Building End-to-End Management Analytics for Enterprise Data Centers

Hai Huang, Yaoping Ruan, Anees Shaikh
IBM TJ Watson Research Center

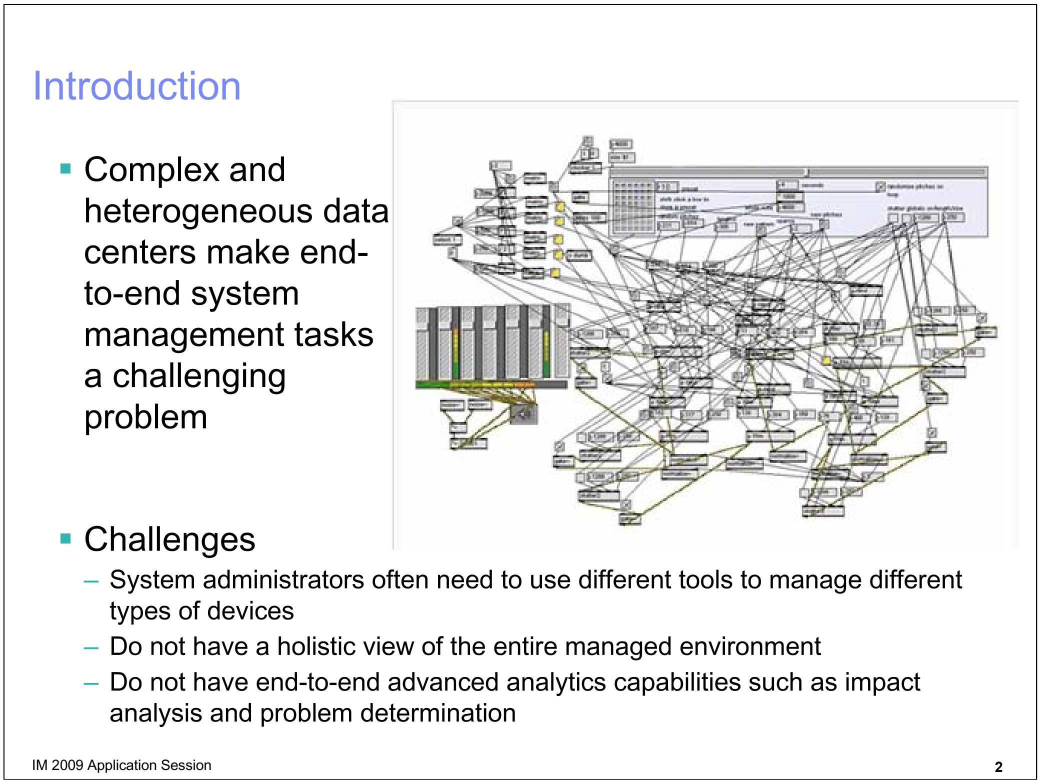
Ramani Routray, Chung-hao Tan, Sandeep Gopisetty
IBM Almaden Research Center

Abstract

The complexity of modern data centers has evolved significantly in recent years. One typically is comprised of a large number and types of middleware and applications that are hosted in a heterogeneous pool of both physical and virtual servers, connected by a complex web of virtual and physical networks. Therefore, to manage everything in a data center, system administrators usually need a plethora of management tools since one tool often manages only one type of devices. The boundaries between the different management tools can limit productivity of system administrators on their daily tasks as each tool only offers a partial view of the entire managed environment. As a result, advanced analytics such as impact analysis and problem determination are generally not achievable using the traditional management tools as they require a holistic view of the entire data center.

In this paper, we describe an integrated management system for applications, servers, network and storage devices called DataGraph. Our system integrates data across heterogeneous point products and agents for management and monitoring to enable the above mentioned management analytics capabilities. A common data model is introduced to federate data collected by the different tools in multiple database repositories so no modifications are needed to existing management tools. A common integrated web user interface is implemented to facilitate management tasks that would otherwise require invoking multiple tools. We deployed this tool in a lab environment and demonstrated these analytics capabilities through several case studies.

