Evaluation of Filesystem Provenance Visualization Tools

Michelle A. Borkin, *Student Member, IEEE*, Chelsea S. Yeh, Madelaine Boyd, Peter Macko,
Krzysztof Z. Gajos, Margo Seltzer, *Member, IEEE*, and Hanspeter Pfister, *Senior Member, IEEE*

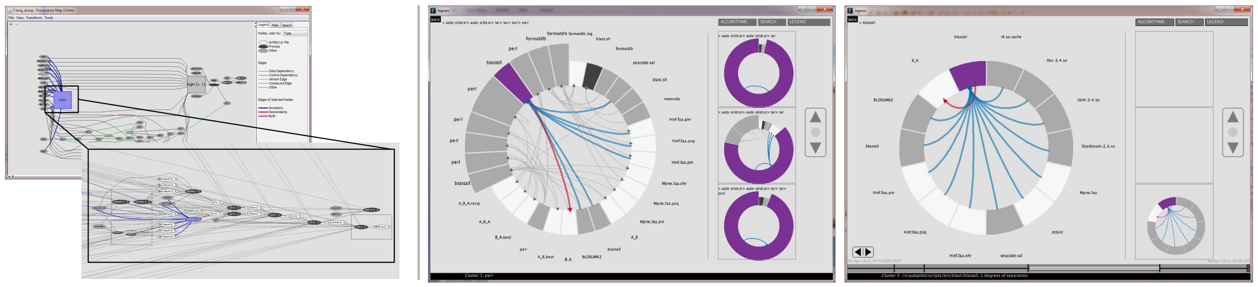

Fig. 1. **Left:** Top: A screenshot of Orbiter, a conventional node-link visualization tool for filesystem provenance data, displaying a data set with the process tree node grouping method. Bottom: A zoom-in on one of the square "super nodes" in Orbiter reveals the sub-nodes and their connections to other nodes. **Middle:** Screenshot of InProv, our new radial-based visualization tool for browsing filesystem provenance data, displaying the same data with the same node grouping as *left*. **Right:** Screenshot of InProv with our new time-based node grouping method with the same data as displayed in the other screenshots (left, middle).

Abstract—Having effective visualizations of filesystem provenance data is valuable for understanding its complex hierarchical structure. The most common visual representation of provenance data is the node-link diagram. While effective for understanding local activity, the node-link diagram fails to offer a high-level summary of activity and inter-relationships within the data. We present a new tool, InProv, which displays filesystem provenance with an interactive radial-based tree layout. The tool also utilizes a new time-based hierarchical node grouping method for filesystem provenance data we developed to match the user's mental model and make data exploration more intuitive. We compared InProv to a conventional node-link based tool, Orbiter, in a quantitative evaluation with real users of filesystem provenance data including provenance data experts, IT professionals, and computational scientists. We also compared in the evaluation our new node grouping method to a conventional method. The results demonstrate that InProv results in higher accuracy in identifying system activity than Orbiter with large complex data sets. The results also show that our new time-based hierarchical node grouping method improves performance in both tools, and participants found both tools significantly easier to use with the new time-based node grouping method. Subjective measures show that participants found InProv to require less mental activity, less physical activity, less work, and is less stressful to use. Our study also reveals one of the first cases of gender differences in visualization; both genders had comparable performance with InProv, but women had a significantly lower average accuracy (56%) compared to men (70%) with Orbiter.

Index Terms—Provenance data, graph/network data, hierarchy data, quantitative evaluation, gender differences

1 Introduction

Provenance is the history of derivation of an object. In filesystems, provenance data is a recording of the relationships of reads and writes between processes and files. In quantitative analysis of scientific data, file provenance offers many benefits. For example, a researcher may receive a third-party data set and wish to use it as a basis for further research or compare the provenance of a repeated experiment to diagnose an error. Without provenance metadata attached, they would have no record of the computations and operations that generated or manipulated that data set. File provenance also offers benefits for IT administrators. Routine administration tasks, such as analysis of log files or finding where viruses were introduced into a system, can be made more challenging by the presence of hidden dependencies. Provenance can expose these dependencies and the interwoven causes of system errors.

Because of these types of potential benefits, systems researchers predict that within the next ten years all mainstream file systems will be provenance aware [43]. However, the provenance data that existing systems generate is of only limited use. For one, the sheer amount of data recorded dwarfs a human's ability to parse through it. Provenance data can be large, sometimes as much as an order of magnitude greater than the data for which the provenance is recorded [31]. Visualization can be a powerful tool for understanding these large data sets.

Many provenance researchers use graph visualization tools to examine inter-relationships on a small subset of nodes. The inability of these tools to visualize large data sets, however, limits the scale at which these data sets can be analyzed and prevents researchers from taking full advantage of the entire provenance database. For instance, a provenance-aware storage system (PASS) recording of a five-minute compilation job of the Berkeley Automator suite of tools has 46,100 nodes and 157,455 edges. Provenance data sets spanning multiple days or even months can grow dramatically in size. Examining only a small subset of the data at one time eliminates the benefits of record-

- *Michelle A. Borkin is with the School of Engineering & Applied Sciences, Harvard University. E-mail: borkin@seas.harvard.edu.*
- *Chelsea S. Yeh is with the School of Engineering & Applied Sciences, Harvard University. E-mail: cyeh@seas.harvard.edu.*
- *Madelaine Boyd is with the School of Engineering & Applied Sciences, Harvard University. E-mail: mboyd@post.harvard.edu.*
- *Peter Macko is with the School of Engineering & Applied Sciences, Harvard University. E-mail: pmacko@seas.harvard.edu.*
- *Krzysztof Z. Gajos is with the School of Engineering & Applied Sciences, Harvard University. E-mail: kgajos@seas.harvard.edu.*
- *Margo Seltzer is with the School of Engineering & Applied Sciences, Harvard University. E-mail: margo@seas.harvard.edu.*
- *Hanspeter Pfister is with the School of Engineering & Applied Sciences, Harvard University. E-mail: pfister@seas.harvard.edu.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

ing such a comprehensive set of information in the first place. These forgone benefits include the ability to compare the activity of multiple process executions over time or the ability to see dependencies linking the cause of a system fault outside the expected region of error. Having an effective, scalable visualization for provenance data is crucial part of the filesystem's effectiveness as an aid for data analysis, system understanding, and knowledge discovery.

In collaboration with the PASS (Provenance-Aware Storage System) [1] group at Harvard University, we set out to develop a new visualization tool to enable easy and effective exploration of filesystem provenance data. Through a qualitative study with provenance domain experts, we put together a set of tasks to address their visualization needs and gain a better understanding of their current visualization practices. Through a task-driven iterative design process we developed a novel filesystem provenance visualization called InProv that utilizes a radial layout (Figure 1, *middle* & *right*). The tool also incorporates our new time-based hierarchical node grouping method. This new method was inspired by feedback from our qualitative user study. The method more closely matches the user's mental model of node creation and evolution, and enables more intuitive data exploration. InProv displays a filesystem provenance graph in a visual format conducive to exploration in addition to focused querying. The current design and implementation of InProv has been tested on graphs of up to 60,000 nodes.

To evaluate the effectiveness of InProv with its radial layout compared to Orbiter [40], a conventional node-link diagram (Fig. 1, *left*), we designed and performed a quantitative user study. The study also compared the effectiveness of our new time-based hierarchical node grouping method to a conventional method. The user study was a mixed between and within-subject user study and evaluated each tool with several real world example data sets. Domain experts knowledgable in the topics of our sample data were recruited to participate in the study. The results of the study demonstrate that the new time-based hierarchical node grouping method is more effective for analyzing data in both tools, and that InProv is more accurate and efficient than Orbiter for analyzing large complex data.

The first contribution of this paper is a set of requirements for filesystem provenance data analysis based on our interviews with domain experts. Our second contribution is InProv, a new radial layout visualization tool for browsing filesystem provenance data. Our third contribution, developed to make InProv more effective by identifying the most important nodes and processes in a system, is a new time-based hierarchical node grouping method for provenance data. Our final contribution is the results of our quantitative user study. We present statistically significant results that people are more accurate and efficient using our new time-based node grouping method, and that the radial based visualization tool, InProv, is more accurate and efficient than Orbiter at analyzing large complex data. Subjectively participants found InProv to be easier to use and preferable to Orbiter. Our user study results also demonstrates one of the first examples of gender differences in visualization tool performance.

