Visualization Techniques for Grid Environments: a Survey and Discussion

Lucas Mello Schnorr, Philippe Olivier Alexandre Navaux
Instituto de Informática
Universidade Federal do Rio Grande do Sul
CEP 91501-970 - Porto Alegre - RS - Brazil
E-mail: {lmschnorr,navaux}@inf.ufrgs.br

Guillaume Huard
Laboratoire LIG
ZIRST 51, avenue Jean Kuntzmann
38330 - Montbonnot Saint Martin - France
E-mail: Guillaume.Huard@imag.fr

Abstract

Grid computing is being widely used to execute applications that need sufficiently computational power. The main characteristics of this kind of environment are the heterogeneity, dynamism and large scale. These features influence directly the way the monitoring task is done in Grids. This influence can be easily perceived when there is a need to analyse the big quantity of information that can be collected in a Grid environment. This paper discusses how some of the visualization techniques of information visualization can be applied to visualize data of Grid environments.

1. Introduction

Grid Computing is being widely developed in order to achieve more computational power to computer applications. The main trend in this development is focused in virtual organizations [13], which basically aggregates a functionality of the Grid. Their resources might be spread geographically. Practical examples of Grids are the Grid5000, CoreGrid, TeraGrid and others, which today are used to execute highly distributed applications, in a multi-site environment usually composed by a big number of processors. Common characteristics of Grids are the heterogeneity, dynamism and size. These characteristics are directly related with all tasks made on top of the Grid, such as monitoring, jobs and task management and virtualization.

Dealing with Grids involves many tasks such as application deployment, security configuration and resource management, for example. One of these tasks is the resource and application monitoring. The main objective of monitoring is to provide information that can be useful in the other activities in the Grid. The information monitored can be from the specific details of Grid applications to the network and resource data. The analysis of all this information can be complex, since thousands of resources can be used in an application execution.

One way that tries to facilitate the process of information analysis is the use of visualization techniques. Their advantage is that visual analysis can give the analyst a better way to find patterns in the data. The visualization approach was already used in minor-scales in clusters, for example, to visualize the behavior of parallel applications. This use is mostly by the presentation of the information with Gantt Charts, each process being represented by rectangles. When dealing with Grids, which has a greater number of resources of different type of metrics to be visualized, the question is how to organize the information visually in order to obtain a understandable and significant visualization.

This text surveys information visualization techniques that can be used to visualize monitoring information. The text is also focused in how these techniques can be applied to Grid environments, mainly to show monitoring information from resources and applications.

The paper is organized as follows: section 2 presents the information visualization techniques already present today that are used to show a hierarchy of information. Section 3 discuss about the application of these techniques in Grid environments and shows examples of existing tools that employ some of the techniques. Finally, section 4 gives a summary of the discussion.

2. Basic Concepts

The classic way of visualization hierarchical information is by drawing node-link diagrams and display them in 2D screens [20]. Figure 1(a) shows node-link diagrams, with nodes representing the information and the links between these nodes showing the hierarchical connections that connect the information. One way to improve this visualization is to merge properties in each node, like the one show in figure 1(b). The additional information is presented close to the nodes and may represent aspects of the data the node represents.

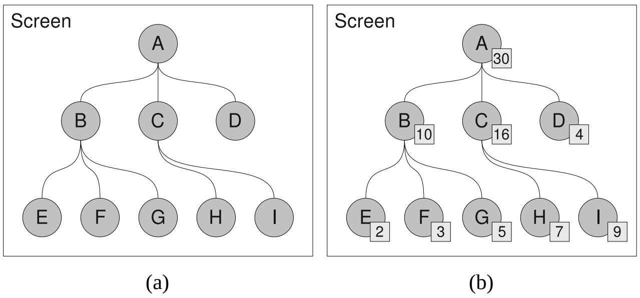


Figure 1. Classical hierarchical visualization with 2D images. Additional information may be inserted close to the nodes and leafs of the tree

To improve the readability of the node-link visualization, other ways to visualize the hierarchical data were presented. One is the Cone-tree approach [21, 22], which applies the use of cones in a virtual 3D environment to show the information. Figure 2 shows the Cone-tree visualization in two different ways. The first one shows a top-down approach and the second presents the data horizontally. This kind of visualization evolved with various interaction techniques [10]. These techniques work mostly by improving the interaction with the user, with animated rotations and zooming, for example.

