



zimory
going beyond...

February 2011

zimory®scale White Paper

Relational Databases in the Cloud



Target audience

- CIO/CTOs/Architects with medium to large IT installations looking to reduce IT costs by creating a more flexible platform for customizing data management.
- Those looking into a managed and safe migration path to VMs and cloud-based data management infrastructures.
- Companies in need of more flexible deployments and migration solutions towards virtualization and cloud computing.

Overview

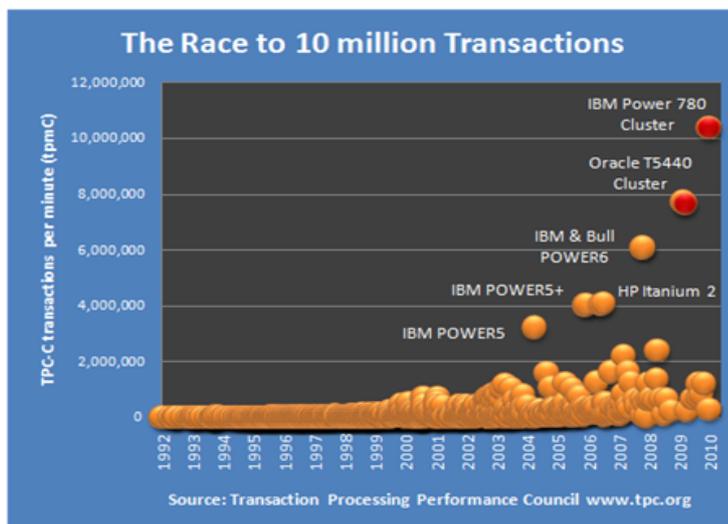
Many of the advantages of cloud computing and virtualization apply solely to the upper layers of the software stack. For those applications that rely on relational databases, features like elasticity, flexible deployment, or reduced capital expenses through the use of cloud services are severely limited. This document discusses the problem in detail and shows how **zimory@scale** can be used to extend relational engines to provide better scalability, more flexibility in the deployment, and the ability to deal with virtualized and cloud environments.

Introduction

Relational databases are the cornerstone of enterprise applications as there's no real alternative in terms of performance, transactional guarantees, and recovery procedures.

Contrary to what many people claim, existing engines are relatively scalable and offer substantial performance for many applications. The reported performance of commercial engines on existing benchmarks proves this to be the case (Figure 1); with the latter years having seen systems that reach several million transactions per minute.

Figure 1: Relational database engines and their performance over time



Besides the performance, relational databases are tried and tested products that many companies have learned to trust and optimize for their applications. Even if there were technical solutions for database users, providing similar functionality and quality (which is still not the case today), the investment in software build for relational databases is so large that a short or medium term migration to other platforms is simply not cost effective. Any improvements to IT infrastructure relating to performance or cost criteria must include relational databases, in order for it to be a realistic and truly effective solution.

Databases, virtualization and the cloud

Useful as they are, existing relational database engines are based on architectures and designs that are decades old. As a result they are facing considerable difficulties in adapting to the ongoing changes in the IT world.

Virtualization, multicore, and cloud computing are somewhat independent developments that bring substantial changes to the hardware where a database runs. Together, they constitute a formidable challenge for relational database engines and their prevailing business models. Database engines differ from one another in many ways, but the problems they face are common to all.

Multicore architectures introduce a degree of parallelism that in many cases require an extensive redesign of the database engine. It's very difficult to imagine how the architecture of existing commercial and open source relational databases will scale to hundreds of cores, let alone the thousands that are already being considered in many new hardware designs.

Virtualization poses the problem of runtime reconfiguration of key engine parameters. Although few studies exist, it's often assumed that database engines can be virtualized, perhaps with a small performance loss but without affecting functionality. Once virtualized, it's possible to migrate a virtual image containing a database. However, engines often make configuration decisions (such as, how many cores to use for running the engine itself or the sizes of memory buffers, etc.) at start time. Thus, runtime migration cannot be used to efficiently adjust the resources available to the engine unless the engine is restarted.

