkit design for interactive structured graphics," *IEEE Trans. Softw. Eng.*, vol. 30, no. 8, pp. 535–546, Aug. 2004.

[12] J. Heer, S. K. Card, and J. A. Landay, "Prefuse: a toolkit for interactive information visualization," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI '05. New York: ACM, 2005, pp. 421–430.

[13] P. Crossno, B. Wylie, A. Wilson, J. Greenfield, E. Stanton, T. Shead, L. Ice, K. Moreland, J. Baumes, and B. Geveci, "Intelligence analysis using Titan," in *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, ser. VAST '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 241–242.

[14] D. Auber, "Tulip a huge graph visualization framework," in *Graph Drawing Software*, ser. Mathematics and Visualization, M. Jnger, P. Mutzel, G. Farin, H.-C. Hege, D. Hoffman, C. R. Johnson, K. Polthier, and M. Rumpf, Eds. Springer Berlin Heidelberg, 2004, pp. 105–126.

[15] K. Moreland, U. Ayachit, B. Geveci, and K.-L. Ma, "Dax Toolkit: A proposed framework for data analysis and visualization at Extreme Scale," in *2011 IEEE Symposium on Large Data Analysis and Visualization*. IEEE, 2011, pp. 97–104.

[16] L.-T. Lo, C. Sewell, and J. Ahrens, "PISTON: A portable cross-platform framework for data-parallel visualization operators," in *Eurographics Symposium on Parallel Graphics and Visualization*, 2012.

[17] J. S. Meredith, R. Sisneros, D. Pugmire, and S. Ahern, "A distributed data-parallel framework for analysis and visualization algorithm development," *Proceedings of the 5th Annual Workshop on General Purpose Processing with Graphics Processing Units GPGPU5*, pp. 11–19, 2012.

[18] M. Beaudouin-Lafon, "Designing interaction, not interfaces," *Proceedings of the working conference on Advanced visual interfaces AVI 04*, vol. 222, pp. 15–22, 2004.

[19] A. V. Moere, "A model for self-organizing data visualization using decentralized multiagent systems," in *Advances in Applied Self-organizing Systems*, ser. Advanced Information and Knowledge Processing, M. Prokopenko, Ed. Springer, London, 2008, pp. 291–324.

[20] R. D. Rimey and D. S. Bolme, "Re-using millions of visualizations," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6495, Jan. 2007.

[21] H. Purchase, N. Andrienko, T. Jankun-Kelly, and M. Ward, "Theoretical foundations of information visualization," in *Information Visualization*, ser. Lecture Notes in Computer Science, A. Kerren, J. Stasko, J.-D. Fekete, and C. North, Eds. Springer, Berlin, 2008, vol. 4950, pp. 46–64.

[22] J. S. Yi, Y. A. Kang, J. Stasko, and J. Jacko, "Toward a deeper understanding of the role of interaction in information visualization." *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1224–31, 2007.

[23] A. Milanés and S. Lifschitz, "Design and implementation of a global self-tuning architecture," in *Brazilian Symposium on Database*, 2005.

[24] G. Weikum, "Towards guaranteed quality and dependability of information services," in *Proceedings of the GI Conference on Database Systems for Office, Engineering, and Scientific Applications*. Citeseer, 1999.

[25] S. Chaudhuri and G. Weikum, "Rethinking database system architecture: Towards a self-tuning risc-style database system," in *Proceedings of the 26th International Conference on Very Large Data Bases*, 2000, pp. 1–10.

[26] M. Woodside, G. Franks, and D. Petriu, "The future of software performance engineering," in *Future of Software Engineering, 2007. FOSE'07*. IEEE, 2007, pp. 171–187.

[27] S. Agrawal, N. Bruno, S. Chaudhuri, and V. Narasayya, "Autoadmin: Self-tuning database systems technology," *IEEE Data Eng. Bull*, vol. 29, no. 3, pp. 7–15, 2006.

[28] S. Babu, N. Borisov, S. Duan, H. Herodotou, and V. Thummala, "Automated experiment-driven management of (database) systems," in *Proceedings of the 12th Conference on Hot Topics in Operating Systems*. USENIX Association, 2009, pp. 19–19.

[29] S. Chaudhuri and V. Narasayya, "Self-tuning database systems: A decade of progress," in *Proceedings of the 33rd International Conference on Very Large Data Bases*. VLDB Endowment, 2007, pp. 3–14.