The problem of these techniques for hierarchical visualization is that they do not scale well. In scenarios with few nodes to be shown, the node-link and cone-tree techniques can be applied. However, if a bigger scenario is taken into account these techniques cannot be used. This happens mostly because of the misuse of the screen space [25]. Even if all the nodes are shown in the same screen, it is difficult

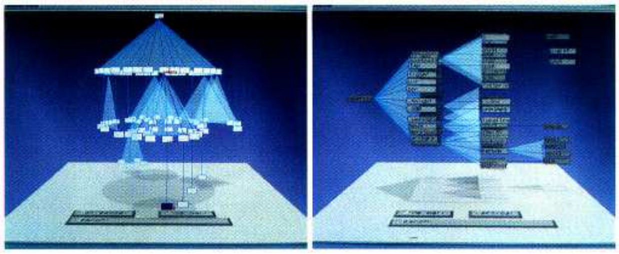


Figure 2. Cone-tree Visualization of hierarchical data [21]

to detect the properties of the nodes, or to understand possible patterns in the visualization.

The treemap technique [17, 25] is a solution to a better use of the screen space. Instead of showing nodes and links among them, it uses all the screen space by using a space-filling algorithm. Figure 3 shows an example of the creation process of a simple treemap based on the hierarchy of figure 1. The algorithm is as follows. For each level of the hierarchy, it creates a different schema of partitioning, changing from horizontal to vertical slices. Every rectangle can be customized to show different colors according to a specific feature of the nodes. Another customizable characteristic is the size of the rectangle, which can be based in another property. The Treemap software [26] has features such as the exaggeration of a specific value associated to a node. Every time this occurs, the screen is updated and the node became more evident.

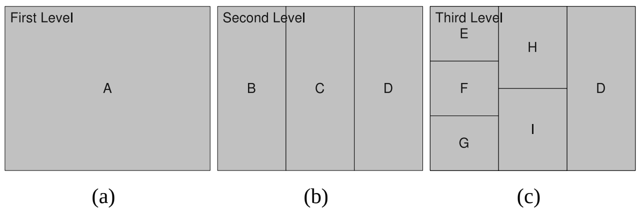


Figure 3. Three levels of a treemap construction based on the tree shown in figure 1(a). Every rectangle can be associated with a specific color and size, according to the characteristics of the node

The treemap algorithm have been evolving since its creation. One of these changes is called the Cushion Treemaps [31]. The difference in this version, each rectangle has an intuitive shading which increases the user perception of what is being visualized. Another development done in treemaps is the concept of Squarified Treemaps [7]. It was proposed to solve the problem of the classical treemap algorithm in the creation of the rectangles in the screen. The problem is that in the classical version, if a node in the hierarchy has many sub-nodes, the visualization may have many thin rectangles that turns difficult the visualization. The Squarified algorithm tries to keep the rectangles appearing like squares.

When the user interacts with a treemap visualization, by using any of the methods above, it is possible to lose the reference of proximity of the nodes when zooming to different levels. To address this issue, it was proposed a different version of treemap called Ordered Treemaps [27]. This version of the algorithm tries to keep the visual relation among the different screens generated when zooming to different levels. The idea of ordering the rectangles also appears in the thumbnail visualization of images. This approach was called Quantum treemaps [2, 3], which imposes a relation from the size of the rectangle with the image preview.

Voronoi Treemaps [1] is a different approach to visualize hierarchical information. Figure 4 depicts a visualization of a tree created with the algorithm. Instead of using rectangles, the Voronoi algorithm uses polygons in order to better represent the relation among them, visually.

