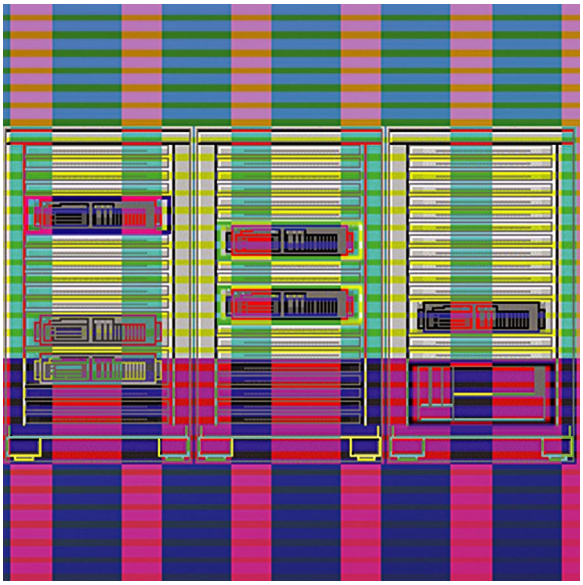
Technical forum



Management of application complexes in multitier clustered systems

Clustered systems are often used as a platform for running different services that compete for the same server resources. For e-commerce applications, these services are provided by different software components, such as load balancers, Web servers, application servers, and database servers. The software components are organized in a multitier architecture and cooperate to provide the e-commerce service. The two main challenges involved in the management of these services are life-cycle management and management of capacity-on-demand. Life-cycle management consists of deploying, monitoring, and maintaining the healthy operational state of the e-commerce service. This requires an understanding of the software dependencies, the management capabilities of the involved components, and their intertier relations. To date, most of these management tasks were performed manually (and hence were error-prone) and there was no holistic view of a service, which led to a significant increase in the total cost of ownership. Managing capacity-on-demand poses another challenge as the workload patterns of many services, especially commercial Internet services, display significant variance. The inability of current management systems to provide dynamic allocation of servers to such services leads to inefficient utilization of the cluster.

This paper describes the Raquarium project, which is aimed at meeting these challenges. Raquarium introduces the notion of "application complex," an entity that significantly simplifies the life-cycle management of services as well as the management of capacity-on-demand. It provides IBM customers with the ability to allocate resources to xSeries* servers in response to changes in load conditions and the allocation properties specified by the customer. Raquarium is integrated seamlessly with IBM Director,[1] the xSeries[2] systems management product. Provisioning supported by Raquarium in the initial release is restricted to configuration of preprimed servers, that is, those preloaded with operating systems (OSs) and applications. Future releases will take advantage of remote deployment capabilities to supply the OS and application images on "bare metal" servers, as well as configure network security using VLAN (virtual local area network).

During the last several years, a few commercial products have come to the market offering provisioning

support. For example, Jareva Technologies, Inc. offers BladeForce Suite, OpForce Suite, and Elemental Server as software products that support deploying of OS and application software and configuring of networks using VLAN.[3] ThinkDynamics Inc. offers ThinkControl Suite, which automates allocating of server and network resources in fulfillment of service level objectives.[4] Terraspring, Inc. provides Control Center, a browser-based user interface that enables an operator to design server farms comprising server, network, and storage resources.[5] The design is captured as an FML (Terraspring's Farm Markup Language) document for later configuration of actual physical resources. These products support multiple OS and hardware platforms.

The Raquarium project stems from the Océano project[6] (the name "Raquarium" was inspired by the "rack"-mounted servers and "acquarium" as carrying ideas from Océano). Raquarium's generalized application management framework and its integration with IBM Director was beyond the scope of Océano, a proof-of-concept prototype.

Application complexes and their management

An *application complex* is composed of multiple tiers, where each tier contains servers that run a specific application component. A typical three-tier application complex may consist of a load balancer, a number of Web servers and a number of application servers. The application complex entity presents a view in which the inter-relations and dependencies, the deployment properties, and operational characteristics of the components are hidden. Raquarium provides the following life-cycle management functions for application complexes:

- Automatic deployment—configures the participating servers to work together, each with its specific application component and role.
- Performance monitoring and analysis—is based on the structural knowledge of the entire application complex.
- Automatic server allocation and configuration—provides capacity-on-demand management and hardware failover support.