2 Related Work

Provenance Data Visualization The conventional visual encodings for provenance data are derived from the fields of network and graph visualization. Having effective visualizations of provenance data is necessary for a person to understand and evaluate the data [35]. The most common visualization strategy for provenance data is the node-link diagram and is employed by common provenance tools such as Haystack [28], Probe-It [17], and Orbiter [40]. With this visual encoding, nodes are represented as glyphs and edges or connections between nodes are represented as lines or curves. These tools utilize a variety of different visual encoding techniques including directed node-link diagrams [17, 28] and collapsible summary nodes [40]. A specific application area for provenance are workflows, such as visualization [30] and scientific workflows (e.g., tracking where data sets originated and how they have been manipulated). Visualizations for scientific workflows are also focused on node-link diagrams and include such tools as VisTrails [7, 14, 46] and ZOOM UserViews [12]. Unfortunately, these node-link visualization strategies are difficult to scale to provenance data sets beyond a few hundred nodes. Traditional node-link diagrams can easily become too visually cluttered for the multi-thousand node filesystem provenance data limiting a user's ability to thoroughly analyze and explore the data. In our tool, we employ an alternative radial layout with hierarchical encoding with an easily navigable time dimension to reduce visual clutter and bring the most important nodes to the forefront.

Network & Tree Diagrams There has been extensive work in the network visualization community on effective techniques for generating and drawing large complex networks [1, 3, 4, 5, 21, 27, 44, 54]. There has also been work on the effective display of networks that change over time, usually employing animation [19, 34]. Most provenance data have hierarchical properties or attributes. Thus, we found visual encoding techniques from the tree visualization community to be useful points of reference [47]. For example, TreePlus is an example of a tree-inspired graph visualization tool that prioritizes node readability and layout stability [37]. The visual interface displays a tree, starting from the graph root or a user-specified starting node. This technique is more effective than a traditional node-link diagram for exploring subgraphs and providing "local overviews," but fails to provide a high-level overview of the relationships in the overall graph. Another tree-inspired visualization tool is TreeNetViz, which displays tree-structured network data using a radial, space-filling layout with edge bundling [23]. For large complex provenance data sets, the strategies employed by TreeNetViz, in which sectors expand, will become visually complex and is not necessarily an efficient use of screen real estate. In our work we employ a similar radial layout to TreeNetViz in which our tool expands sectors, but they expand into a full new radial plot to maximize label readability and take advantage of available screen space.

Radial Plots Radial or circular layouts bring visual focus to the relationships between nodes rather than the relative spatial locations of nodes. One of the earliest examples of radial layout visualization was proposed by Salton et al. [45] for visualizing text data. Since then, many successful visualization tools using this radial layout have been produced to visualize everything from file systems to social network data to genomics data [16, 18, 20, 29, 33, 38, 41, 51]. Spatial encoding can reflect useful attributes for smaller graphs [10, 39], because the human eye is acutely attuned to deciphering 2D spatial positions. We employ a radial plot layout to reduce visual clutter and easily show connections and nodes relevant to our user base. Processes and unique activity are accentuated while system libraries and ubiquitous workflows such as system boot-up are minimized.

In the following sections we present a more detailed background on provenance and related terminology, discuss the domain specific set of tasks that motivated the design of InProv, and present the design and implementation of InProv. We then describe a new time-based hierarchical grouping method for provenance data developed for InProv. Finally, we present the results of our quantitative user study to evaluate the performance of InProv relative to Orbiter [40], a conventional node-link graph visualization tool. We conclude by discussing the results presented in the paper and highlighting areas of future work.

3 Provenance Data

We focus on filesystem provenance data (i.e., the relationship between files and processes and their interactions). Filesystem provenance data are inherently an annotated directed acyclic graph. We tested InProv on output from PASS, a "provenance-aware storage system" created by the Systems Research Group at Harvard (SYRAH) [2]. **Nodes** may be processes (an instance of an execution of a program that may read from and/or write to files or pass data or signals to other processes), files (static representations of data), pipes (communication channels

[1]http://www.eecs.harvard.edu/syrah/pass/

[2]http://www.eecs.harvard.edu/ syrah/

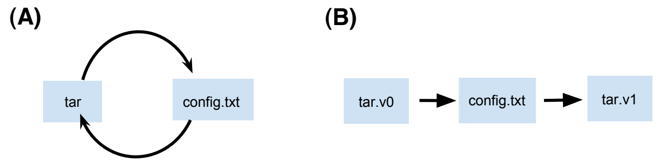

Fig. 2. (A) There exists a cycle between tar and config.txt. (B) By versioning the tar process, PASS ensures that the graph remains acyclic.

between processes), non-provenance files (files whose actions are not recorded), or "other" (filetypes unrecognized by the PASS system). **Edges** represent the dependency relationships between the nodes. For example, edges could represent "a process writes to a file", "a process reads from a file", "a process spawns another process", or "a user controls a process."

Each node may have a variety of attributes such as node name, filesystem path, and process node ID. This information is important to investigate specific processes or gain a deeper understanding of what is occurring in the system. Nodes also have an indegree and an outdegree. Indegree and outdegree refer to the number of edges that lead into or out of a given node.

To ensure the resulting provenance graph is acyclic, the PASS system uses a cycle reduction algorithm that assigns a version number to each node. (Fig. 2). The PASS system records a **timestamp** called "freezetime" as an attribute of when each versioned node is created. It also records the "exectime" when a process executes.

4 Requirements Analysis

We conducted an informal qualitative formative user study with provenance researchers who have been developing and using provenance capture systems for over five years. Our goal was to identify the domain specific analytic tasks that an effective visualization should address. We conducted semi-structured interviews with seven provenance data experts, all of whom work with filesystem provenance data, to learn about their data analysis and exploration tasks, current visualization solutions they use, and the limitations of their existing visualizations and workflow. The interviews lasted approximately one hour. The interviews were contextual and, in addition to answering the interviewer's questions, the interviewee demonstrated the workflow and analysis tools they were currently using. Each interviewee was asked questions relating to their current area of research, the analysis tools they used, the data formats with which they interacted, and the analysis tasks they performed.

We used affinity diagraming [11] to analyze the data from the interviews to identify common domain tasks. Despite the range of task requirements, a common theme emerged: while researchers could effectively analyze small subsets of a provenance graph, understanding the system as a whole usually required line-by-line analysis of the original (raw) data. The lack of an effective way to visualize large graphs prevented researchers from extracting an informative whole-graph analysis. We thus concluded that the ability to provide a quick summary of the overall unique system activity was a key priority. Other task requirements closely echo many canonical information visualization data exploration tasks [48].

InProv was designed to handle the following domain tasks (with analytic tasks, using definitions from Amar et al. [2], in parentheses):

1. Summarize system activity — Hierarchically group provenance graph by time of system activity (Cluster, Find Anomalies). A researcher frequently needs to analyze a provenance data set generated by someone else or a personal data set that was generated long ago. Understand such data sets requires that user quickly obtain a high-level overview or summary of the activity represented by the data set. A good visualization should highlight the main events that occurred during the recording of the data.

2. View filtered subset of system data — Display selected provenance subgraph (Filter). Users also frequently need to more deeply analyze a subset of a data set. For example, after obtaining a high level overview as in Task 1, a user will frequently identify one or more high level tasks that warrant more detailed analysis. Alternately, a user analyzing a current trace might already have identified objects or processes of interest and may want to view the subset of the data set pertaining to them. These are both domain-specific instances of the more general "zoom and filter" operations. An effective visualization should allow the user to naturally select a subset of nodes, either manually (e.g., by clicking) or formally specified (e.g., using a query or filter). Although interested in only a subset of the data, most users want to view and understand these subsets in the context of the entire data set. In other words, when examining some subset of nodes in a provenance graph, the user should see the selected subset of nodes in the context of the whole graph.