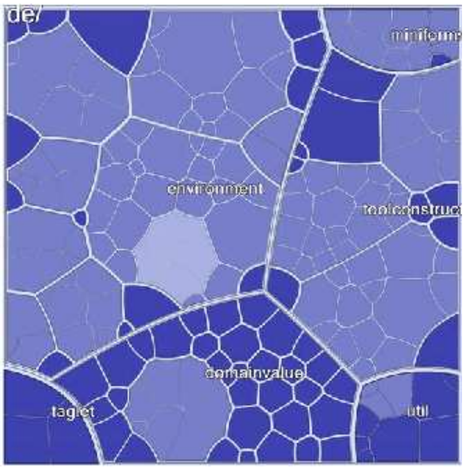


Figure 4. Voronoi treemaps with polygons instead of rectangles [1]

Beamtrees [29] is based in the original treemap visualization. But instead of using the hole screen space to visualize information, it scales down the rectangles in order to increase the perception of the hierarchy itself. The use of techniques like shading in the 2D visualization improves the notion of depth. The beamtrees were also presented with 3D graphics, by using cylindrical objects to represent each level. In both 2D and 3D, the user can change the size of the space used to represent a node or a leaf of the hierarchy by using a percentage of the space used by the original treemap algorithm.

3. Applications in Grid Contexts

Examples of utilization of treemaps include network monitoring with security purposes [18], grid resource monitoring visualization [23], visualization of photos within the treemap [4], visual analysis of stock market [30] and even applied to visualize a million items [12].

Grid environments can be characterized as being the aggregation of multiple resources that are spread geographically. Grid5000 [8] is an example of this kind of Grid. It gathers 9 sites distributed in France featuring a total of 5000 CPUs. This section discuss how the techniques previously described can be used in a grid environment, like Grid5000.

3.1. Hierarchy Visualization

The visualization of the Grid can be achieved by organizing the network in different ways. One way to visualize is to organize the data based in the network organization. Figure 5 shows the connections in Grid5000 with the Renater network.

As an example, we can organize the nodes using a single site as a point of view. This can be useful to the user since it has a notion of how many routers the communications will have to pass if its application is running in more than one site. Figure 6 shows the hierarchy in the point of view of the Grenoble site and the corresponding visualization in a treemap fashion. If each of the sites has multiple nodes, they can be drawn inside the appropriated spaces in the visualization to represent computing nodes, for instance.

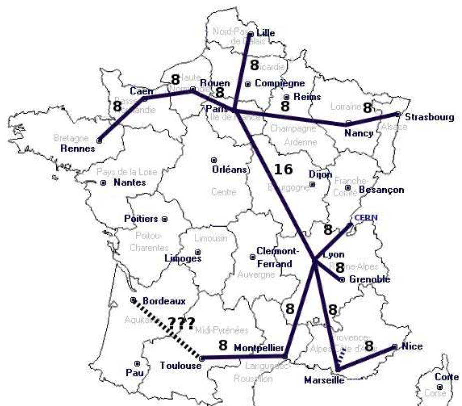


Figure 5. Grid5000 network organization using the renater dark fibre [8]

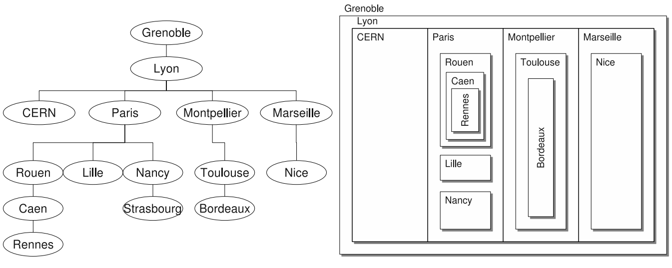


Figure 6. (a) Hierarchy visualization of Grid5000 network organization in the Grenoble point-of-view (b) Treemap associated with that point-of-view