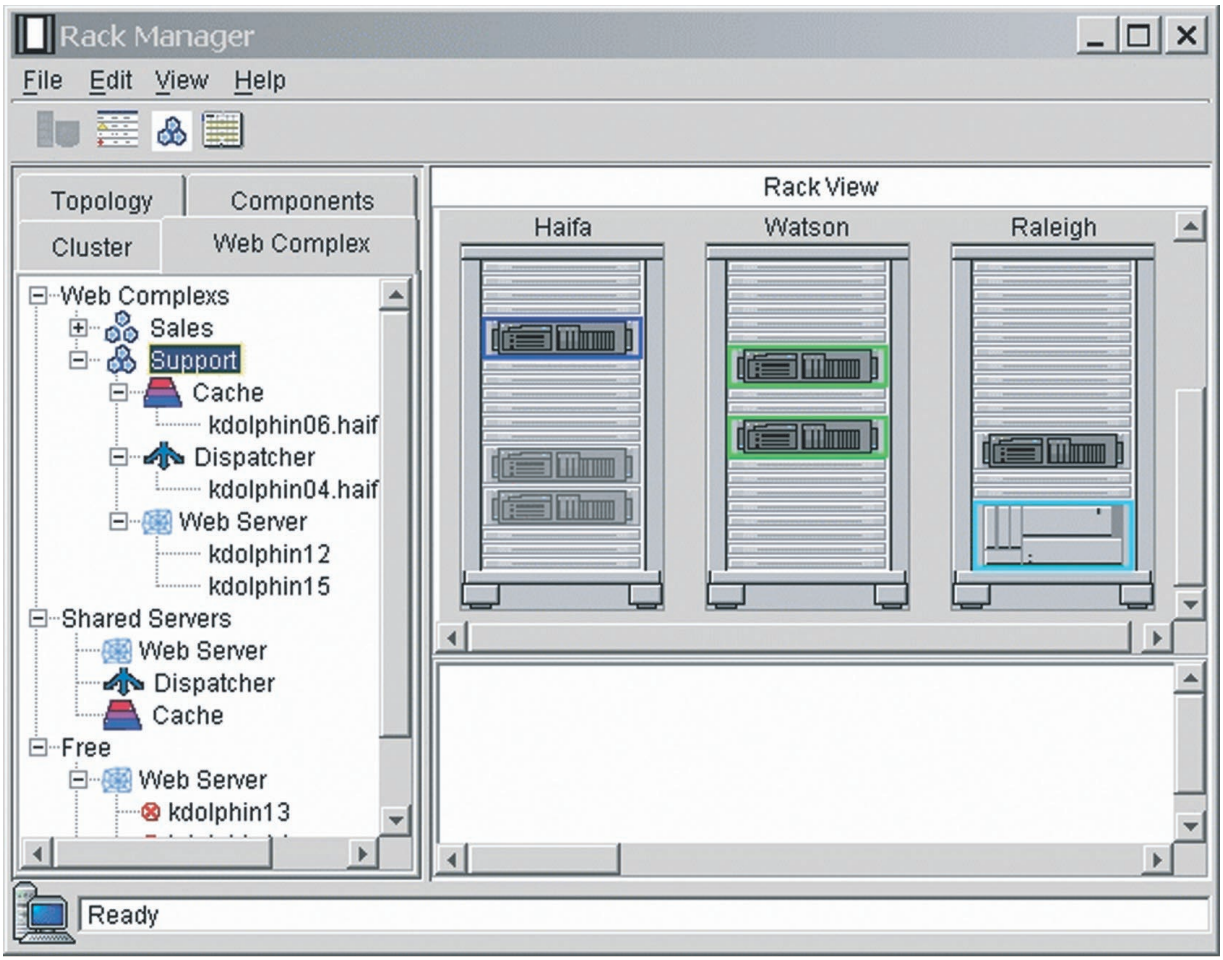
The initial release of Raquarium provides no network isolation such that access between servers within a cluster is not restricted. This is not a problem as long as the entire cluster is owned by a single customer. The initial release also restricts provisioning to preinstalled applications. Future releases will provide OS and application deployment based on image repositories, and also application complex isolation based on VLANs in order to support multiple customers environments.

