

# Trends and paradigm change in ERP datacenter management

Selmeci Attila, Orosz Tamás Gábor

In today's world each of the companies use ERP (Enterprise Resource Planning) systems, except the smallest micro companies. In a retail, utility, manufacturing or other bigger company (or even institute) the ERP is not the only system, but they have at least some core business solutions, warehousing system or some components of the ERP II model (SRM, CRM, SCM, PLM, etc.), and a business analytic solution. These systems are communicating with each other. Some of them are front systems communication with end-users, or customers, others only do the daily work in the background. These systems should be managed, controlled and maintained correctly. Some companies build their own data centers, others place their equipment into external datacenters, some of them outsource their application management as well, and the newest ones use virtual environments, services, infrastructures from the cloud. Each of the possible datacenters should be well designed, well organized, and well managed. The management frequently stops at the edge of the hardware infrastructure. It means only monitoring, alerting, some time information about system / application failure or stop. The business requires a bit more; it wants good performance, always-available services, flexible infrastructure on logical and physical level as well, improved IT costs, etc.

The ERP (Enterprise Resource Planning) systems (and generally the others as well) as most of the bigger applications are built according to client-server architecture. The software-oriented view of the architecture generally connects three parts of the system: database layer, application layer and presentation layer. According to newer ERP or business application the application layer is mainly a web server with special content, or portal server. The older applications, like SAP have a huge engine with its own interface to GUI (Graphical User Interface). The presentation layer for the older application has thick client or sometimes web based thin client as well, but the newer ones are more filigree by implementing the newest technologies. All of them are using database layer, but the technique and content may differ. The hardware-oriented view of ERP systems' client-server architecture shows us different level of infrastructures. In each one we can meet network elements and active components, storage layer and computing elements (servers, virtual machines, zones, etc.). In this architecture not only one-one connections are available, but one-many as well. As an example on the software-oriented view we can have one database and many applications servers, or many clients connecting to one application server. On the hardware-oriented view an example for 1-n connection is the physical host, which can be mounted into a rack or blade center, or one storage area offered via NFS (Network File System) can be attached to several operating systems. Many-many connections can be also found in software layer if we would handle parallel queries, e.g. using Oracle RAC (Real Application Cluster) with multiple clients. The software and hardware-oriented layers are not strictly separated, because the modern hardware elements can be controlled, managed by software tools. These tools offer more and more features for the given hardware elements. A well-known example can be a hypervisor based virtualization environment, where many (n) physical hosts can run many (m) virtual machines.

Each part of this high level conglomerate can be handled as an object. We should think of them as composite object, because they are not simple ones, but contain many smaller, important parts (just like a storage contains the single HDDs, SSDs, PCI Express caches, or even internal operating system, etc.). The objects have special relationships via their connections to each other. All these lead us to think about an object oriented model of a datacenter. The model describes the infrastructure element and the software elements as well containing their properties and statuses as well. The model is the first step to the new management, because it only describes the datacenter, the used infrastructure and the software, applications and services running on the infrastructure.

Our goal is to build a management tool using the object model to flexible maintain the software defined elements, monitor the environment, alert on request and proactively offer tasks or even actions to be executed automatically. The object model enables these proactive tasks by having events and handlers for them. A very simple example can be well during the night a huge data load is started into the data warehouse system without previously notifying the operators. The operating system based host agent sends data to the object, which checks the predefined thresholds. If there is not enough place is available an event is raised to notify the storage object to allocate more place to the used volume. Of course the management tool should notify the operators as well about the emergency volume resize function. In the first level we try to monitor, control and manage services like database, application instance. We implement and adaptive solution, which runs services elastically distributed on physical or virtual hosts in the environment. Parallel we would refine, refactor the object model to converge our abstraction to the real datacenter architecture, elements and their links to each other. Our future plan is to collect and analyze IT data, because the great power in such a solution is not only the monitoring and automation, but also the analytic and the behavior changes according the analysis.