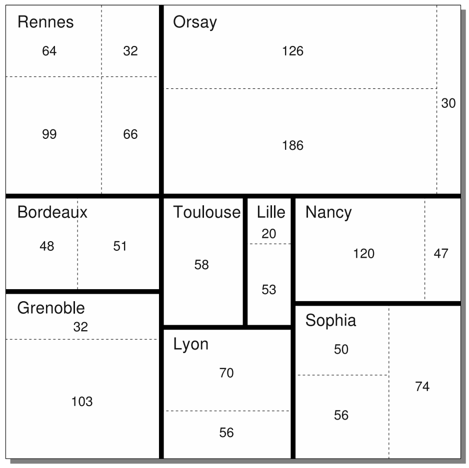


Figure 7. Logical Hierarchical Definition of Grid5000 based in the number of nodes [8], and a division in *Grid - Sites - Cluster - Nodes*

Another way to visualize the Grid as a treemap fashion is to divided the physical sites in logical hierarchical definitions. In this way, we could have the term *Grid*, representing the hole Grid5000; the term *Site* representing each of the sites spread geographically; the term *Cluster* to represent an aggregation of nodes and finally the term *Node* to represent a useful computing resource. Additional definitions could be added to the hierarchy in order to better represent the system is being visualized. By taking the example of Grid5000, figure 7 shows a treemap that visualizes in a single screen all the resources of the 9 sites of the Grid. Each rectangle has an approximate size based in the number of nodes of each site. Figure 7 also shows the specific clusters of each site. Orsay, for instance, has 3 clusters, one with 126 nodes, another with 186 nodes and the smaller one with 30 nodes. The darker lines show the separation among the sites, and the dashed lines shows the separation of each cluster in a site. This visualization can be more specific if the nodes of each cluster are added. Colors can be added to represent a specific feature collected, like the CPU utilization or the memory available.

3.2. Interaction Mechanisms

Using graphical visualization, the users can benefit from interaction mechanisms, like zooming, animations and online update. These techniques are used to improve the user perception to take conclusions of what is being analyzed. The zooming technique, for instance, can be applied in the example of the figure 7 by choosing one of the sites. After the zooming, the site chosen appears in the hole screen and additional information can be visualized. This technique was applied in the visualization of stock market [30] and is also present today in the *Map of the Market* [19].

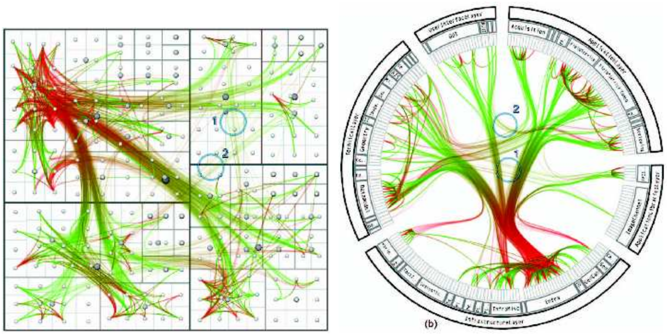
The graphical animation with online updating can also be applied by dynamically change the graphical visualization. This technique resizes the rectangles and changes their colors, for example. These changes usually reflect new values collected by the monitoring system. Another type of interaction mechanism is the distortion techniques [9], which tries to amplify the visualization in specific regions of the graphic. The fisheye technique [24] is an example that helps the user to obtain details, without losing the context. Three dimension visualization [5, 6] was also considered to visualize treemaps. Following the example of figure 7, the 3D approach can show boxes on top of the rectangles that represents a site. These boxes can be the graphical visualization of the nodes. If the Grenoble rectangle is taken into account, and the rectangle with value 32 is chosen, 32 boxes representing the nodes are drawn. The height of these boxes can represent how much memory they have, for instance. A more developed 3D approach to treemaps is the Treecube concept [28], where a cube in a 3D environment represents the data to be visualized.

3.3. Beyond Resources Visualization

Most of the visualization techniques already presented show the data in the point of view of the resources. In Grid'5000 example, shown in figure 7, the visualization presents the node count for each site separated in rectangles. A different approach is to give the users a point of view of the applications and tasks the users own.