978-1-4244-3487-9/09/$25.00

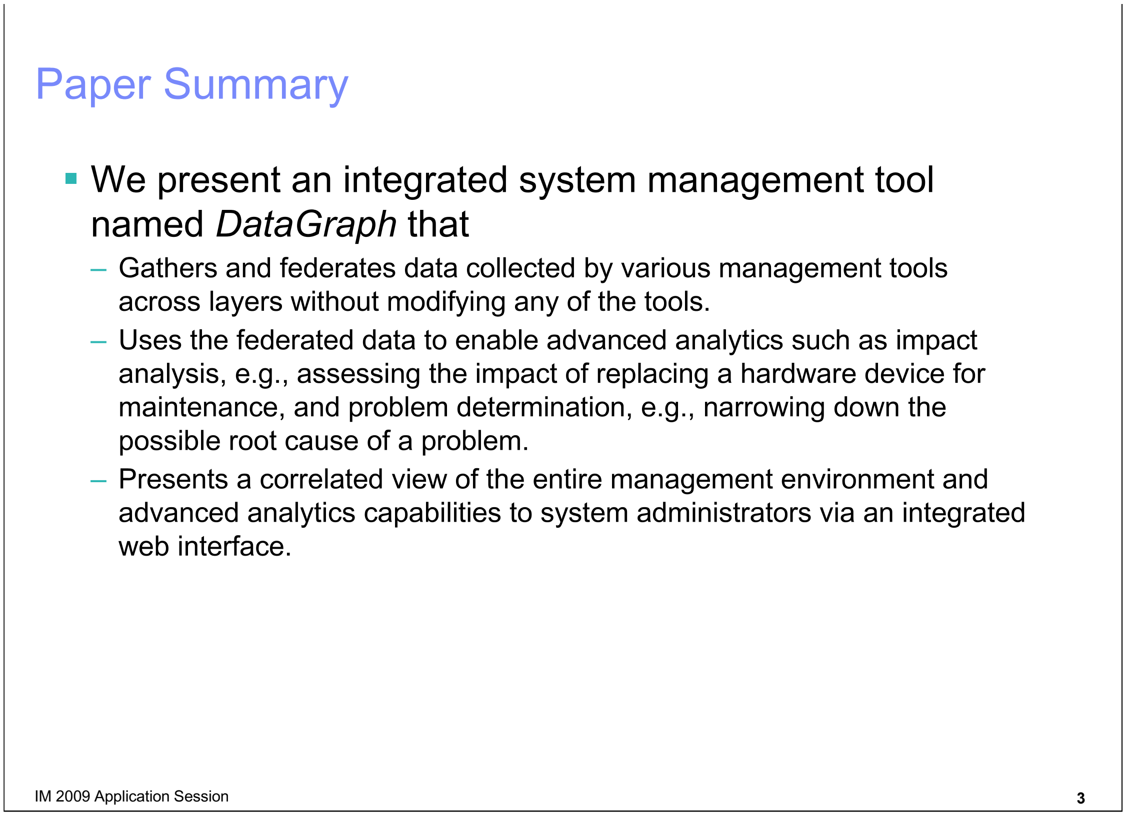


1. Introduction

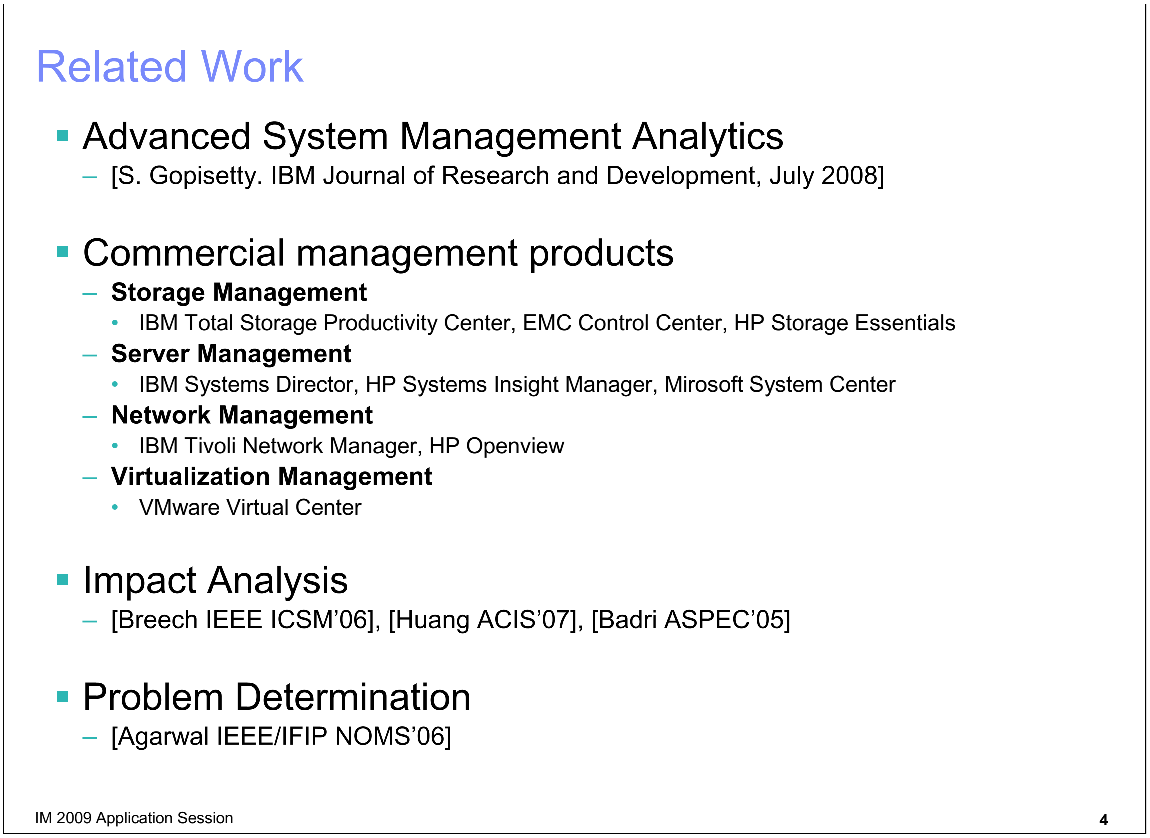
The rapidly growing cost of managing IT infrastructure is a well-known trend over recent years, and has been well documented by IT industry analysts. Recent studies have shown, for example, that management cost can constitute up to 60% of the total cost of ownership (TCO) in data centers. Moreover, it is expected that in the long run, the cost to of IT personnel and services will far outstrip the cost of hardware and software infrastructure.

To combat the challenges of managing complex enterprise data centers, IT staff turns to a variety of management tools, particularly for platforms and devices (e.g., networking infrastructure, various server types, and different classes of storage subsystems). Often these tools are provided by the infrastructure vendors themselves, or are provided by third party ISVs who specialize in managing specific types of devices. Naturally, systems administrators often adopt a "best of-breed" approach, using those tools that are best-suited for various management tasks. While this approach provides a high level of function for each specific area, the maintenance and training costs associated with deploying many tools becomes prohibitive. More importantly, the narrow focus of the different management tools often limits productivity of systems administrators in their operational tasks, as each tool only offers a partial view of the entire managed environment.

Ideally, an end-to-end systems management tool needs to gather data across heterogeneous devices and across IT stack to present an uniform correlated view of the managed environment. Such a federated view is necessary for key management analytics capabilities such as impact analysis and root-cause problem determination. Currently available commercial tools are limited in their ability to consolidate and correlate data across layers, i.e., from application layer all the way down to storage layer, flowing through server and networking layers.



In this paper, we describe an integrated management system for applications, servers, network and storage devices called DataGraph. Our system integrates data across heterogeneous point products and agents for management and monitoring to enable the above mentioned management analytics capabilities. A common data model is introduced to federate data collected by the different tools in multiple database repositories so no modifications are needed to existing management tools. A common integrated web user interface is implemented to facilitate management tasks that would otherwise require invoking multiple tools. We deployed this tool in a lab environment and demonstrated these analytics capabilities through several case studies.