Raquarium is implemented as a management framework (a container-style implementation) where application complexes are introduced as "plugins" that provide the specific wisdom of their application complex type, by implementing a "Configuration Provider Interface." The Configuration Provider plugin defines the structure of the application complex, including the structure of the tiers, the application component in each tier, whether it is scalable by adding instances of the applications, and whether it is shareable with other application complexes. The plugin also identifies the properties that require the administrator to supply input values (e.g., threshold values for controlling events). It also collects and publishes monitoring data selected by the administrator via the management framework, generates status events, and issues requests to add or remove servers. The management framework (or simply "framework") provides the visualization GUI (graphical user interface) of application complexes and clustered servers, and manages the free server pool and the allocation of servers. Finally, it applies a power management policy on the free pool such that the servers will be powered off until they are needed.

As an example, consider a two-tier application complex in which the first tier consists of an IBM e-network dispatcher that acts as load balancer, and multiple Web servers comprising the second tier. When the plugin detects an excess load on the Web servers, it requests an additional Web server from the framework. If the framework determines that an available Web server can be selected (from the free pool), it then calls the plugin to perform the necessary configuration actions. These may include updating the network dispatcher IP (Internet Protocol) tables, provisioning the correct content on the Web server, and so on. The plugin would also be called to perform the necessary configurations if the same scenario is initiated by a user operation such as dragging a Web server icon from the free pool onto an application complex view.

Raquarium provides an SDK (software development kit) to assist in the development of application complex plugins that meet the Configuration Provider Interface. These plugins can be managed by the Raquarium framework to provide a consistent user

Figure 1 WebComplexes on the IBM Director Rack Manager task



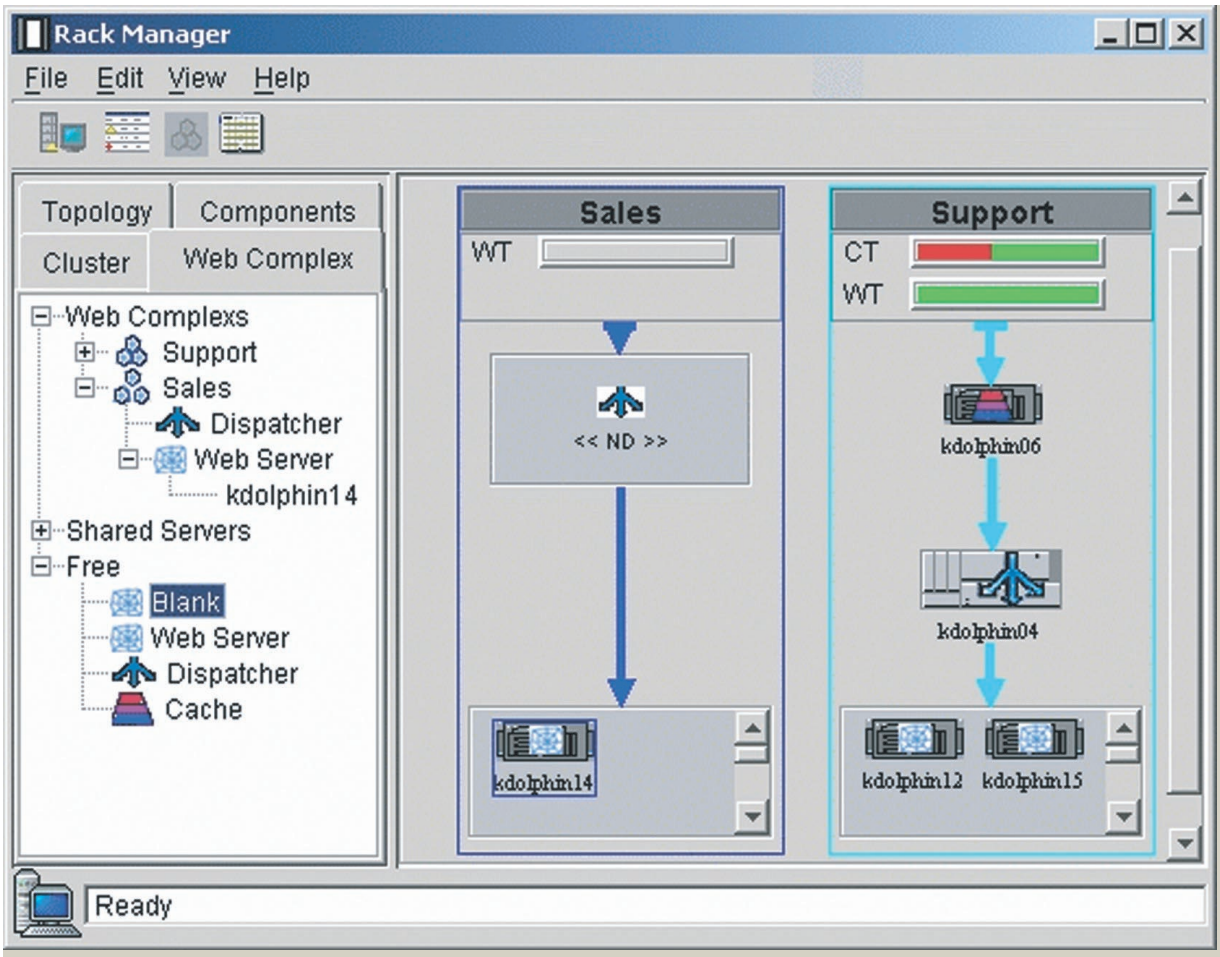
experience for different types of application complexes. The plugins are loaded during initialization time and cannot be changed dynamically.