Cloud computing requires both elasticity and flexibility in deployment. Existing database engines are definitely not elastic, not even in the most lenient interpretations of elasticity. Some degree of elasticity can be achieved through different mechanisms but almost always within a fixed configuration that cannot be changed dynamically. In terms of flexibility, virtualization could be a way to go but, as explained above, it cannot be efficiently used to dynamically assign more or less resources to a database engine as load and need dictates a basic premise of cloud computing.

The solution offered by commercial database engines to these problems and limitations is well known: significant over-provisioning of the database engine to cope with peak loads, future growth in demand, and even to support the elasticity of the applications running above the database. This approach fits the business model of database vendors very well. Database vendors understand this and price database engines accordingly; aware that the demand for reliable ways to manage data is increasing every day and there are not many alternatives available.

The immovable application stack

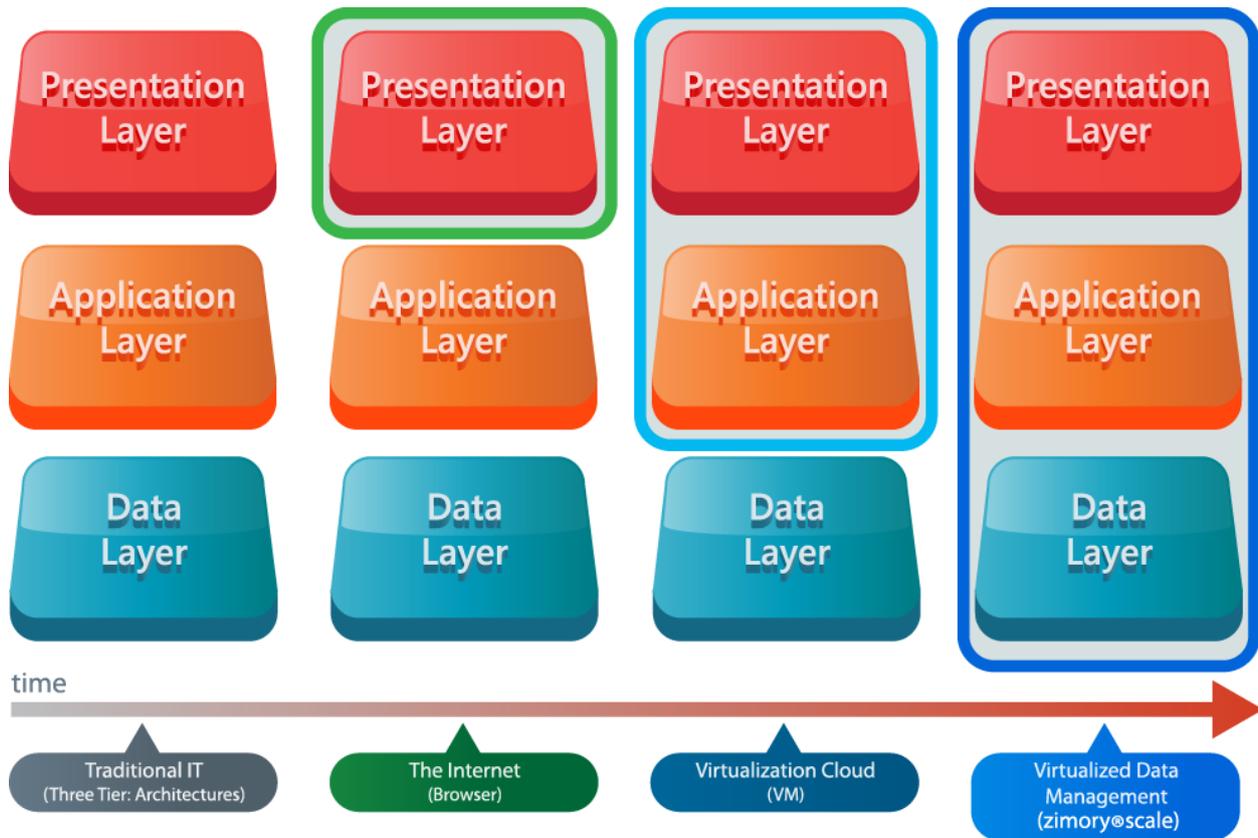
Operating even a simple IT infrastructure, with an industrial strength application server and a commercial database is a complex and expensive affair. Licensing costs are high; the possibilities to tailor the system with small and necessary extensions will immediately lead to higher costs and less flexibility. Not to mention the personnel cost related to administration and maintenance. Virtualization and cloud computing are being proposed as the solution to such problems. However, experience shows that often savings on one side of the system are smaller than the cost increases resulting from the new configurations. In many cases, the underlying platform and supporting software needs to be hugely over-provisioned. The main reason for this situation is the lack of flexibility on current software stacks and the need to resort to complex workarounds to deploy systems tailored to the problem at hand.