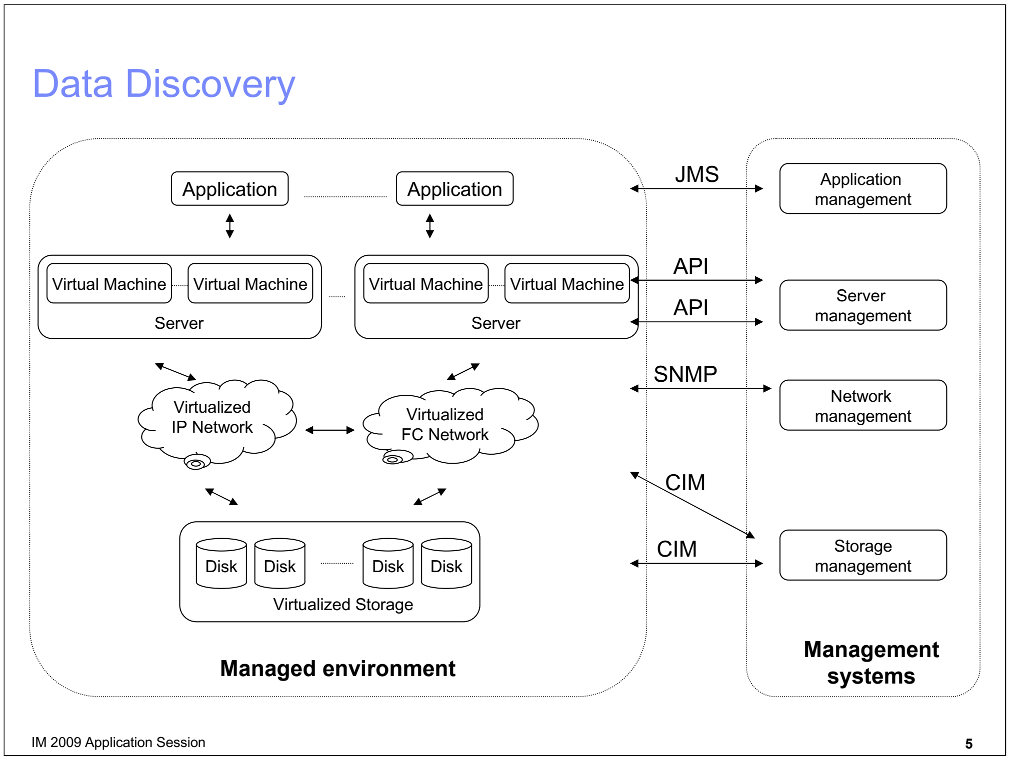


2. Related Work

DataGraph is an extension of a research effort [16] that demonstrates an advanced architecture which presents formatted data correlated in a scalable relational database repository. This repository forms the basis for any advanced analytics that could increase the productivity of modern enterprise data center.

Several products in the market place have tried to address the customer pain points of end-to-end data path tracking. But, most of them have focused on a single layer of the stack. Products such as [11], [2], [5] have focused on the storage management, [6], [3] have focused on server management and [10], [4] have focused on Network management. Approach of these management suites are complementary to our approach and can be consumed as an input data source into our Data-Graph tool.

Lots of work has been done regarding impact analysis in the area of software maintenance [14], [17], [13]. But, these approaches have not been explored in the area of systems management. DataGraph can incorporate the techniques learned from impact analysis in the area object oriented software maintenance. DataGraph also tracks the history of device characteristics that can be leveraged for problem determination by applying temporal analysis [15].

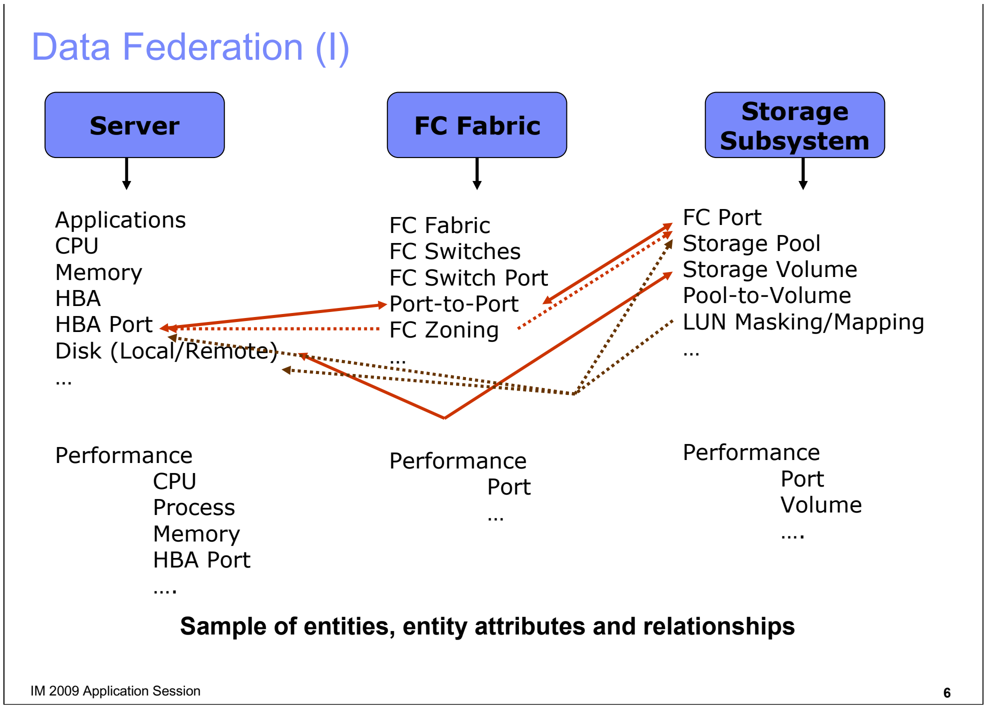


3. Data Discovery and Federation

3.1 Data Discovery

Each management tool suite usually needs to gather two basic categories of data: (i) entity configuration and topology data, and (ii) entity performance and availability data. Though almost all hardware and software vendors provide management interfaces for management tool to collect these data from each entity, there is a wide spectrum of interfaces and tools available to select from. Basic support for standard-based exposure of management information usually is available nowadays. For example, Simple Network Management Protocol (SNMP) is commonly supported by network devices such as network switches and even some adapters. Common Information Model (CIM) has been supported by increasingly more OS vendors. For information which cannot be obtained through these standard interfaces, management tools may deploy their own data collecting agents on the managed system to collect data through proprietary interfaces. Vendors also provide management suites for a class of devices known as element managers such as IBM DS Management Console for its DS series of storage subsystems. Also, management suites that provide management capabilities focusing on one particular layer of the IT stack are also available such as IBM TotalStorage Productivity Center and EMC Control Center for storage management, ITNM and HP Openview for network management. Later in the paper, we would be using the term Management Information Point (MIP) to refer to any of the above entity that exposes management information or control point.

DataGraph implements a framework that addresses the above mentioned problem by first providing a platform that enables information gathering across the layers by providing an end-to-end data center systems management repository.



3.2 Challenges in Data Federation

• Sequence in which devices are discovered and data is reported: DataGraph starts a set of discovery processes that talk to device management information points in a parallel fashion. But, the relational schema has to be designed in such a way that persistence and representation of data is agnostic to the sequence in which data is retrieved from multiple points. For example, fiber channel fabric discovery would discover and persist the connected storage port World Wide Name (WWN) without its parent entity. Details about the parent entity (storage subsystem) of the port would get persisted after the storage subsystem discovery is done.