This different approach still needs as basis the resources used by the applications. The use of treemap as this basis are often used to give the user a notion of what are the resources in use [11]. If we apply the techniques of software calls visualization presented by Holten [15, 16] in Grid contexts, we can visualize the parallel application on top of a treemap (Figure 8(a). The processes of each site may be represented by circle dots in the rectangles. The communication among the processes may be represented by arrows or links. Different variations of using this technique can be arranged if the interaction mechanisms, like zooming, animations and online update are used together with treemaps.

The use of circles as the resource base of the visualization of a parallel application can also be used. The approach was also used by Holten [15] to show the links bundles technique he used to turn better a visualization with lots of links. Figure 8(b) shows an example of this approach.



(a) (b)

Figure 8. Figures extracted from [15]. Examples that can be used in Grid context showing communications and processes: (a) treemap based and (b) circles based

3.4. Visualization Tools Examples

CoViz [23] is a Grid visualization tool developed for PlanetLab environment. This tool renders a treemap graphical representations to show different characteristics of the Grid, collected with the underlying software called CoMon. All the information is update every 5 minutes. CoViz offers to users and administrator eight types of images, organized in three different categories. The first category, Resources, shows images about CPU utilization, memory and bandwidth. The second, Efficiency, shows two types of graphics that presents bandwidth as a function of CPU or memory. The third category, Usage, is divided in three possible graphical visualizations: slices, slivers and nodes. Each of these graphics has different colors to represent additional information. The size of each rectangle represent how significantly the node is among all the others. Figure 9 shows the treemap visualization generated by CoViz with information about the CPU. Each rectangle in the figure represents a set of nodes, and the numbers inside the rectangles represents the value used to generate the size of the rectangle. The colors represent how much of the CPU is being used.

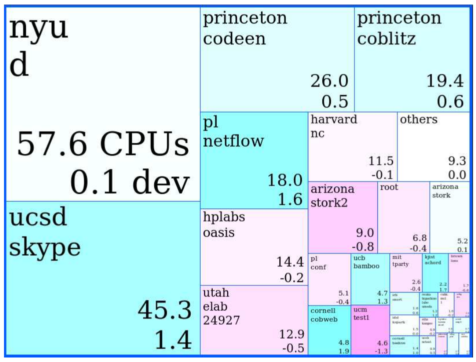


Figure 9. Treemap visualization of CPU load with the CoViz [23]

Treemaps were also used to visualize the workload in grid and distributed systems [14]. This work was published before CoViz announcement, probably being the first to apply the treemap concept for distributed systems. Another work, that combines traditional node-link views with treemaps, is called Elastic Hierarchies [32].

4. Conclusions

This paper presented different techniques that are used in the information visualization research area. It was also shown that treemaps were largely used in different fields of computer science. In grid context, their implementation were applied to resource visualization, through CoViz [23].

The work presented by Holten [15], which develops a technique to make bundles of links, was applied to software visualization. This work can be extended to the visualization of parallel applications on top of the resources they use. The use of edge bundles can be used in grid context together with treemaps to show the parallel applications behavior on top of the resources of the grid. Since 3D environment and colors in treemaps were already proposed, a big number of information can be provided in a single screen. The zooming techniques can be merged with monitoring techniques in order to collect only the information needed to render a single screen.

References

[1] M. Balzer, O. Deussen, and C. Lewerentz. Voronoi treemaps for the visualization of software metrics. In *SoftVis '05: Proceedings of the 2005 ACM symposium on Software visualization*, pages 165–172, New York, NY, USA, 2005. ACM Press.

[2] B. Bederson. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 71–80, 2001.

[3] B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics (TOG)*, 21(4):833–854, 2002.

[4] B. Bederson, B. Suh, and H. Kang. Photomesa version 3.1.2. http://www.cs.umd.edu/hcil/photomesa/. Last accessed February 20, 2007.

[5] T. Bladh, D. Carr, and M. Kljun. The Effect of Animated Transitions on User Navigation in 3D Tree-Maps. In *Proceedings of the Ninth International Conference on Information Visualisation*, pages 297–305, 2005.