The major reason for the lack of flexibility in today's deployments relates to the data management layer. While the presentation and application layers have been gaining in flexibility, scalability, and configurability; data management layers implemented as relational databases remain monolithic solutions. Of the entire software stack, the only part that cannot run on large clusters, be easily reconfigured, be extended using commodity components, and be virtualized; is the relational database at the bottom of the stack.

Figure 2 below shows the evolution of multi-tier architectures over the last two decades. From the introduction of three tier architectures, every step in the evolution of enterprise computing has involved adding flexibility to the tiers. The presentation layer was virtualized in a way through the use of web browsers. Web browsers eliminated the tight coupling of client and servers and enable having clients for a system anywhere on the Internet. The next step was the virtualization of the application layer. Virtual Machines separated the application logic from the underlying hardware and opened up the door for cloud computing, where the flexibility provided by virtualization is taken to an extreme.

The natural next step in the evolution of technology is the virtualization of the data management layer: provide databases with the same degree of flexibility, scalability, and loose coupling that is possible for the presentation and application logic layers. This is the step that Zimory has taken with its **zimory@scale** architecture.

Figure 2: The evolution of multi-tier architectures

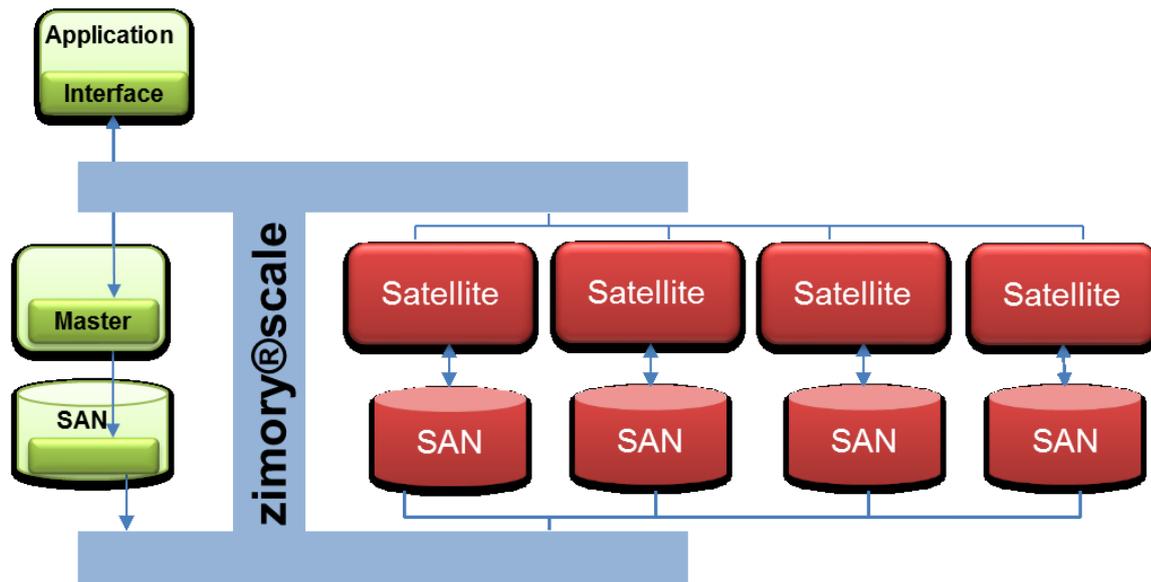


Model architecture and components

zimory@scale offers a solution that extends beyond existing relational database engines in ways that provide elasticity, scalability, and increased performance at a lower cost. It also has the possibility to combine databases with virtualized and cloud based environments. **zimory@scale** provides a new degree of extensibility and customization of database platforms.