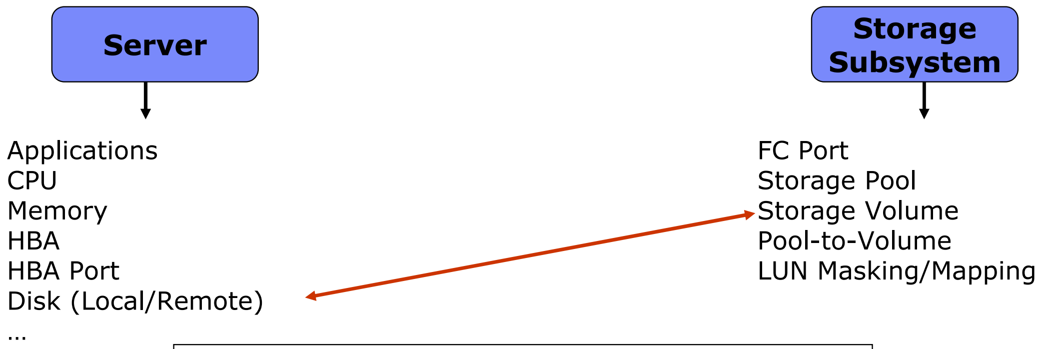
• Identity reconciliation. Identity of an entity can be exposed by multiple management interface points in different ways. This scenario occurs due to loose adoption of standards or proprietary extensions by vendors. DataGraph uses a set of custom algorithms to reconcile the naming convention differences by maintaining a knowledge base of standard naming formats, derivatives and vendor extensions.

• Global Namespace: DataGraph maintains the namespace in the context of the entire managed environment so that each device can be identified back for communication after the information gathering process.

• Data Correlation: Correlating information gathered from different management interface points is one of the major tasks of our tool. This module provides the basis for creating end-to-end data path for the hosted applications. Some examples of this step are the port to port connectivity correlation, server to storage correlation, etc.

• Monitoring. Once, data is discovered and correlated, our monitoring infrastructure keeps the data fresh with respect to the change in configuration by subscribing to different monitoring primitives such as CIM indications, SNMP traps, system events

Data Federation (II)



Disk: Remote
StorageSubsystemID: …BC4563E / StorageVolumeID: …02AF415 : Vendor: IBM

Sample discovery raw data for remote disk

entity1	entity1 class	entity2	entity2 class	…
swarm-bs11	HOST	disk1	DISK	
swarm-bs11	HOST	C:	FILESYSTEM	
swarm-bs11	HOST	DB2	APPLICATION	

Sample relationship table for HOST

IM 2009 Application Session 7

3.3 Data Federation

The federation operation is performed by a series of SQL queries, which correlate complementary information and reconciliate redundant information reported by different management discovery agents. The naming convention of different tools is captured in a mapping table. To capture the relationship between different entities, we define a series of entity classes and categorize them into three domains:

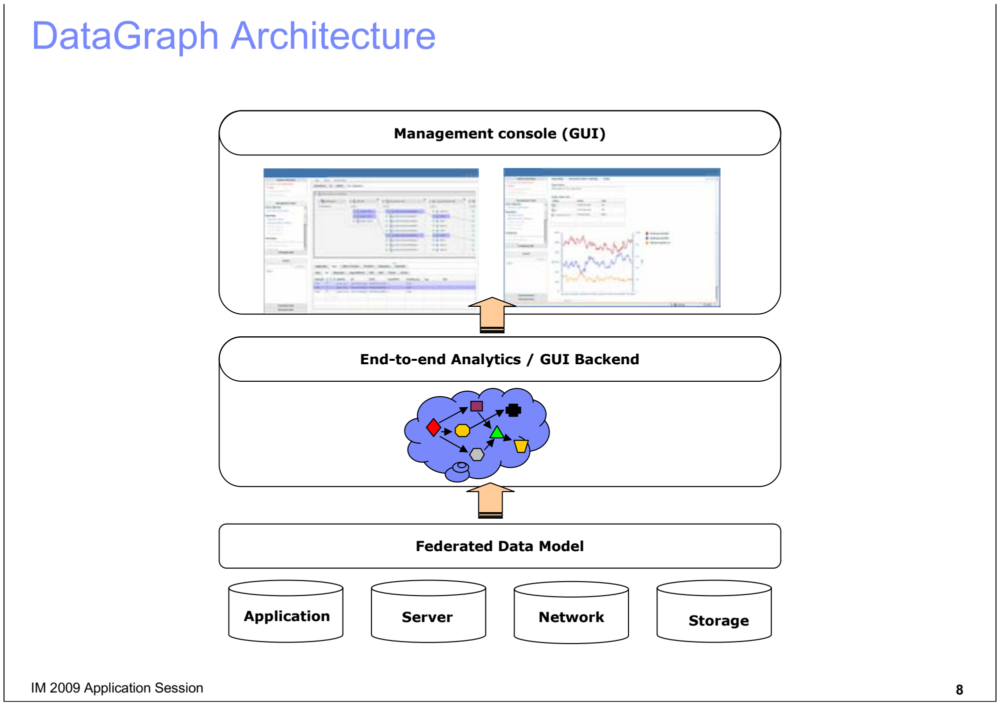
• host domain D{H}: APPLICATION, HOST, FILESYSTEM, HOST LOGICVOLUME, HOST DISK, HOST BUS ADAPTER, HBA FIBER CHANNEL PORT, HOST NETWORK PORT,

• link domain D{L}: FIBER CHANNEL SWITCH, FC FIBER CHANNEL PORT, NETWORK SWITCH, NETWORK SWITCH PORT

• storage domain D{S}: STORAGE SUBSYSTEM, STORAGE POOL, STORAGE SUBSYSTEM LOGICVOLUME, STORAGE SUBSYSTEM DISK

Our data tables are also named after these entity classes. Thus each entity class is corresponding to a data table which hosts all of the entities belong to this class.

Relationship information returned by discovery agents is usually undirectional, but rather, it is reachability information. For example, from a host's perspective, it can reach file system, disk, applications on it. It also can reach other physical entities such as fiber channel switch, IP network switch etc.