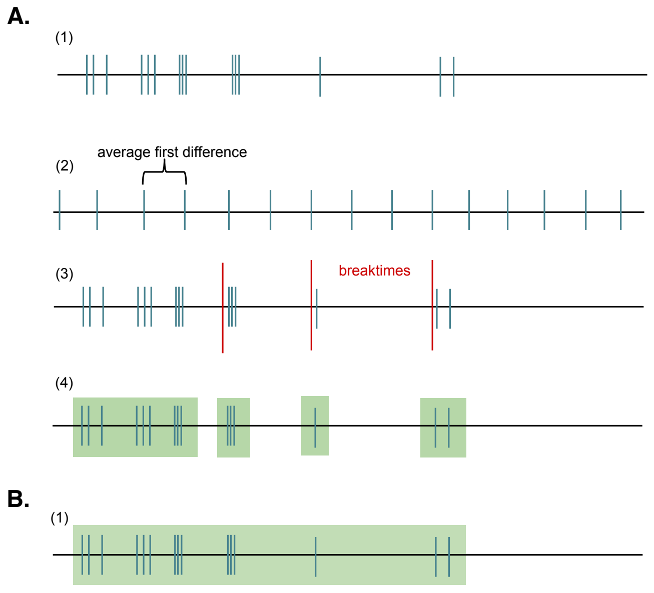

Fig. 3. **A.** Time-based hierarchical grouping sorts the provenance graph according to time attributes of nodes. (1) Most system activity is distributed unevenly on a timeline. (2) Our algorithm computes the "average first difference" of timestamps, i.e., the difference between timestamps if they followed a perfectly even distribution. (3) Gaps in activity of above-average duration are marked as "breaktimes," or borders between clusters. (4) These breaktimes bookend each time-based cluster. **B.** Conventional methods group all the nodes across time into a single group based on process ID.

3. View node attribute details — Display attribute value (Retrieve Value). Each node in a provenance graph typically has a variety of attributes (e.g., date created, date modified, number of dependencies, etc.). Users often wish to analyze how these metrics vary across and reflect the structure of the graph. Important metrics should be visually encoded or at least displayed in a node detail view.

4. Examine object history — Display provenance subgraph within one edge of queried node (Filter). The most common provenance query is the lineage query, whose response explains how an object came to be in its present state. These lineages can be quite large, depending on how long the system has been running and/or how deep in a derivation tree the object appears. Thus, a visualization should offer a node-specific view with information on how that node was created and modified over time. This task is equivalent to a query asking for information on the ancestors of a particular node.

5 Time-based Hierarchical Grouping

Due to the size, scale, and varying levels of granularity of provenance data, a hierarchical grouping of the nodes in the provenance graph is necessary to ensure users can comprehend a typical data set.

We initially chose Markov Chain Clustering (MCL) [53] to cluster the provenance graph. The algorithm runs by simulating a random walk on the graph. Since nodes in the same cluster have a high probability of being connected, and two nodes in different clusters have a low probability of having an edge between them, a random path beginning in one cluster has a high probability of remaining in that cluster. If a cluster is particularly large, contained nodes were divided hierarchically into subgroups by file path because files within the same folder tend to be associated with similar workflows.

However, our initial attempts to use MCL proved ineffective. The structure of the created summary nodes did not properly communicate what was going on in the system, and the visualization's users struggled to find a way to describe the contained activity. Tellingly, one of the expert users did not even recognize that the data displayed was one of his/her own provenance data files. Furthermore, users noticed that, regardless of the data they examined, the details they could see pertained to system boot-up. This ubiquitous system boot-up activity was not pertinent to their investigations and tasks.

To have the node grouping more closely reflect the mental model of the users, we developed a time-based hierarchical grouping method that revolves around the temporal attributes of the provenance data. Through our discussions with experts in our qualitative formative user study, it became evident that understanding the filesystem provenance data was easier in many cases with a temporal context as compared to other grouping methodologies. For example, with a temporal context, a researcher can follow the exact steps a computer user took to preform a specific tasks or execute a series of programs; this provides the researcher with additional insight as to the purpose of each action.

Each job or execution in a computer system produces a burst of system activity and the recording of multiple "freezetime" and "exectime" timestamps (Sec. 3). These bursts of activity are usually separated by longer periods of relative inactivity. Thus, grouping together provenance nodes with roughly simultaneous timestamps allows for a hierarchical subdivision of system activity at varying levels of granularity. Hadlak et. al similarly use time attributes of data to visualize hierarchies [24]. The summarizations created by our algorithm map to the summaries of system activity provided by provenance experts (Task 1, Sec. 4). Feedback from users indicated this clustering approach more closely matches the users' mental models of the organization of the data (i.e., processes relevant or related to each other in a temporal context are visually near each other). This was the motivation for one of our main hypotheses in our quantitative user study (H4, Sec. 7).

The method we developed works as follows (Fig. 3): First, all the timestamps in a given set of nodes and edges are sorted chronologically. Next the "average first difference," i.e., the total duration of activity in the data set divided by the total number of timestamps, is computed. Then the timestamps are scanned in order and the first difference (the previous timestamp subtracted from the current timestamp) between each is computed. Whenever the first difference is above a threshold, i.e., there is a significantly long gap in recorded activity (default being twice the average first difference based on expert input and pilot testing of different thresholds), that time is recorded as a break between node groups. Nodes with activity occurring between two subsequent break times are defined as new groups.

The algorithm tries to produce between five and sixty groups, with each group limited to fifty nodes. Based on our formative study, these heuristics marked the observed limits of a user's ability to comprehend and to explore a data set. If a group has more than fifty nodes, the algorithm will attempt to divide it hierarchically into subgroups of nodes so that the user is not overwhelmed by the display of too many nodes. This hierarchical subgrouping of nodes based on time is beneficial to both "bushy" and "deep" provenance trees. Bushy trees result from widely used tools (i.e., compiler has lots of descendants) and deep trees result from continued data derivation (i.e., extract items, analyze them, re-do analysis and repeat). In both cases subdividing and grouping by temporal information will usually broaden deep trees and summarize bushy trees for easier comprehension.

One of the limitations of the current implementation is that during dense periods of activity an excessive number of nodes will be grouped at one particular time step. The other limitation is that certain patterns of user activity are sometimes not optimally split. For example, a script that compiles a tool and then immediately runs a workload that uses it. A user would expect that the compile would be in one group and the workload in another. However the workload may instead be split so that one group represents the compile plus the beginning of the workload, and the other cluster has the rest of the workload.

We plan to implement in future work "smarter" breaks in system activity (e.g., [6]). It should also be noted that this grouping method collapses versions resulting in a non-directed acyclic graph. This does not conflict with the tasks discussed in Sec. 4, but needs to be examined in future work if ordering is important to the task at hand.

6 INPROV BROWSER

Based on our formative interviews and task-driven iterative design process with domain experts, we developed a new provenance data browser called InProv (Figs. 1 & 4). Motivated by Task 1 (Sec. 4), the need to have an effective high-level overview of the system, we adopted a hierarchical radial layout for the visual display of the provenance node graph as this provides focus on the overall structure of the graph and makes it easy to read the edges connecting nodes. We will show the utility for specific features of the layout in the remainder of this section. Also motivated by Task 1 (Sec. 4), the default node grouping method for the provenance graph in InProv is our new time-based hierarchical method (Sec. 5). The timeline at the bottom of the screen provides temporal context for each group. Each of these groups of nodes is displayed in the center of the screen as a ring divided into multiple sectors. Each sector in a ring is either a single node or a subgroup of nodes, visually encoded as a thicker sector (e.g., Fig. 1, *middle*), which can be expanded into a ring of its own (Task 2, Sec. 4). A text path at the top of the screen, as well as the "context view" rings on the right of the screen, provides context on the sequence of node or node group expansions. InProv was implemented using Java and Processing. We plan to make it open-source available.

Nodes: Nodes, visually encoded as sectors in a ring, are colored according to their type: processes are dark grey, files are white, and all other files (including non-provenance files and node groups) are grey. Subgroups of nodes are represented as thicker sectors than individual nodes (e.g., Fig. 1, *middle*). The width of a node subgroup sector in radians is proportional to the number of nodes it contains (e.g., Fig. 4, "bash" contains more nodes than "sshd" thus it covers a larger fraction of the radial plot). Nodes are drawn clockwise around the ring in order of increasing Provenance Node ID, or PNODE (analogous to the INODE of a file). InProv originally did not have a deterministic algorithm to order sectors. This was confusing to users because the same ring could look different upon multiple viewings. PNODE was chosen as an ordering index because PNODEs are assigned by the PASS system in monotonically increasing order, thus a PNODE number is an effective heuristic for creation date. This enables a "clock" metaphor, where a user can read the procession of nodes around the circle as the progression in time of node creation. To adapt InProv to display provenance information of a different format, PNODE could be replaced with any other ordinal metric, such as creation time or last modification time. This representation of ordered nodes, or groups of nodes, provides a compact easy to see representation of the system activity (Task 1, Sec. 4).