[6] T. Bladh, D. Carr, and J. Scholl. Extending Tree-Maps to Three Dimensions: A Comparative Study. In *Proceedings of the 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI 2004)*, pages 50–59. Springer, 2004.

[7] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. In *Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42. IEEE Press, 2000.

[8] F. Capello, B. Plateau, T. Priol, D. Vandromme, F. Cappello, M. Dayde, F. Desprez, E. Jeannot, Y. Jegou, S. Lanteri, N. Melab, R. Namyst, M. Quinson, O. Richard, and P. V.-B. Primet. Grid5000, February 2007. http://www.grid5000.fr. Last accessed February, 26, 2007.

[9] M. Carpendale, D. Cowperthwaite, and F. Fracchia. Extending Distortion Viewing from 2D to 3D. *IEEE Computer Graphics and Applications*, 17(4):42–51, 1997.

[10] J. Carriere and R. Kazman. Research report. Interacting with huge hierarchies: beyond conetrees. In *Information Visualization, 1995. Proceedings.*, pages 74–81, 1995.

[11] J. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant. Overlaying Graph Links on Treemaps. In *Proceedings of IEEE Information Visualization Symposium Posters Compendium*, 2003.

[12] J.-D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 117, Washington, DC, USA, 2002. IEEE Computer Society.

[13] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222, 2001.

[14] S. Heisig. Treemaps for workload visualization. *Computer Graphics and Applications, IEEE*, 23(2):60–67, 2003.

[15] D. Holten. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Trans Vis Comput Graph*, 12(5):741–8, 2006.

[16] D. Holten, R. Vliegen, and J. van Wijk. Visual Realism for the Visualization of software Metrics. In *Proceedings of VISSOFT*, pages 27–32, 2005.

[17] B. Johnson and B. Shneiderman. *Tree-Maps: a space-filling approach to the visualization of hierarchical information structures*. IEEE Computer Society Press Los Alamitos, CA, USA, 1991.

[18] S. Mansmann, F.; Vinnik. Interactive exploration of data traffic with hierarchical network maps. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1440–1449, Nov-Dec 2006.

[19] S. Money. Map of the market, 2007. http://www.smartmoney.com/marketmap/. Last accessed February 27, 2007.

[20] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In *Proceedings of International Conference on Pattern Recognition*, volume 1, pages 347–349, 1984.

[21] G. Robertson, J. Mackinlay, and S. Card. Cone Trees: animated 3D visualizations of hierarchical information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pages 189–194. ACM Press New York, NY, USA, 1991.

[22] G. G. Robertson, S. K. Card, and J. D. Mackinlay. Information visualization using 3d interactive animation. *Commun. ACM*, 36(4):57–71, 1993.

[23] S. Saengsuwarn and V. Pai. Coviz - visualization of planetlab project, February 2007. http://codeen.cs.princeton.edu/coviz/. Last accessed February, 20, 2007.

[24] M. Sarkar and M. H. Brown. Graphical fisheye views. *Commun. ACM*, 37(12):73–83, 1994.

[25] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.

[26] B. Shneiderman, C. Plaisant, G. Chintalapani, and A. Aris. Treemap 4.1.1, February 2004. http://www.cs.umd.edu/hcil/treemap/. Last accessed February, 19, 2007.

[27] B. Shneiderman and M. Wattenberg. Ordered Treemap Layouts. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*. IEEE Computer Society Washington, DC, USA, 2001.

[28] Y. Tanaka, Y. Okada, and K. Niijima. Treecube: Visualization tool for browsing 3d multimedia data. In *IV '03: Proceedings of the Seventh International Conference on Information Visualization*, page 427, Washington, DC, USA, 2003. IEEE Computer Society.

[29] F. van Ham and J. van Wijk. Beamtrees: compact visualization of large hierarchies. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 93–100, 2002.

[30] M. Wattenberg. Visualizing the stock market. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 188–189, New York, NY, USA, 1999. ACM Press.

[31] J. J. V. Wijk and H. van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, page 73, Washington, DC, USA, 1999. IEEE Computer Society.

[32] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 8, Washington, DC, USA, 2005. IEEE Computer Society.