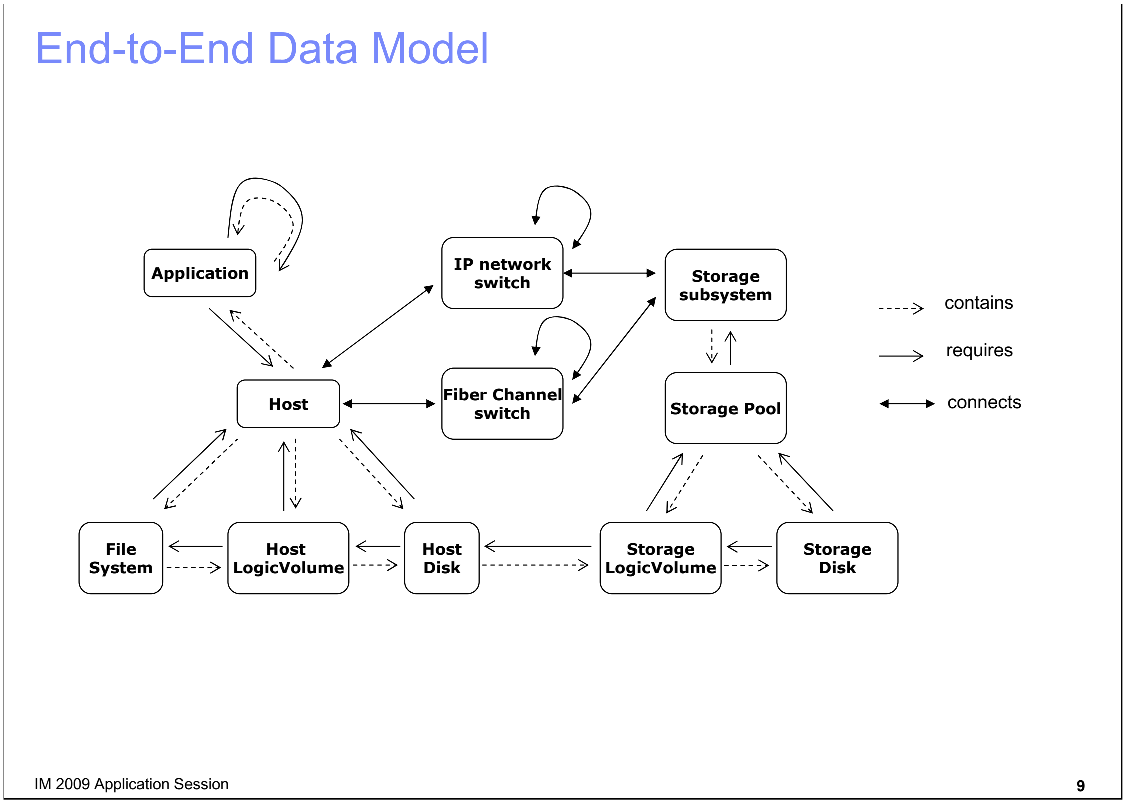
4. DataGraph Architecture & Implementation

4.1 Architecture

Once data is discovered and federated into our consolidated database, we build an in-memory end-to-end data model on top of the federated data to host our analytics capabilities. This data model also serves as the back-end for GUI display. Though entity relationships have been captured in the database, rendering an end-to-end view usually requires multiple data retrieval. Using mechanisms such as database Views may reduce the complexity of data retrieval. However, rendering data without an in-memory layer may significantly impact user perceived latency. More importantly, an in-memory data model provides flexibility to simulate some of the analytics such as impact analysis, which requires the status of the originating entity to be changed.

Our end-to-end data model essential is a linked graph with each entity as node, and relationships as arcs. Our analytics is largely based on such a linked graph. By traversing on the graph, we find entities which can reach each other thus derive the conclusion about impact analysis and root cause analysis. For other analytics which are not presented in this paper, for example, bottleneck analysis, the algorithms also can be based on the same linked graph we have here. Our graph traversal is different from simple event propagation which might be already available in some management suite, because our system traverses both physical and logical relationship and goes across of the IT functionality stacks.

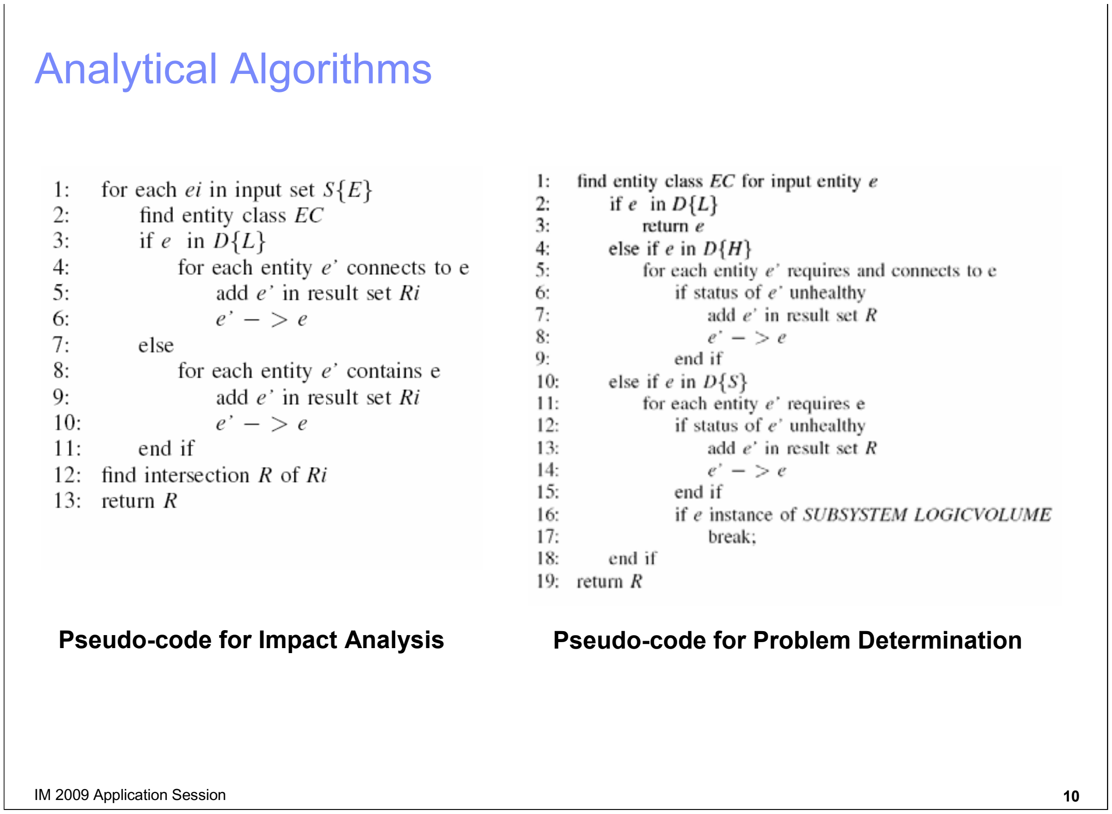
Some of our analytics require directional relationship in the end-to-end relationship. For example, in a scenario where a fiber channel switch is connected both to a host and a storage subsystem, failure of the fiber channel switch would have an impact on the host (or applications on the host), but not necessarily impact on the storage subsystem. For this reason, we define relationship between entity classes in three categories: CONTAINS, REQUIRES, and CONNECTS, to represent the dependents, pre-requisites, and peers for each entity. Based on the type of analytics is requested, only one direction usually is traversed.