Edges: Edges, visually encoded as lines, are drawn in the center of the ring in the direction of data flow (i.e., from parent nodes to their children). As compared to other visual encodings, such as node-link diagrams, the radial layout's edges are clean and easy to read with minimal visual clutter (Task 1, Sec. 4). While canonical provenance direction flows from children to parents, following an object's history up through the chain of ancestry, this directionality was found to be counter-intuitive by participants in our formative qualitative study (Sec. 4), thus InProv draws edges from parent to child nodes. Edges are also drawn for subgroups of nodes. If subgroups A and B are sectors in the same ring, and a node in group A has an edge to a node in group B, an edge will be drawn from sector A to sector B (e.g., Fig. 4, at least one node in the "uname" group has an edge to a node in the "bash" group, but no nodes in the "uname" group are connected to any nodes within "sshd"). For more detail about the edges to and from

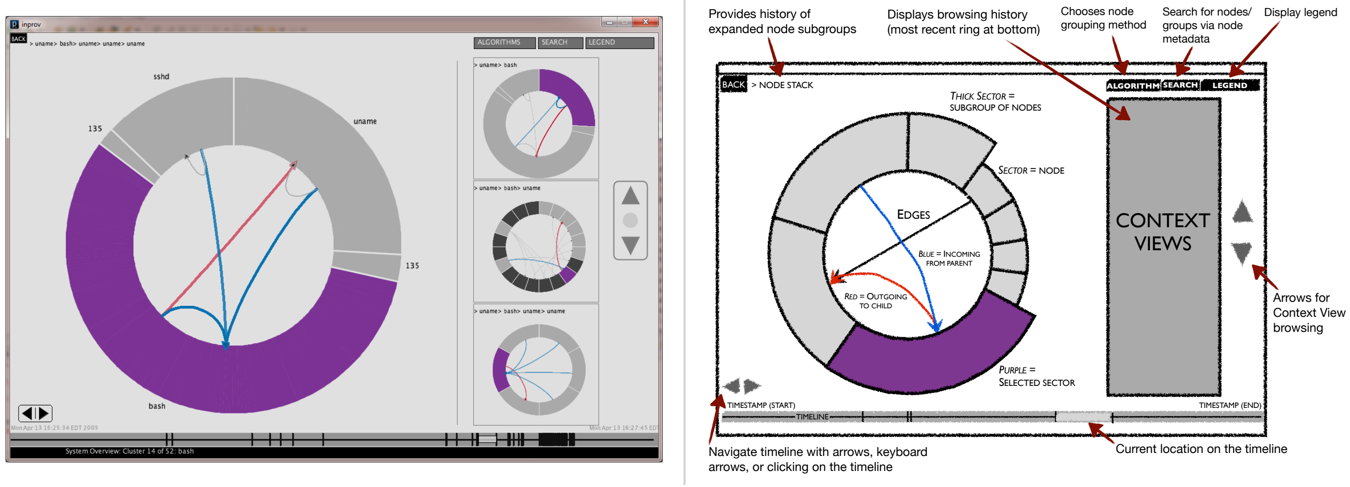
inprov
BACK
> uname> bash> uname> uname> uname
ALGORITHMS
SEARCH
LEGEND
sshd
uname
135
135
bash
> uname> bash
> uname> bash> uname
> uname> bash> uname> uname
System Overview: Cluster 14 of 52: bash
Provides history of expanded node subgroups
Displays browsing history (most recent ring at bottom)
Chooses node grouping method
Search for nodes/ groups via node metadata
Display legend
BACK > NODE STACK
ALGORITHM
SEARCH
LEGEND
THICK SECTOR = SUBGROUP OF NODES
SECTOR = NODE
EDGES
BLUE = INCOMING FROM PARENT
RED = OUTGOING TO CHILD
PURPLE = SELECTED SECTOR
CONTEXT VIEWS
Arrows for Context View browsing
TIMESTAMP (START)
TIMESTAMP (END)
TIMELINE
Navigate timeline with arrows, keyboard arrows, or clicking on the timeline
Current location on the timeline

Fig. 4. **Left:** Screenshot of InProv showing the interactions of the node "bash" with its parent and child nodes. The blue edges represent incoming edges from parent nodes, and the red edges represent outgoing edges to child nodes. **Right:** Schematic drawing displaying the key visual encodings and interaction features for InProv. The "node stack" and "context views" both provide context of browsing history as well as location within the hierarchical structure.

particular sectors, a user can click and select those sectors. The incoming and outgoing edges will be highlighted with bright colors so that they visually pop from the other edges in the ring. Incoming edges, from parents, are colored blue (e.g., from "sshd" to "bash" in Fig 4), while outgoing edges, to children, are colored red (e.g., from "bash" to "uname" in Fig 4). We initially drew the edges as thin solid lines. We changed the design to arrows because edge directionality was important to users. The opacity of edges between sectors indicates how many edges there are between the two sectors. Stronger connections are more opaque and more visible. This draws the user's eye to more active connections (Task 4, Sec. 4).

The visualization does not distinguish between control dependency (exchanged signals), data dependency (exchanged data), or version edges (connecting different instances of the same node). The provenance researchers we interviewed explained that they did not need to distinguish these edge types for any of their primary tasks (Sec. 4). Since this visualization was designed to give a high level overview of a provenance data set without overwhelming the user, this design choice is reasonable.

Timeline: Each ring represents a group of system activity that happened around the same time. However, users need to be able to examine the evolution of the system over time (Task 4, Sec. 4), thus InProv has the ability to browse data over time. The duration of this activity is shown on the timeline (e.g., bottom of Fig. 4). The dates above the timeline show the earliest and latest timestamps in the data file. From these timestamps, the user can infer the duration of data collection. The duration of the currently viewed cluster is represented on the timeline as a grey rectangle. As the user scrolls left and right through the available clusters by using the left and right arrow keys or clicking the onscreen arrows, the grey rectangle moves along the timeline to update the user on his/her current contextual location. Clicking a sector will highlight its associated timestamps on the timeline as black hashmarks. The timeline partially solves the need for context by showing how the viewed cluster and any selected sectors relate to the overall graph (Task 2, Sec. 4). The timeline is only enabled when the data are grouped with the time-based hierarchical node grouping algorithm.

Algorithms: In addition to our new time-based node grouping method, InProv can also group nodes using a conventional "process tree" node grouping method based on control flow information [40]. This method creates summary nodes by treating processes as primary nodes and constructing a summary node for each primary node. It arranges these summary nodes in a way that reflects the process tree reconstructed from the control flow information found in the provenance metadata (Fig. 3, *B*). Each summary node contains a primary node and all of its immediate ancestral and descendant secondary nodes (non-processes). InProv is able to group the nodes and draw the ring(s) with either algorithm; by hovering over the "Algorithms" button, the user can choose between the time and process tree node grouping methods.

Navigation and Interaction: Hovering the mouse over a sector displays a tool tip with more information about that particular sector. This design feature was motivated by the users' need to investigate more detailed information about a particular node (Task 3, Sec. 4). If the sector is a subgroup of nodes, hovering will display information such as the number of contained sectors, as well as the numbers of contained files and processes. Clicking on a sector selects it, turning it purple, and clicking again on the selected sector expands it. If the sector represents a subgroup of nodes, those nodes will expand to fill a new ring (Task 2, Sec. 4). We investigated expanding sectors in place, as in TreeNetViz, but decided that limiting the total number of sectors displayed to the user at any given time for comprehensibility was a greater priority [23]. If the sector is a single node, the new ring will display all nodes one edge away from the current node regardless of what timestamp they were in originally. The user can thus see what connections a node has outside of the group it which it was initially displayed in (Task 4, Sec. 4).

Node Stack: Each time a sector is expanded, its name is added to a list of expanded node groups, or nodes, displayed at the top of the screen as a text path. Next to the "node stack" text path is a "BACK" button for returning to the previous ring (e.g., top of Fig. 4). This list of sectors communicates the path the user took to get to the current view. We added this feature in response to user feedback. During qualitative feedback sessions with an early version of InProv, users repeatedly complained that, upon expanding a node, they were confused as to how they had ended up in their new location and were unclear on the current view's location in the overall graph. The addition of the node stack greatly helped the users to keep contex and understand the hierarchical structure as node subgroups were expanded (Tasks 1 & 2, Sec. 4).