Integration of Raquarium into IBM Director

The Raquarium prototype is integrated into the IBM Director (Reference 1 provides an overview of the IBM Director architecture). IBM Director is an Enterprise Management System that provides management functions (e.g., performance monitoring, inventory queries, and process information) for individual Intel-based servers, as well as the ability to manage racks and server blade enclosures. The integration of Raquarium into IBM Director provides a new view of the multicomponent services and the ability to perform multiserver and multitier application complex configurations, using single drag-and-drop operations. In addition to this function, the administrator can use the same GUI to perform the standard management of any individual server that participates in the application complex.

Currently, Raquarium supports application complexes, called WebComplexes, that belong to the WebSphere* Edge Server domain.[7] The WebComplexes were added to IBM Director as a new generic managed object class. The Raquarium framework is integrated into the rack-manager component of the IBM Director. Figure 1 shows the association between the physical view of the servers (their location inside the rack enclosure) and the logical view

Figure 2 WebComplexes graphical view



(the application complex to which they belong). The tree panel on the left presents the logical tier structure of each of these application complexes. Specifically, it shows two application complexes: Support and Sales. Selecting the Support WebComplex on the left panel causes the participating servers to be highlighted on the right panel. The free servers (i.e., servers not allocated to any application complex) are presented according to their role under the free pool branch.

The graphical view of the application complexes is shown in Figure 2. This view provides the tier-structure visualization of the application complexes, along with the performance monitors, and supports drag/drop operations on servers. In this example, the Sales application complex is "nonoperational" since its network-dispatcher tier is not populated. The scenarios described below help explain the management and automation capabilities provided for application complexes, along with main interactions between the administrator, the framework, and the Configuration Provider (plugin). The scenarios refer to a simple two-tier application complex, such as Sales in Figure 2, composed of a network dispatcher and a tier of Web servers.

- Administrator creates a new application complex. First, the framework queries the Configuration Provider associated with the specified application complex type for the required properties and pops

up a properties dialog window. The properties can be classified into the following types:

- Configuration parameters: These parameters enable the Configuration Provider to apply all the required configurations when a new server is added or removed. Examples are the virtual IP address to be used by the e-Network Dispatcher[8] and the Web content source for the Web servers.
- Performance thresholds: These values are used by the Configuration Provider to determine the performance status of the application complex, and as triggers for automation actions. An example is: maximum average throughput of a Web server.
- Automation actions: These are parameters (including approval) of actions to be applied as a result of changes in performance states. An example of this would be dynamic allocation of a Web server from the free pool to the application complex, when its throughput threshold is exceeded.

Next the framework queries the Configuration Provider on the structure of the tiers and the monitors, and presents the visualized model for the application complex.

- Administrator configures a Web server. Consider an example where the administrator drags a Web server from the free pool into the Web servers tier of the application complex. The framework calls the Configuration Provider to perform all the required configurations to cause the new server to participate. In our example, these would include reconfiguration of the network dispatcher to dispatch requests to this new Web server as well. It would also include reconfiguration of the newly assigned Web server with the virtual IP of the network dispatcher and configure the Web content on the new Web server. Similar reconfiguration occurs when the administrator drags the Web server from another application complex. In addition, the Configuration Provider of the source application complex is called to perform the configuration following the removal of its Web server.

- The Configuration Provider changes the status of the application complex to one of the non-OK values. The framework provides a status indication on the application complex object and fires an event into the IBM Director event system. The administrator can define an action plan for this event using any of the IBM Director built-in actions (e.g., message pop-up to the operator, mail, paging). The framework is flexible such that different action plans can be defined for different application complexes.