[30] S. Elnaffar, W. Powley, D. Benoit, and P. Martin, "Today's DBMSs: How autonomic are they," in *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*. IEEE, 2003, pp. 651–655.

[31] M. Mamei, R. Menezes, R. Tolksdorf, and F. Zambonelli, "Case studies for self-organization in computer science," *Journal of Systems Architecture*, vol. 52, no. 8, pp. 443–460, 2006.

[32] G. Weikum, A. Moenkeberg, C. Hasse, and P. Zabback, "Self-tuning database technology and information services: From wishful thinking to viable engineering," in *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 20–31.

[33] S. Chaudhuri and V. Narasayya, "Automating statistics management for query optimizers," in *Data Engineering, 2000. Proceedings. 16th International Conference on*. IEEE, 2000, pp. 339–348.

[34] T. Malik and R. Burns, "Workload-aware histograms for remote applications," *Data Warehousing and Knowledge Discovery*, pp. 402–412, 2008.

[35] T. Malik, R. Burns, and N. Chawla, "A black-box approach to query cardinality estimation," *Conference on Innovative Database System Research*, 2007.

[36] T. Malik, R. Burns, N. Chawla, and A. Szalay, "Estimating query result sizes for proxy caching in scientific database federations," in *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*. ACM, 2006, p. 102.

[37] SPADE, "Support for provenance auditing in distributed environments," http://spade.csl.sri.com.

[38] T. Malik, L. Nistor, and A. Gehani, "Tracking and sketching distributed data provenance," in *IEEE Sixth International Conference on e-Science (e-Science)*. IEEE, 2010, pp. 190–197.

[39] T. Malik, A. Gehani, D. Tariq, and F. Zaffar, "Sketching distributed data provenance," in *Data Provenance and Data Management for eScience, Studies in Computational Intelligence series*. Springer, 2012.

[40] T. Malik, X. Wang, R. Burns, D. Dash, and A. Ailamaki, "Automated physical design in database caches," in *IEEE 24th International Conference on Data Engineering Workshop (ICDEW 2008)*. IEEE, 2008, pp. 27–34.

[41] S. Papadomanolakis and A. Ailamaki, "Autopart: Automating schema design for large scientific databases using data partitioning," in *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*. IEEE, 2004, pp. 383–392.

[42] D. Zilio, C. Zuzarte, S. Lightstone, W. Ma, G. Lohman, R. Cochrane, H. Pirahesh, L. Colby, J. Gryz, E. Alton *et al.*, "Recommending materialized views and indexes with the IBM DB2 design advisor," in *Proceedings of the International Conference on Autonomic Computing*. IEEE, 2004, pp. 180–187.

[43] S. Chaudhuri and V. Narasayya, "An efficient, cost-driven index selection tool for microsoft sql server," in *Proceedings of the International Conference on Very Large Data Bases*. Citeseer, 1997, pp. 146–155.

[44] N. Bruno and S. Chaudhuri, "Automatic physical database tuning: a relaxation-based approach," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. ACM, 2005, pp. 227–238.

[45] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. Cambridge University Press, 1998, vol. 2.

[46] M. Manasse, L. McGeoch, and D. Sleator, "Competitive algorithms for on-line problems," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 1988, pp. 322–333.

[47] A. Borodin, N. Linial, and M. Saks, "An optimal on-line algorithm for metrical task system," *Journal of the ACM (JACM)*, vol. 39, no. 4, pp. 745–763, 1992.

[48] M. Chrobak, H. Karloff, T. Payne, and S. Vishwanathan, "New results on server problems," in *Proceedings of the first annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 1990, pp. 291–300.

[49] A. Fiat, R. Karp, M. Luby, L. McGeoch, D. Sleator, and N. Young, "Competitive paging algorithms," *Journal of Algorithms*, vol. 12, no. 4, pp. 685–699, 1991.

[50] T. Malik, X. Wang, D. Dash, A. Chaudhary, A. Ailamaki, and R. Burns, "Adaptive physical design for curated archives," *Scientific and Statistical Database Management*, pp. 148–166, 2009.

[51] T. Malik, R. Burns, and A. Chaudhary, "Bypass caching: Making scientific databases good network citizens," in *Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)*. IEEE, 2005, pp. 94–105.