Context Views: Each time a sector is expanded, a miniature version of its previous ring and its node stack path are added to the "context view" displayed on the right side of the screen. The context view displays three rings at a time. The rings are stored starting from the bottom of the screen, where the most current ring is displayed. The context view scroll, i.e., the up and down arrows to the right of the context view, allows the user to view their navigation history. The sector that was clicked-on for expansion is colored purple in each of the context view rings. This helps the user remember their browsing history as well as give hierarchical context. For example, expanding a series of node subgroups in a ring will show the hierarchical context of the data (Task 2, Sec. 4). When the data is clustered by time, each break in time (as denoted by the hashmarks) has its own context view. Thus, the user's context view is not lost during navigation.

7 Quantitative User Study

We conducted a quantitative user study to evaluate the accuracy and efficiency of InProv compared to Orbiter, a conventional filesystem provenance data visualization tool using node link diagrams. In the same study, we also compared our new time-based hierarchical grouping method (see Sec. 5) to a conventional process ID node grouping method. We implemented both new and conventional node grouping methods into InProv and Orbiter for the user study.

To ensure broad relevance of the results, we included two different types of tasks, two levels of task difficulty, and four different user populations.

7.1 Hypotheses

Our hypotheses entering the user study were:

H.1 Participants will be able to complete tasks more accurately in InProv than Orbiter. The radial layout utilized in InProv more concisely summarizes and presents the information to users compared to the node-link diagram utilized in Orbiter. This simpler representation will enable users to more accurately complete tasks.

H.2 Participants will be able to complete tasks more efficiently in InProv than in Orbiter. Navigation and context viewing in InProv allows users to track their visited paths more easily than in Orbiter. The increased amount of zoom in or out required to explore the node-link diagram in Orbiter will make it more difficult for users to remember their visited paths.

H.3 Participants will subjectively prefer using InProv to Orbiter and find the tool easier to use. Following the reasoning in H1 and H2, users will find InProv overall easier to use for task completion.

H.4 Participants will perform tasks more accurately and more efficiently in both tools when the nodes are grouped according to our new time-based hierarchical node grouping. We hypothesized that the time-based grouping of nodes would be more consistent with the users' mental models of the historical file system activity than the hierarchal dependency grouping, thus users will be more accurate and efficient in both tools when completing tasks with the time-based grouping.

7.2 Participants and Apparatus

Because our use case scenarios focused on both IT professionals and scientific applications, we recruited study participants from these fields. Twenty-seven members of the Harvard community participated in the study (20 men, 7 women; 19–59 years old, M=34). Thirteen participants were professional IT staff. Ten were scientists representing domains covered by our tasks (6 bio/medical and 4 astrophysics computational scientists). The remaining 4 participants were provenance research experts. Participants received monetary compensation for their time.

We required that all participants be familiar with Linux/Unix operating systems as the minimal background knowledge required to participate in the study. We also required that all participants have normal color vision (i.e., are not color blind).

All of the user study sessions were conducted in the same indoor room utilizing the identical Lenovo ThinkPad 15" (1600x900 screen resolution) laptop running Windows Vista with Logitech wireless mouse with scroll wheel. Camtasia Studio 8 was used for screen and audio capture.

7.3 Tasks

We had two types of tasks. The first type was focused on finding an explicit file or process node, and the second type was focused on understanding larger concepts demonstrated by the sample provenance data. This first task type is derived from the second and fourth task requirements in our set of tasks, and the second task type is derived from the first task requirement in our set of tasks (see Sec. 4). The following question is an example of the first task type: "A radiologist is analyzing a patient's medical imaging data. Which process is responsible for aligning and warping the images?". The following question is an example of the second task type: "A user is complaining about their computer acting weird. Looking at the user's provenance data from before the complaint, what was the application the user invoked?". For each task in the study, a data set was loaded into the tool and the participants were asked a question prompting them to complete one of these two types of tasks. Participants were presented with an equal number of both task types during the study. For each task type, we had 5 instances. Out of all 10 instances, 5 of them were easy (42-346 nodes) and 5 were hard (1192-5480 nodes). The boundary between easy and difficult tasks was determined in a pilot experiment in which tasks with 10s, 100s, 1000s, and 10,000s of nodes were compared.

The tasks used real world sample data and the questions were designed to mimic such real world scenarios. The sample questions above are examples of a bio/medical imaging scenario, and an IT scenario, respectively. The wording of the questions relating to our scientific scenarios were derived from the questions asked as part of the First and Third Provenance Challenges [42, 49]. The data sets from these two challenges were used as the domain scientific data in our study. The data are standardized and publicly available [3]. The 1st Provenance Challenge's data is on brain atlases from the fMRI Data Center and the 3rd Provenance Challenge's data set on the Pan-STARRS project. The other IT related questions, as well as the PASS team's sample data from these provenance challenges, are also publicly available online through the PASS Team Website [4]. All participants were presented with the same set of tasks which included tasks from multiple domains.

7.4 Procedure

Each study session started off with a basic demographic survey and a series of multiple choice questions to assess each participant's prior knowledge of Linux/Unix operating systems as well as filesystem provenance. Next, the participants were presented with two pages of background information on filesystem provenance data in order to make sure all participants possessed a basic understanding of provenance. Then the participants received instruction (demonstrated and read from a script by the experimenter) on how to use each of the two visualization tools and received a practice task to perform with each tool. The practice tasks were similar to the tasks given during the main study. The practice data sets also were of varying difficulty (one "easy" and one "hard"), thus representative of the two levels of complexity in data they would see during the study. Finally, the participants moved on to the main part of the study and completed 8 tasks alternating between tools for each task.

For the main part of the experiment, participants were given a series of eight tasks with a specific data set associated with each. All participants completed the same set of mixed-domain tasks with identical associated data, and task orderings were balanced both in the order of tool presentation as well as difficulty level. Participants alternated between the two tools for each task in order to minimize learning effects. The participants also alternated between pairs of "easy" and "hard" data sets. Genders and populations (i.e., astronomer, bio/medical scientist, IT specialist, and provenance expert) were balanced between the two algorithms, between which tool they started with, and between which data difficulty they started with.

The participants were instructed to "talk out loud" while completing the tasks, to verbally state when they had a preliminary guess, and to state what their final answer was. This additional verbalized information was critical to evaluating the participant's performance. The verbalization, applied to a relatively simple task with static data, and was applied equally in all conditions to all participants. The duration of each task was timed from the screen capture from the moment the participant first moved the mouse (after they finished reading the

[3]http://twiki.ipaw.info/bin/view/Challenge/WebHome

[4]http://www.eecs.harvard.edu/syrah/pass/traces/

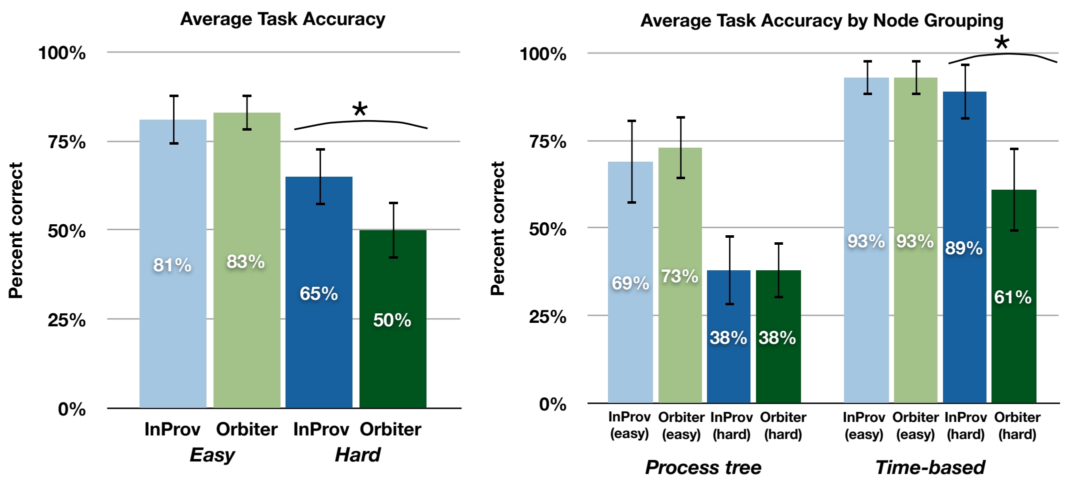

Fig. 5. **Left:** Average accuracies of participants sorted by data difficulty level (easy vs. hard) and tool. Although performance was comparable between tools for easy data, InProv had higher accuracy for hard data. **Right:** Average accuracies of participants sorted by difficulty level, tool, and node grouping method. Error bars correspond to the standard error and the asterisks indicate results of statistical significance.

question) to the statement of their final answer. Except for the practice tasks, users were not given feedback during the session whether their answer was correct or incorrect.