- The Configuration Provider detects that a threshold for allowed automation was crossed. Certain automation actions may be performed by the Configuration Provider without framework interaction (e.g., restart a participating server; see software rejuvenation description in the next section). If the Configuration Provider requires additional server resources, it will request them from the framework. If the framework determines that a server can be added (from the free pool), it calls the plugin to perform all the required configurations as in the manual drag/drop case. Otherwise, the framework fires a "no free server" event into the IBM Director event system, which can be connected to the event action plan as above. In this way, the administrator can choose not to be notified as long as servers can be allocated from the free pool to handle the performance loads.

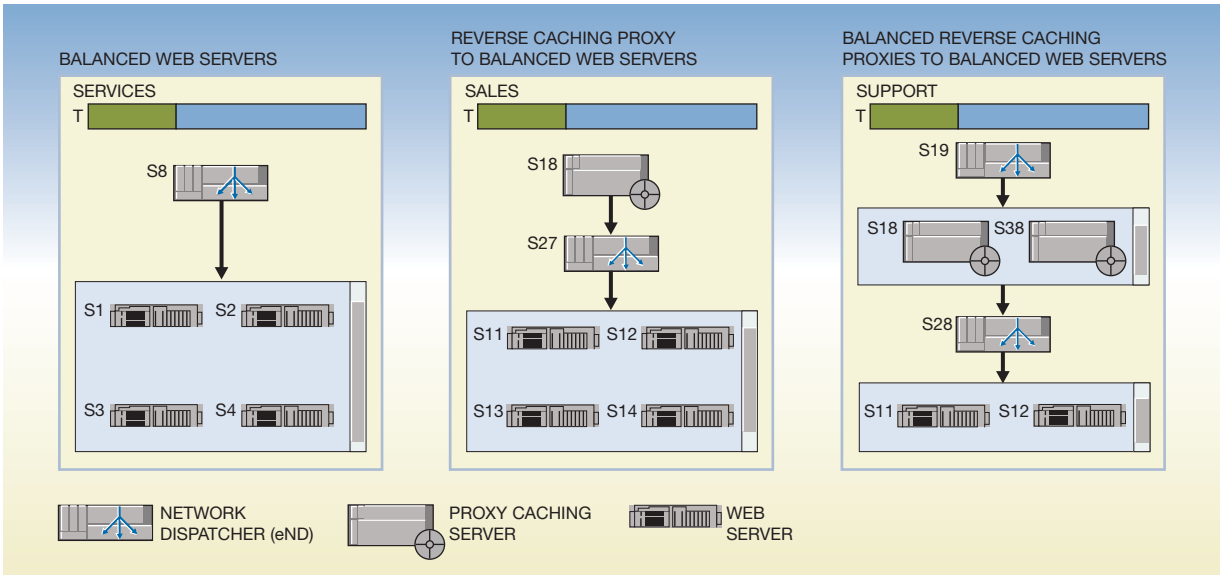
- The administrator chooses to discover existing application complexes. This is useful at a new customer site, where the actual configurations are imported into IBM Director. The framework calls each of the installed Configuration Provider plugins to discover existing application complexes of the type it supports. The administrator will be required to edit the properties of the automatically created application complexes and specify properties that could not be captured automatically.

Figure 3 shows an example of supported WebComplex types on the IBM WebSphere Edge Server domain. The first WebComplex consists of Network Dispatcher sending HTTP (HyperText Transfer Protocol) requests to the Web servers. The second consists of a three-tier application suite consisting of a cache that redirects cache misses to the dispatcher. The dispatcher then forwards the requests to one of the Web servers. The third consists of a four-tier complex where the dispatcher redirects the requests to caching proxies. The caching proxies send the cache misses to a dispatcher, which then redirects them to Web servers.

Autonomic computing aspects of application complexes

According to the IBM Research Autonomic Computing Manifesto, an "autonomic system will need detailed knowledge of its components, current status,

Figure 3 Supported WebComplex types on the IBM WebSphere Edge Server domain



ultimate capacity . . . It will need to know the extent of its 'owned' resources, those it can borrow . . . "[9] The application complexes introduced by Raquarium can be viewed as distinct autonomic computing entities that live together on the cluster. Each of these entities knows only its own internal structure and operational characteristics (this knowledge is represented by the Configuration Provider plugin), yet they all interact with a central framework for the utilization of cluster systems. The Configuration Provider knowledge enables performance analysis of the application complex as a whole, identifying bottlenecks, failure, and overload conditions, and it even predicts potential failures (based on resource exhaustion identification, described below). Self-healing in these situations is achieved by interacting with the framework to move additional cluster components of the required role. All the required configurations to make the newly assigned components participate in the application complex are done automatically, based on the plugin's knowledge (self-configuration on the application complex level).