4.2 End-to-End Data Model

Since our current prototype is targeted for enterprise LAN environment, layer 3 routing information is not available. Therefore, we have simplified to use use physical connectivity to capture the relationship within IP network. In our model, the relationship between host, IP network switch, fiber channel switch and storage subsystem is bi-directional, which is denominated as CONNECTS. Within these four entity classes, IP network switch and fiber channel switch may also connect to another entity in the same class. CONTAINS and REQUIRES are the two relationships between two entities with opposite direction. For example, a host contains an application and the application requires the host. A host may also contain one or more file system, host logic volume, host disk etc. Similarly, a storage subsystem may contain one or more pool, and a pool may contain some subsystem logic volume and subsystem disk etc.

Instead of building a directional graph consisting of all of the entities, we use a relationship model as a navigator for graph traversal. By using this model, the complexity of the end-to-end graph is significantly reduced. Data structure for each entity remains largely the same as their data table schemas. Figure 4 illustrates the data structure used to represent host logic volume. Reference to other entities is achieved by a global ID assigned during data federation process.

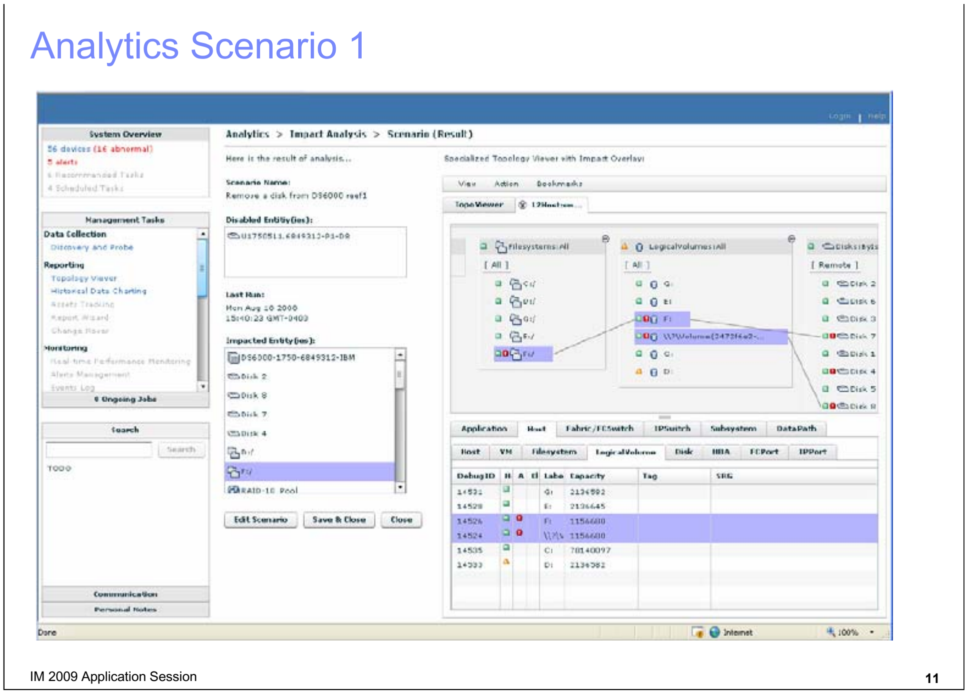


4.3 Analytical Algorithms

Once we have end-to-end data model for all of the cross layer entities, we implement graph traversal algorithms to find out the answer set. The algorithms are analytic capability specific since each capability may require different ways to verify each entity and may look for different format of answer set. This paper focuses on two analytics: impact analysis and problem determination. The main goal of impact analysis is to aid system administrator in understanding the impacted entities if one or more entities changes. The capability can be used in scheduling updates, risk management, or even in understanding the infrastructure.

As we discuss with system administrators, we define that the input of impact analysis is one or more entities, and the output should be a set of entities which are impacted by each of the input entity.

Similarly, the output of problem determination is also a set of entities. If the set contains only one entity, it is the possible root cause entity. If the set contains multiple entities, they are entities along the path which leads to the likely root cause. But different than impact analysis, we limit input of problem determination to only one entity to reduce the complexity of GUI display.



5. Analytics Scenarios

5.1 Impact analysis

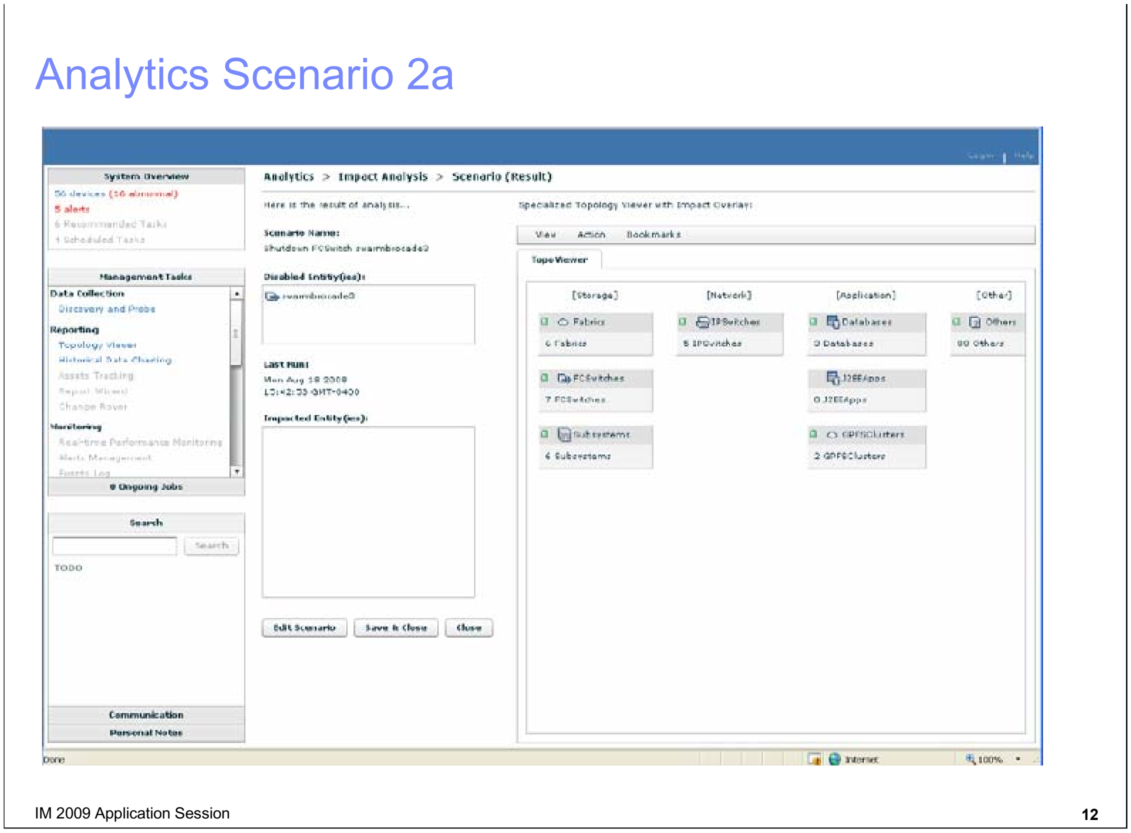
When a system administrator needs to perform a change in her managed environment, e.g., maintenance, upgrade, it is often a difficult job to assess the impact of these actions. However, in DataGraph, using the information various system monitoring tools have gathered on how various devices are connected, what software runs on which hardware and depends on what other services, and their causal relationships, impact analysis can be deduced to a simple graph traversal problem.