With both tools, the participants were given complete freedom to highlight/select nodes, pan/browse the visual representation, zoom in/out, and expand node groups. The terminology, color encodings and node labels were identical in both tools' UIs. To advance to the next level of the hierarchy in a node subgroup, users double-clicked a thick subgroup sector in InProv while in Orbiter users could either zoom in with the scroll wheel on the mouse or double-click on a "summary node" box. When using Orbiter, users could pan around the node-link diagram by clicking and dragging. (No panning is required with the radial layout of InProv.) When viewing data with the time-based hierarchical grouping algorithm, both tools would display a timeline along the bottom of the screen and a user could either click the left-and-right arrows with a mouse, use the left and right arrow keys on the keyboard, or click/drag the timeline marker to navigate.

The study participants were asked to complete each task in as timely a manner as possible. If the participant was unable to complete the task within 5 minutes, the participant was asked whether he or she had a final answer and was given the post-task questionnaire. Based on a pre-study pilot, it was observed that if a participant was not able to provide an answer within 5 minutes then the participant generally was never able to provide the correct answer.

After each task was completed, the participants were presented with a questionnaire with nine questions to respond to on a 7-point Likert scale. The first six questions were the raw NASA-TLX standard questions for task load evaluation [25, 26], and the remaining three questions gauged subjective ease of use, self-efficacy, and subjective assessment of the tool's effectiveness for the task: "How easy was it to use the tool?", "How confident are you in your answers(s)?", and "How easily were you able to accomplish this task?".

At the end of the session, participants were verbally asked which visualization tool they preferred to use and why, and whether they had any other general comments or feedback. The entire session lasted approximately 60 minutes.

7.5 Experimental Design & Analysis

The study was a 2 x 2 x 2 mixed between- and within-subject design with the following factors and levels:
• Tool (InProv or Orbiter)
• Difficulty (size, complexity) of data (easy or hard)
• Node grouping method (process tree or time-based)

Tool and difficulty were within-subject factors and node grouping method was a between-subject factor. Our dependent measures were number of correctly completed tasks, time to complete a task, and participants' subjective responses recorded on a 7-point Likert scale. Accuracy was a binary measure (i.e., correct or incorrect answer), and the answer keys for each data set were generated by filesystem provenance data experts.

Because many participants waited until the five minute time out to declare their answer, the timing data had a bimodal distribution and we thus used a non-parametric test to analyze them. Also, because normal distributions cannot be assumed for Likert scale responses, we used non-parametric tests to analyze subjective responses as well. For within-subjects comparisons (i.e., to investigate the effects of tool and difficulty) we used the Wilcoxon signed rank test, and for between-subjects comparisons (for investigating the effects of node grouping method) we used the Mann-Whitney U test.

For accuracy, we used a Generalized Linear Model with a binomial distribution. In the model we included the following factors and interactions: tool, data difficulty, node grouping method, tool$\times$difficulty, and tool$\times$node grouping. Additionally, we controlled for effects of population (astronomy, bio/medical, IT, provenance) by including it as an additional factor. Finally, we also included gender and gender$\times$tool as additional factors because our initial analyses revealed possible gender-related differences in performance.

8 User Study Results

8.1 Accuracy

We observed a significant main effect of node grouping method on accuracy with participants being more accurate with the new time-based hierarchical node grouping as compared to the process tree node grouping method ($\chi^2_{(1,N=216)} = 22.74$, $p < 0.001$) as shown in Fig. 5.

Participants were on average more accurate using InProv (M=73%) than using Orbiter (M=67%), but the difference was not statistically significant ($\chi^2_{(1,N=216)} = 2.000$, p > 0.05). As we expected to potentially see a difference in performance between easy and hard data sets, as it has been observed that node-link diagrams are difficult to read if too dense [22], we repeated the analysis separately for the two difficulty levels. While there were no significant effects of tool on performance for easy data sets ($\chi^2_{(1,N=108)} = 0.861$, p $= 0.354$), on hard data sets participants were significantly more accurate with InProv than with Orbiter ($\chi^2_{(1,N=108)} = 7.787$, p $= 0.005$). These results are illustrated in Figure 5.

8.2 Efficiency

As shown in Fig. 6, there was a main effect of node grouping method on average completion time (U = 30, p = 0.003, r = -0.570). With both

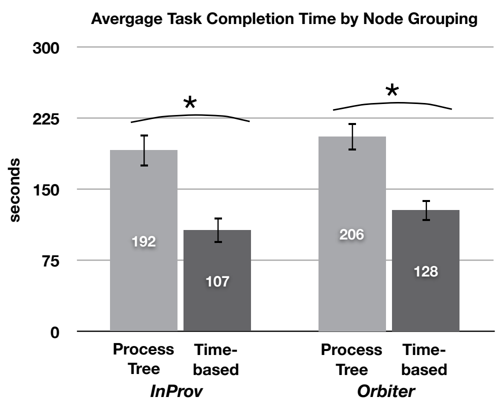

Fig. 6. Average task completion time for each tool broken down by node grouping method. Participants were more efficient using the time-based method. Error bars correspond to the standard error and the asterisks indicate results of statistical significance.

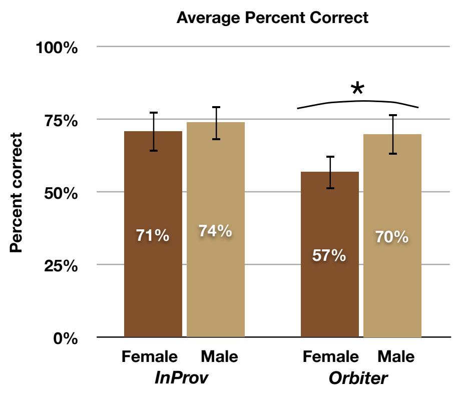

Fig. 8. Average accuracy for each tool broken down by gender. Although men and women had comparable preformance with InProv, men were significantly more accurate than women when using Orbiter. Error bars correspond to the standard error and the asterisks indicate results of statistical significance.

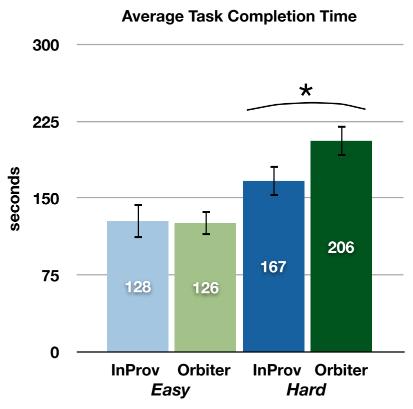

Fig. 7. Average task completion time for easy and hard data sorted by tool. Participants in the study took longer to complete hard data tasks with Orbiter. Error bars correspond to the standard error and the asterisk indicates results of statistical significance.

tools, participants were almost twice as efficient with the time-based node grouping method as compared to the process tree method.

Participants were more efficient on average with InProv than with Orbiter, but the difference was not significant when both data set difficulty levels were considered together ($z = -1.201$, $p > 0.05$). Breaking down the analysis by difficulty, there was no observed effect of tool with easy data but there was a statistically significant effect with hard data ($z = -2.057$, $p = 0.040$). As shown in Fig. 7, participants were more efficient with an average task completion time for hard data of 167 seconds with InProv compared to 206 seconds with Orbiter.

8.3 Subjective Responses

We observed statistically significant effects of tool and node grouping method on participants' responses to certain subjective questions as shown in Table 1. As discussed in Sec. 7.4, the participants rated their answers on a 7-point Likert scale with questions 1-6 being raw NASA-TLX measures. These measures positively reflect upon InProv with participants stating it required less mental activity (Q1), less physical activity (Q2), required less work (Q5), and was less stressful (Q6).

A statistically significant effect of node grouping method was also evident in the subjective data. These measures positively reflect upon the new time-based node grouping method with participants stating they felt less time pressure (Q3), were more successful performing their task (Q4), were more confident (Q8), and found it easier to accomplish their task (Q9). The one measure which favored the process tree node grouping is that participants stated it required less physical activity (Q2). These statistically significant results for node grouping method also have strong effect sizes (Table 1).

