

# Large Visualizations for System Monitoring of Complex, Heterogeneous Systems

Daniel M. Russell, Andreas Dieberger, Varun Bhagwan, and Daniel Gruhl

IBM Almaden Research Center,  
650 Harry Road, San Jose, CA 95120 USA  
{daniel2, andreasd, vbhagwan, dgruhl}@us.ibm.com  
<http://www.almaden.ibm.com>

**Abstract.** As systems grow larger in size and complexity, it becomes increasingly difficult for administrators to maintain some shared sense of awareness of what's going on in the system. We implemented a large public display with appropriately designed visualizations that allow for rapid assessment and peripheral awareness of system health. By placing the visualizations on a large display in a shared, commonly used team location, system administrators can monitor behavior as they walk past. Such a display helps administrators identify emerging problems early on and be a focal point for discussions of the system. It allows them not only to share information with colleagues on an “as-noticed” basis, but also highlights interconnected problems that would not be otherwise evident. We found that this approach significantly reduces the workload of individual system administrators, changing the nature of their work by radically simplifying a complex task through social sharing of peripherally noticed state.

## 1 Introduction

Complex systems exhibit complex failure modes. Traditional system monitoring tools work well for small and isolated problems, but they quickly become swamped when cascading faults occur. System admin experience teaches us that the sheer number of things going wrong in a very short period of time requires novel approaches to monitoring and information sharing. Such approaches are not meant as a replacement but as a complement to traditional methods of administering these systems.

As an example of a large, heterogeneous system with complex behaviors, the WebFountain [3] system developed at the IBM Almaden Research Center is ideal: it is a large, loosely coupled system of over 500 multi-processor computers working on very large-scale text analytic problems. WebFountain has shown a wide variety of complex problem modes which have proven very difficult to diagnose using traditional tools.

To address these issues we built Shepherd, a visualization tool running on a large display, to help us gain a high-level overview and awareness of the health of the entire cluster. The tool does not show all details about every aspect of the system, but instead provides a global overview of the cluster. It runs on a large, public display located in a space shared by all sysadmins. Shepherd's goal is to give admins a one-glance view of the overall system health such that they can notice if something is peculiar or out-of-balance, and a touch-screen to allow immediate drill-down to details.

## 2 Shepherd: A Tool to Monitor Large System Behavior

Over the past several years the cost basis of running large systems has significantly shifted from large hardware and software costs to the point where people costs now dominate the cost of large systems infrastructure. The more time people spend waiting for a problem to develop, the less time they can spend actually running and tuning the system, performing upgrades, isolating problematic subsystems, etc.

Therefore, a monitoring solution based on a low-level awareness of system state, that allows operators to notice problems as they develop without time-consuming conscious monitoring would allow for tremendous time and cost savings. [2] Once a problem is noted, an analysis needs to be made as to the extent of it, and a solution strategy quickly developed and implemented – as soon as the issue is resolved the operations staff needs to get back to their “day job.” [1]

Shepherd is built around a simple visualization designed to be rapidly perceivable. People can easily spot asymmetry in a field of small multiples. Shepherd represents each machine in the cluster using a star visualization, where each point indicates a relevant status value for that machine, such as free memory, or CPU load. The values are normalized such that a “healthy” machine results in a mostly symmetric star.

Each row of small stars in Figure 1 corresponds to a numbered rack. Shepherd runs on a display with touch screen so administrators can just tap a star and it will move into the detail area on the right. There the visualization is shown larger and additional details on the machine are displayed below. With another double screentap, admins can open a secure shell connection to a machine shown in detail.

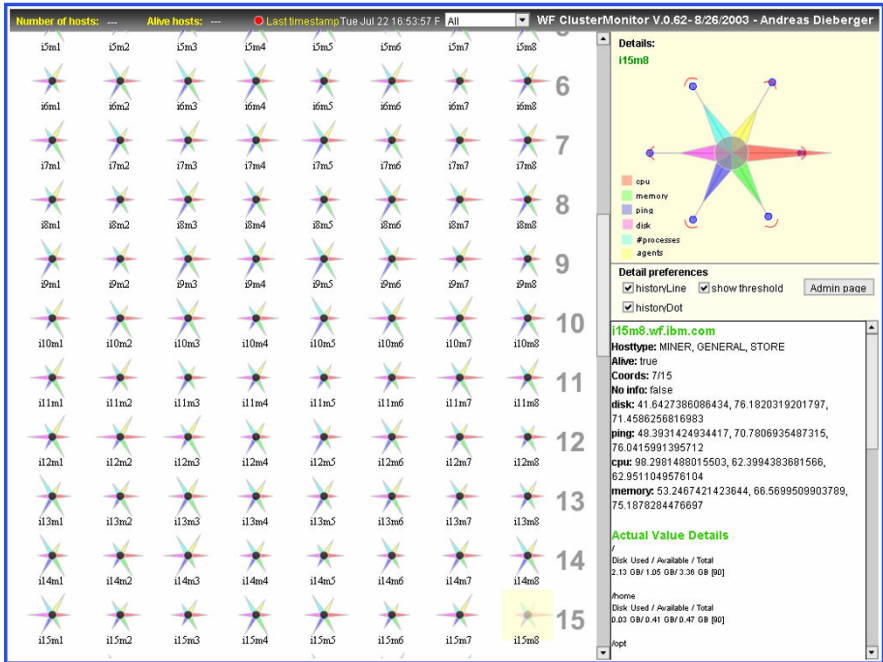
The physical layout of machines in the rack corresponds to the visualization layout, allowing problems stemming from local common resource problems (e.g., in-rack network bandwidth, in-rack cooling issues, a whole rack being down due to PDU issues, etc.) as well as point problems, or randomly distributed issues (unbalanced load on the cluster, for example) to be quickly understood as physical, not network, problems.