Software rejuvenation is a good example of self-healing in managing application complexes. Software resources on a system may become exhausted due to extended use. The exhaustion of these resources can be detected by analyzing the utilization trend of system parameters such as CPU, memory, memory pools, file system utilization, and so on. Monitoring agents deployed on the systems collect these parameter data over time and construct a predictive model of utilization. One can obtain a reasonable estimate of the time to exhaustion by extrapolating the data and comparing the data with a threshold (for details see Reference 10).

The Configuration Provider can offer software rejuvenation as an automation action. When it is determined that the threshold may be exceeded, the resource needs to be shut down and restarted. Knowledge of application topology is used to gracefully shut down the resource by first stopping the requests for the resource. In our application complex, if the resource is a Web server working behind a load balancer, an event can be generated to configure a spare system as a replacement for the problem system. Once the replacement system is operational, another event will be generated to inform the load balancer to stop sending further requests to the old Web server. After a certain amount of time, it can be ascertained that all the remaining inflight requests have been completed and the resource (or in many cases the entire physical system) can be allocated to the free pool. Alternatively, if there are multiple Web servers for a customer behind the load balancer and the workload is not extremely heavy, stopping the requests to the problem system and restarting may be sufficient.

Summary

In this Technical Forum article, we describe a generalized management framework for application complexes in multitiered cluster environments. The application complexes can be viewed as distinct autonomic computing entities that live together in the cluster. The management framework was implemented on a rack-mounted cluster of servers using the IBM Director management tool.

*Trademark or registered trademark of International Business Machines Corporation.

Cited references

1. IBM Director, IBM Corporation (2002), http://publib-b.boulder.ibm.com/residents.nsf/0c11ca9140c325cc85256ad1005c9063/8d06e76dd1c1ee9f85256c320054adc4?OpenDocument.
2. e-Servers from IBM, IBM Corporation, http://www.pc.ibm.com/us/eserver/xSeries/.
3. Jareva Technologies, Inc., http://www.jareva.com/.
4. Think Dynamics Inc., http://www.thinkdynamics.com/.
5. Terraspring, Inc., http://www.terraspring.com/.
6. K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt, M. Kalantar, S. Krishnakumar, D. P. Pazel, J. Pershing, and B. Rochwerger, "Océano—SLA Based Management of a Computing Utility," *Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management*, IEEE, New York (May 2001).
7. Websphere Edge Server, IBM Corporation, http://www-3.ibm.com/software/webservers/edgeserver/.
8. IBM WebSphere Performance Pack: Load Balancing with IBM SecureWay Network Dispatcher, IBM Corporation, http://publib-b.boulder.ibm.com/Redbooks.nsf/9445fa5b416f6e32852569ae006bb65f/720d98cfa1a26f04852567e0006116f6?OpenDocument.
9. P. Horn, *Autonomic Computing: IBM's Perspective on the State of Information Technology*, IBM Corporation (October 15, 2001); available at http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf.
10. V. Castelli, R. E. Harper, P. Heidelberger, S. W. Hunter, K. S. Trivedi, K. Vaidyanathan, and W. P. Zeggert, "Proactive Management of Software Aging," *IBM Journal of Research and Development* **45**, No. 2, 311–332 (March 2001).

Accepted for publication September 18, 2002.

Antonio Abbondanzio
IBM Server Group
Raleigh, North Carolina

Yariv Aridor
IBM Research Division
Haifa Research Laboratory
Haifa, Israel

Ofer Biran
IBM Research Division
Haifa Research Laboratory
Haifa, Israel

Liana L. Fong
IBM Research Division
Yorktown, New York

German S. Goldszmidt
IBM Research Division
Yorktown, New York

Richard E. Harper
IBM Research Division
Yorktown, New York

Srirama M. Krishnakumar
IBM Research Division
Yorktown, New York

Gregory Pruett
IBM Server Group
Raleigh, North Carolina

Ben-Ami Yassur
IBM Research Division
Haifa Research Laboratory
Haifa, Israel