Finally, as part of the qualitative feedback solicited at the end of the study sessions, participants were asked which tool they preferred using. Participants overall preferred InProv (56%) to Orbiter (41%), with one participant stating that he/she preferred "neither." The reasons most commonly cited by those who preferred InProv include that it was "easier to navigate", "easier to see the data", and "looks nicer". Those who preferred Orbiter most commonly cited that it "has a data representation I am used to seeing" and that it is "easier to understand". Of those who preferred InProv, 67% used the time-based node grouping method in the study. In contrast, of those who preferred Oribter 64% used the process tree node grouping method in the study.

8.4 Additional Analyses

We observed a significant interaction effect between tool and gender on accuracy for both easy ($\chi^2_{(1,N=108)} = 4.275$, $p = 0.039$) and hard ($\chi^2_{(1,N=108)} = 6.672$, $p = 0.010$) data. As shown in Fig. 8, although men and women performed similarly with InProv with 74% and 71% average accuracies, respectively, men were significantly more accurate (M=70%) than women (M=57%) when using Orbiter. The Cohen's effect size value ($d = 0.36$) suggests a slight to moderate practical significance for the difference in accuracy when using Orbiter.

9 Discussion

The results of our user study demonstrate that the time-based node grouping method resulted in significantly faster and more accurate performance with both tools than the conventional process tree method. These results provide support for our fourth hypothesis. The method's ability to pull the most relevant processes to the top of the hierarchy, and present the system boot-up processes only in the first time step or two, made a large difference in participants' performance. Through the qualitative feedback session at the end of each study, participants commented on how easy it was to use the tools with this grouping since the time element helped them understand the data and system activity. The time metaphor was more intuitive to interpret the events captured in the provenance data compared to the process tree node grouping. With the time metaphor, participants were able to easily reconstruct the original user's actions.

These strong results based on the node grouping methodology are evidence of how important it is to pick a node grouping method for network and graph based data that both presents the most relevant data to the user as well as matches the user's mental model. Regardless of visual encoding, node grouping will affect how a viewer sees, reasons through, and interprets the data presented to them in a visualization.

Table 1. Average subjective data responses to the raw NASA-TLX and subjective questions. The answers were rated on a 7-point Likert scale. Asterisks indicate results of statistical significance, and "d" is the Cohen's d effect size.

#	Question	*Tool*				*Node grouping method*			
		InProv	Orbiter	Sig.?	d	Process tree	Time-based	Sig.?	d
1	How much mental and perceptual activity was required? Was the task easy or demanding, simple or complex? *(1 = low, 7 = high)*	**3.2**	3.6	*	0.30	3.5	3.3		
2	How much physical activity was required? Was the task easy or demanding, slack or strenuous? *(1 = low, 7 = high)*	**2.0**	2.3	*	0.18	**1.7**	2.6	*	0.62
3	How much time pressure did you feel due to the pace at which the tasks or task elements occurred? Was the pace slow or rapid? *(1 = low, 7 = high)*	3.2	3.5			3.9	**2.8**	*	0.64
4	How successful were you in performing the task? How satisfied were you with your performance? *(1 = low, 7 = high)*	4.8	4.4			3.8	**5.3**	*	0.94
5	How hard did you have to work (mentally and physically) to accomplish your level of performance? *(1 = easy, 7 = hard)*	**3.1**	3.5	*	0.25	3.4	3.2		
6	How irritated, stressed, and annoyed versus content, relaxed, and complacent did you feel during the task? *(1 = relaxed, 7 = stressed)*	**2.9**	3.3	*	0.20	3.4	2.8		
7	How easy was it to use the tool? *(1 = easy, 7 = hard)*	3.4	3.7			3.4	3.6		
8	How confident are you in your answers(s)? *(1 = low, 7 = high)*	4.6	4.4			3.7	**5.3**	*	0.93
9	How easily were you able to accomplish this task? *(1 = easy, 7 = hard)*	3.7	4.0			4.4	**3.3**	*	0.76

Our first and second hypotheses were partially supported. Although InProv did not significantly improve accuracy or efficiency for all data set difficulty levels, InProv did prove to be more accurate and more efficient when dealing with large (i.e., $>$1,000 nodes) data sets compared to Orbiter. Thus, as the data increase in size and complexity, InProv with its radial layout is able to maintain higher accuracy levels than Orbiter with its node link diagram (Fig. 5). In other words, task completion on large complex data sets was more accurate with the radial layout and significantly more accurate than Orbiter when utilizing the time-based node grouping method.

Examining the cases in which participants gave incorrect answers, the most common reason was because they ran out of time. Our pilot study results showed that if a participant was unable to provide an answer by the 5 minute mark, then the participant was never able to provide a correct answer. This trend was primarily seen with participants who used the process tree node grouping algorithm. The other common reason for incorrect responses was that participants would get lost in the hierarchy. This was, again, more frequently observed with the process tree node grouping method since the hierarchy was so deep compared to the time-based node grouping. Also, a common problem in Orbiter, was that participants would get lost browsing the large node-link diagram and forget where certain nodes or node clusters were located.

The results of our subjective data analysis support our third hypothesis: participants overall preferred InProv over Orbiter. The subjective ratings reveal that participants overall found InProv easier to use, requiring less work, and with the hard data tasks they felt more successful and more confident in their answers. However, there was a slight trend with node grouping method in which those who preferred InProv had the time-based node grouping method whereas those who preferred Orbiter had the process tree node grouping method. Also, those who preferred Orbiter commonly stated their preference was due to the fact that they were familiar with node-link diagrams, even if they had higher accuracy with InProv.

We also observed an unexpected effect in our study: a statistically significant interaction effect between gender and tool on accuracy. Although men and women had comparable performance with InProv, women performed worse using Orbiter with a much lower accuracy rate. However, the distribution of overall preferred tool by women matched the same response distribution as men, and women gave no verbal feedback that was significantly different from men. Women also had the same distribution of age, educational background, area of expertise, expertise with Linux, and previous knowledge of provenance data as the men in our study. Because our participants were all highly trained professionals in their respective fields, the results are unlikely to be due to differences in education or general cognitive ability. Other than gender, we are not able to find another factor that could have affected the women's performance.

We believe our results are possible evidence of gender specific differences in software design. There are known low level differences between genders that have been observed in psychology lab studies such as differences in spatial reasoning (e.g., [32, 36]). However, the question remains whether these low level gender differences can translate up to higher level tasks or interactions such as problem solving skills and, more specifically, computer visualization software. To date gender differences have been observed, for example, in confidence levels using computer software [8, 13], problem solving strategies [9], behaviors and interaction techniques with software [9, 8, 52], and hardware interfaces [15, 50].

Our observation of women having poor performance with Orbiter, within the context of this previous work, is one of the first examples of measured gender differences in visualization. Given the statistically significant systematic differences presented in our study, but with a small to moderate effect size, it seems worthwhile for future research to look deeper at potential gender-related differences in visualization and related interfaces. This could help to identify best practices in designing visualizations and interfaces for all users.

10 Conclusions

We are continuing to develop InProv for filesystem provenance data exploration. Based on feedback from the quantitative evaluation, we plan to add functionality to load and view multiple files at the same time to support data set comparison. Incorporation of graph difference algorithms will help with this multi-file comparison, and incorporation of the ability to connect directly to provenance databases will enable comparisons and faster data exploration. We also plan to investigate additional or alternative methods for grouping nodes to enable more efficient or different analyses of the data such as fingerprinting-based pattern matching, manual classifications by the user, and machine learning techniques based on user-classified data sets. Finally, we plan to scale both InProv as well as the time-based node grouping method so that they will be able to handle data sets containing hundreds of thousands of nodes.

The results of the quantitative evaluation imply that radial layouts, with the right node grouping method, can be an effective visual encoding for provenance data. The visual encoding we developed in InProv may also be applicable to other types of provenance data and to network data in general. We hope that providing a tool that offers an intuitive summary of provenance data sets will help researchers and developers utilizing provenance enhanced systems, especially those dealing with large data sets. We also hope that the availability of a better tool for understanding provenance will promote the adoption of provenance recording systems and encourage more research in the provenance field.