Scenario 1: disk replacement: In the first scenario, we assume a system administrator needs to replace a physical disk from a storage subsystem. This action will have profound implications to the entire environment, as it not only affects storage pools and volumes within the storage subsystem, it might also affect servers that mounted from those storage volumes. Furthermore, applications that make use of any files from the impacted file systems will also be affected. Other applications can also be indirectly affected if they make use of services provided by those applications that are directly impacted.

One can of course manually track these relationships among the various types of entities, but one can imagine this requires a lot of focus, a lot of time, and the system administrator to be highly experienced with all types of entities (network, storage, applications, etc.) in the environment, which is unlikely, to be able to back-trace all these dependencies.

With the information DataGraph gathers, this becomes a very simple task. In the graph we show a snapshot of DataGraph's web user interface. A system administrator can simply drag and drop a device (in this case, a physical disk in a DS6000), namely U1750511-6849312-P1-D8, into the "Disabled Entity" box. Clicking on the "Run Analysis" button (not shown) will give a list of impacted entities as shown in the figure.

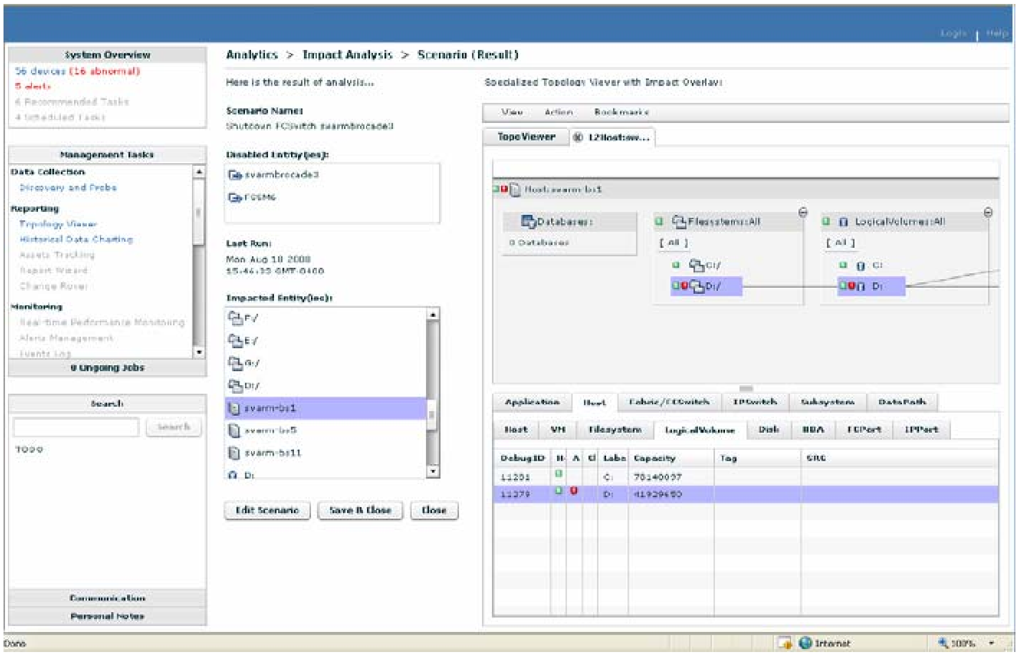
Essentially, we provide system administrators the ability of getting a quick view of what will be impacted if a particular action is taken. It gives system administrators' a chance to take remediation steps if an unexpected entity is going to be impacted for the action that is to be taken. As many IT problems are change related, and some due to dependencies not taken into account when the change was made, this can be a very useful tool to system administrators.



Scenario 2: fabric switch upgrade: Storage Area Network (SAN) is commonly deployed in large enterprise environments to provide isolation between network traffic and storage traffic. Fiber channel fabric topology is typically used to connect accessing elements (servers) to storage elements (volumes). For fault tolerance and security reasons, fabrics can sometimes be configured in very complicated ways.

In this scenario, we assume the system administrator needs to replace an old version of Brocade SAN switch with a newer model. She will first need to offline the old switch before replacing it with the new one. Before any action is taken, she can use DataGraph to simulate the impact of this potentially dangerous action by dragging the Brocade switch to the Disabled Entities box and clicking on the Run Analysis button. The result is shown in Figure 6. As we can see, no entity is impacted due to this action. The reason being there are enough redundancies in the fiber channel fabric that disabling one switch will not disrupt an existing data path.