Shepherd is implemented in Macromedia Flash talking to a Java-based backend using XML Socket connections. It is deployed on an BlueBoard [4], a large display with basic touch screen capability provided by a touch overlay. The display was placed at the entrance to the sysadmin cubicle farm where it could be easily seen whenever entering or exiting the common workspace, but also from any point in the farm when standing up.

After Shepherd had been in place for six months, we conducted a set of semi-structured, in situ interviews with half of the end user population (6 admins) to discover their use patterns and war stories when the Shepherd display had been especially useful or problematic. These interviews revealed several strikingly consistent results:

*Simplified task:* The admins we interviewed uniformly told us that Shepherd simplified the task of correlating application problems and process tracing. For instance, if one node in the cluster is down, a quick visual scan will find other nodes may be down as well. The history line might indicate that network load recently was/is high and might suggest a diagnosis. The admin’s interaction with Shepherd allows one to

trace the entire path without going through tedious logs and notifications. Users have found it is far simpler to see the patterns of failure using Shepherd than to mentally correlate co-failures of nodes with arbitrary names.



**Fig. 1.** The Shepherd cluster status visualization takes advantage of symmetric displays to permit rapid determination of variance from standard server behavior. The star's points show current values and the gray lines show a 30 minute average of the same value, providing trend information. Blue dots make the end of the gray lines more visible from a distance. Alert thresholds are shown as red curves on the spikes. The center circle indicates special situations (crashes, alerts) by changing color.

*Shows large-scale state effectively:* In several cases, an administrator was able to spot a rapid rise in disk utilization (caused by an incorrectly configured link) early on because an obvious spike appeared on one of the stars. The spike became immediately visible to passing admins, and the problem was caught before it became serious.

*Social sharing of task information:* A somewhat unanticipated effect of Shepherd was social sharing of team information. As admins walked by they would glance at the display, informally noting the current status of the system. Since their job involves continuously moving about, the net effect was that they noticed not just problems that affected their own task assignments, but also those of their colleagues. Telling someone to “look at the big display” became a common way to let someone else know that a significant change in their part of the system was underway. In the end, this meant that more eyes were watching for common problems in the cluster, but each person spent less time doing so. [5]

*Focal point for team discussions:* Teams also found that they would gather at the big display to talk over what was going on with the system in the large. Since the display wasn't owned by any one person and constantly showed overall system state, teams would gather at the display, discussing the visualization and using the access tools to get more information to investigate ideas about what was going on and why.

*Daily time savings:* The consensus of opinion among the admins is that Shepherd saved them typically one hour a day per admin. This estimate is based on time saved by debugging complex problems faster, in turn made possible by having a dedicated public display resource with constantly updating performance monitoring data. This freed up time from repeatedly looking up information that was instantly available.

### 3 Summary

The utility of placing an “ambient” monitoring system in the ops room cannot be overstated in terms of its effects on the performance of the sysadmin team. We found that an overview visualization of a complex system allows administrators to gain a general awareness of system health and allows them to discover emerging problems in a cluster early on. Large groupwork displays are becoming common in practice and research settings. Typically, these systems are either public displays of personal information, group work space or generic displays for common awareness purposes. By contrast with other system state display visualizations [6], we have shown that Shepherd dramatically changes admin behavior by providing peripheral awareness of critical system status and a place to interact with the information.

### References

1. Barrett, R., Kandogan, E., Maglio, P., Haber, E., Takayama, L. A., Prabaker M. “Cases from the field: Field studies of computer system administrators” Proc. 2004 ACM Conf. on Computer supported cooperative Work (2004)
2. Card, S. K., Mackinlay, J. D., Shneiderman, B. *Information visualization, Readings in information visualization: using vision to think* Morgan Kaufmann Publishers Inc., San Francisco, CA, (1999)
3. Gruhl, D., Chavet, L., Gibson D., Meyer, J., Pattanayak, P., Tomkins, A., Zien, J. ” How to build a WebFountain: An architecture for very large-scale text analytics” IBM Systems Journal, v 43, n 1 (2004) , p. 64-77
4. Russell, D., Sue, A., “Large interactive public displays: Use patterns, support patterns, community patterns” in O’Hara, K., Perry, M., Churchill, E., Russell, D. M. (eds.) *Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies*. Kluwer Academic Publishers. (2003)
5. Russell, D. M., Drews, C., Sue, A. “Social aspects of using large public interactive displays for collaboration” Proceedings of Ubiquitous Computing (UBICOMP) Conference, Goteborg, Sweden (2002), 229-236
6. Skog, T. & Holmquist, L.E. WebAware: Continuous Visualization of Web Site Activity in a Public Space. In Extended Abstracts of Computer-Human Interaction Conf. (CHI) (2000)