Acknowledgments

The authors wish to thank Uri Braun and the entire PASS team at Harvard University for their dedicated time and support. M. Borkin was supported by the Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program, and C. Yeh was supported by the Harvard College Research Program (HCRP).

References

[1] J. Abello, F. van Ham, and N. Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE TVCG*, 12(5):669–676, Sept. 2006.

[2] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 15–, Washington, DC, USA, 2005. IEEE Computer Society.

[3] D. Archambault, S. Member, T. Munzner, and D. Auber. Grouseflocks: Steerable exploration of graph hierarchy space. *IEEE Trans. on Visualization and Computer Graphics*, 2008.

[4] D. Archambault, T. Munzner, and D. Auber. Grouse: Feature-based, steerable graph hierarchy exploration. In K. Museth, T. Mller, and A. Ynnerman, editors, *EuroVis*, pages 67–74. Eurographics Association, 2007.

[5] D. Archambault, T. Munzner, and D. Auber. Tuggraph: Path-preserving hierarchies for browsing proximity and paths in graphs. In *IEEE PacificVis '09*, pages 113 –120, april 2009.

[6] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[7] L. Bavoil, S. P. Callahan, P. J. Crossno, J. Freire, and H. T. Vo. Vistrails: Enabling interactive multiple-view visualizations. In *IEEE Visualization 2005*, pages 135–142, 2005.

[8] L. Beckwith, M. Burnett, S. Wiedenbeck, C. Cook, S. Sorte, and M. Hastings. Effectiveness of end-user debugging software features: Are there gender issues? In *Proceedings of SIGCHI*, pages 869–878. ACM, 2005.

[9] L. Beckwith, C. Kissinger, M. Burnett, S. Wiedenbeck, J. Lawrance, A. Blackwell, and C. Cook. Tinkering and gender in end-user programmers' debugging. In *Proceedings of SIGCHI*, pages 231–240. ACM, 2006.

[10] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

[11] H. Beyer and K. Holtzblatt. Contextual design. *interactions*, 6(1):32–42, 1999.

[12] O. Biton, S. Cohen-Boulakia, S. B. Davidson, and C. S. Hara. Querying and managing provenance through user views in scientific workflows. *IEEE ICDE*, pages 1072–1080, 2008.

[13] T. Busch. Gender differences in self-efficacy and attitudes toward computers. *Journal of Edu. Computing Research*, 12(2):147–158, 1995.

[14] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Vistrails: visualization meets data management. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, SIGMOD '06, pages 745–747, New York, NY, USA, 2006. ACM.

[15] M. Czerwinski, D. S. Tan, and G. G. Robertson. Women take a wider view. In *Proceedings of SIGCHI*, pages 195–202. ACM, 2002.

[16] L. Daisy Analysis. Daisy, 2003.

[17] N. Del Rio and P. da Silva. Probe-it! visualization support for provenance. In *Advances in Visual Computing*, volume 4842 of *Lecture Notes in Computer Science*, pages 732–741. Springer Berlin / Heidelberg, 2007.

[18] G. Draper, Y. Livnat, and R. Riesenfeld. A survey of radial methods for information visualization. *IEEE TVCG*, 15(5):759–776, 2009.

[19] Y. Frishman and A. Tal. Online dynamic graph drawing. *IEEE TVCG*, 14(4):727–740, July 2008.

[20] J. Galloway and S. J. Simoff. Network data mining: methods and techniques for discovering deep linkage between attributes. In *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling - Volume 53*, APCCM '06, pages 21–32, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[21] E. R. Gansner, Y. Koren, and S. C. North. Topological fisheye views for visualizing large graphs. *IEEE TVCG*, 11:457–468, 2005.

[22] M. Ghoniem, J. Fekete, and P. Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *IEEE Symposium on Information Visualization*, pages 17–24. IEEE, 2004.

[23] L. Gou and X. L. Zhang. Treenetviz: Revealing patterns of networks over tree structures. *IEEE TVCG*, 17(12), December 2011.

[24] S. Hadlak, C. Tominski, H. Schulz, and H. Schumann. Visualization of hierarchies in space and time. In *Workshop on Geospatial Visual Analytics: Focus on Time at the AGILE International Conference on Geographic Information Science*, 2010.

[25] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 904–908. Sage Publications, 2006.

[26] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human mental workload*, 1(3):139–183, 1988.

[27] I. Herman, G. Melancon, and M. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE TVCG*, 6(1):24 –43, jan-mar 2000.

[28] D. Huynh, D. Karger, and D. Quan. Haystack: A platform for creating, organizing, and visualizing information using rdf. In *Semantic Web Workshop, Hawaii, USA*, 2002.

[29] S. S. IT. Solid source explorer, 2009.

[30] T. J. Jankun-Kelly, K.-L. Ma, and M. Gertz. A model and framework for visualization exploration. *IEEE TVCG*, 13(2):357–369, 2007.

[31] M. Jayapandian, A. Chapman, V. G. Tarcea, C. Yu, A. Elkiss, A. Ianni, B. Liu, A. Nandi, C. Santos, P. Andrews, B. Athey, D. States, and H. V. Jagadish. Michigan molecular interactions (mimi): putting the jigsaw puzzle together. *Nucleic Acids Research*, 35(suppl 1):D566–D571, 2007.

[32] C. M. Jones and S. D. Healy. Differences in cue use and spatial memory in men and women. *Proceedings of the Royal Society B: Biological Sciences*, 273(1598):2241–2247, 2006.

[33] M. I. Krzywinski, J. E. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 2009.

[34] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. *IEEE TVCG*, 12:805–812, 2006.

[35] M. Kunde, H. Bergmeyer, and A. Schreiber. Provenance and annotation of data and processes. In J. Freire, D. Koop, and L. Moreau, editors, *IPAW '08*, chapter Requirements for a Provenance Visualization Component, pages 241–252. Springer-Verlag, Berlin, Heidelberg, 2008.

[36] C. A. Lawton. Gender, spatial abilities, and wayfinding. In *Handbook of gender research in psychology*, pages 317–341. Springer, 2010.

[37] B. Lee, C. Parr, C. Plaisant, B. Bederson, V. Veksler, W. Gray, and C. Kotfila. Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE TVCG*, 12(6):1414 –1426, nov.-dec. 2006.

[38] Y. Livnat, J. Agutter, S. Moon, and S. Foresti. Visual correlation for situational awareness. In *InfoVis*, 2005.

[39] J. D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM TOG*, 5:111–141, 1986.

[40] P. Macko and M. Seltzer. Provenance map orbiter: Interactive exploration of large provenance graphs. In *TaPP*, 2011.

[41] M. Meyer, T. Munzner, and H. Pfister. Mizbee: A multiscale synteny browser. *IEEE TVCG*, 15(6):897 –904, nov.-dec. 2009.

[42] L. Moreau and et al. Special issue: The first provenance challenge. *Concurrency and Computation: Practice and Experience*, 20(5):409–418, 2008.

[43] K.-K. Muniswamy-Reddy, D. A. Holland, U. J. Braun, and M. Seltzer. Provenance-aware storage systems. In *2006 USENIX Annual Technical Conference*, 2006.

[44] A. J. Pretorius and J. J. Van Wijk. Visual inspection of multivariate graphs. *CGF*, 27(3):967–974, 2008.

[45] G. Salton, J. Allan, C. Buckley, and A. Singhai. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science*, 264:1421–1426, June 1994.

[46] C. Scheidegger, D. Koop, E. Santos, H. Vo, S. Callahan, J. Freire, and C. Silva. Tackling the provenance challenge one layer at a time. *Concurrency and Computation: Practice and Experience*, 20(5):473–483, 2008.

[47] H.-J. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, 2011.

[48] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[49] Y. Simmhan, P. Groth, and L. Moreau. Special section: The third provenance challenge on using the open provenance model for interoperability. *Future Generation Computer Systems*, 27(6):737 – 742, 2011.

[50] D. S. Tan, M. Czerwinski, and G. Robertson. Women go with the (optical) flow. In *Proceedings of SIGCHI*, pages 209–215. ACM, 2003.

[51] C. Tominski, J. Abello, and H. Schumann. Axes-based visualizations with radial layouts. In *SAC '04*, 2004.

[52] S. Turkle. Computational reticence: Why women fear the intimate machine. *Technology and womens voices: Keeping in touch*, pages 41–61, 1988.

[53] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[54] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.-D. Fekete, and D. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *CGF*, 30(6):1719–1749, 2011.