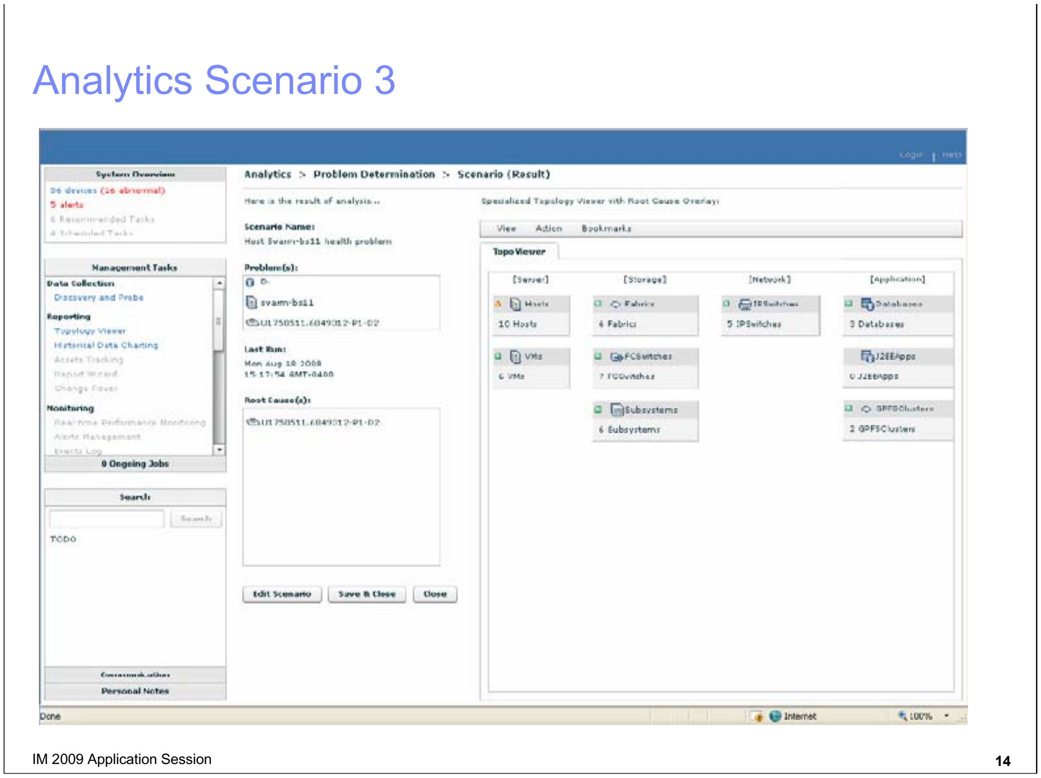
Analytics Scenario 2b



IM 2009 Application Session

13

For the sake of exploring this scenario further, if the system administrator needs to upgrade two fiber channel switches — a Brocade and a QLogic. She can use our tool to check if replacing both at the same time is feasible. The result is shown in Figure 7. As we can see, this environment does not have enough path redundancy to withstand two disconnected fiber channel switches. Many entities are impacted as a result.



5.2 Problem determination

In a large enterprise environment, at any particular time, there could be hundreds or even thousands of warning/error messages. When their number goes beyond a certain threshold, their usefulness diminishes as it becomes more difficult for system administrators to differentiate what are important and what not. From our experience, many warning/error messages are actually related to one another. For example, a SAN switch having a port problem will cause additional warnings/errors reported from other entities such as remotely mounted file systems that have this port in its data path, and the server that uses this file system, etc.

Error suppression capability we have built in DataGraph takes into account of the relationships between managed software and hardware entities and organizes potentially related warnings/errors in a hierarchical order, so system administrators are only presented with the first-level warning and error messages, where the true root cause is likely to be located. This allows system administrators to concentrate on the real root 1This is not a realistic scenario as taking two switches offline even with enough path redundancy is likely to cause severe performance problems. We use this example purely to demonstrate the flexibility of our tool. cause of problems, and solving which will usually eliminate multiple warning/error messages.

In the figure we show a screen shot of DataGraph performing error suppression. In this example, we have 3 warning/error messages, one from the host machine, Swarm-bs11, one from its file system, D:, and one from a storage volume U1750511-68493212-P1-D2. Error suppression allows us to identify the real cause of these error messages, which is the problem reported on the storage volume.

This heuristic will not always work. For instance, in the example shown in Figure 8, the problem that the file system D: is having might not be caused by the storage volume even though the file system is mounted from there. This capability is not meant to be 100% accurate. Rather, it is intended to be used by system administrators to filter out those secondary or tertiary warning/error messages. It is particularly useful when there are many such messages.

Conclusions

- DataGraph is an end-to-end system management tool
  - Integrates traditional management tools from different layers of an IT stack at the data level
  - No modifications are needed to the existing management tools
  - Provides a holistic view of the entire management environment
  - Provides a single unified user interface
  - Provides impact analysis and problem determination capabilities

IM 2009 Application Session 15

Conclusion:

In summary, DataGraph is an end-to-end management tool that integrates traditional management tools from different layers (application, server, network, and storage layers) of an IT stack at the data level, so that no modifications are needed to the existing management tools. This seamless approach to integrate existing management tools provides a holistic view of the entire management environment to system administrators. Under a single user interface, it shows how variety of different types of elements are connected and related. More importantly, impact analysis and problem determination capabilities are also provided, which would not be possible using traditional management tools.

References

[1] CIM: Common information model. http://www.dmtf.org.

[2] EMC controlcenter family. http://www.emc.com/products/family/controlcenter-family.htm.

[3] HP insight. http://h18013.www1.hp.com/products/servers/ management.

[4] HP openview. http://welcome.hp.com/country/us/en/prodserv/software.html.

[5] HP storage essentials. http://h18006.ww1.hp.com/storage/ software/srmgt/index.html.

[6] IBM director. http://www-03.ibm.com/systems/management/ director.

[7] IBM system storage. http://www-03.ibm.com/systems/storage/disk/ds8000/index.html.

[8] IBM websphere application server. http://www-306.ibm.com/software/webservers/appserv/was.

[9] SNMP: Simple network management protocol. http://www. ietf.org.

[10] Tivoli network manager ip edition. http://www-306.ibm.com/software/tivoli/products/netcool-precision-ip/.

[11] TPC: IBM totalstorage productivity center suite. http://www-03.ibm.com/systems/storage/software/center/index.html.

[12] Workload manager. http://www-03.ibm.com/servers/eserver/zseries/zos/wlm.

[13] L. Badri, M. Badri, and D. St-Yves. Supporting predictive change impact analysis: a control call graph based technique. In APSEC conference on Software Engineering Conference, 2005.

[14] B. Breech, M. Tegtmeyer, and L. Pollock. Integrating influence mechanisms into impact analysis for increased precision. In 22nd IEEE International Conference on Software Maintenance, 2006.

[15] M. K. A. et al. Problem determination in enterprise middleware systems using change point correlation of time series data. In Network Operations and Management Symposium, 2006.

[16] S. G. et al. Evolution of storage management: Transforming raw data into information. In IBM Journal of Research and Development, July/September 2008.

[17] L. Huang and Y. Song. Precise dynamic impact analysis with dependency analysis for object-oriented programs. In 5th ACIS International Conference on Software Engineering Research, Management & Applications, 2007.