As shown in Figure 3, the starting point for the Database Hypervisor architecture is a *master database*. This is a possibly pre-existing, relational database engine implementing the data management layer for some application(s). Through interception of the traffic between the applications of the databases and capturing of the changes performed on the database, the master database can be treated as a logical unit that can be cloned, extended, migrated, and ported across different environments. These operations take place through *satellite databases*. Satellite databases are relational database engines (not necessarily the same as the master) where all the changes performed at the master are continuously and consistently applied. These satellites are then used to support the applications which are unaware that part of their load is being handled by satellites instead of by the master. Through innovative routing of requests and on the fly request translation, applications are offered a single system image, even if the satellites are implemented with an engine that is different from the one used at the master.

Figure 3: zimory@scale model architecture and components



Added value

Technically, **zimory@scale** differs from existing solutions in that it combines both SQL statement forwarding and log mining in a single tool. This combined with extensive management and deployment tools and support for virtualized storage management, **zimory@scale** goes well beyond the narrow, single system focus and aims at medium to large scale deployments.

zimory@scale is finding considerable traction among Service Providers and companies running large IT infrastructures as it has the potential to provide scalable database services at a reduced cost. Companies who have independently evaluated the impact of **zimory@scale** on their costs have found potential savings of between 30 and 60% for large database installations. This excludes the potential savings in labor and maintenance costs derived from Zimory's additional management tools.

Rather than a replication or data integration solution, **zimory@scale** offers an open and flexible platform that can be customized and tailored for a wide range of scenarios.

Unique features include:

- **Single system image:** Applications are unaware that their requests are not being processed by the master. The behavior and semantics of the system are exactly the same as those of the master database.

- **Support for homogeneous and heterogeneous settings:** The master and satellites do not have to be identical relational engines. This creates the possibility of extending conventional database installations with open source products tailored to individual applications.
- **Support for local and WAN satellites:** The satellites can be placed in the same data center as the master, in remote data centers or even in cloud based environments. **zimory@scale** can still maintain a single system image and keep all satellites consistent at all times.
- **Support for virtualized and non-virtualized environments:** **zimory@scale** makes no assumption about whether master or satellites are running inside VMs. However, the satellites, notably those based on open source databases can be run inside VMs. In these cases **zimory@scale** incorporates a sophisticated management platform for the automatic creation of images, monitoring, and bridging of data centers.

Advanced features

The openness, extensibility, and low-cost features of **zimory@scale** are complemented with an extensive set of management and support tools for controlling large IT deployments within and across data centers, virtualized environments, and cloud infrastructures (**zimory[®]connect** and **zimory[®]manage**). These tools allow users to dynamically manage their IT infrastructure regardless of where it is located: either in a local data center or in a remote Service Provider. Furthermore, **zimory@scale** enables users to bridge infrastructures by providing the functionality necessary to move entire systems across data centers, to and from clouds, and dynamically aggregate server computing capacity available in other data centers. **zimory@scale** incorporates these management tools that cannot be found in competing products as many of them predate virtualization and cloud computing. Combined with these tools, rather than being a database and data integration solution, **zimory@scale** is a comprehensive tool and development platform for the data management layer of IT infrastructures. As such, **zimory@scale** includes support for management of virtualized environments, deployment of VM images of databases inside and across data centers, support for federated clouds, sophisticated storage management including COW (copy on write) features that complement those of the database and the technical solutions discussed above.

With **zimory@scale** it's possible to dynamically create open source satellite copies of an Oracle database residing in a local data center and place those copies in remote data centers or in clouds. It also provides a single system image over such configuration, thereby giving companies the possibility to dynamically expand their resources when needed using external clouds, and thus avoiding expensive and unnecessary over-provisioning.

zimory@scale is also seen by potential customers as a safe migration path from conventional environments to VMs and eventually to the cloud. This is thanks to the seamless integration of **zimory@scale** in all three environments and the ability to extend a conventional deployment with satellites deployed in VMs or in the cloud, while still maintaining full control over the initial configuration.

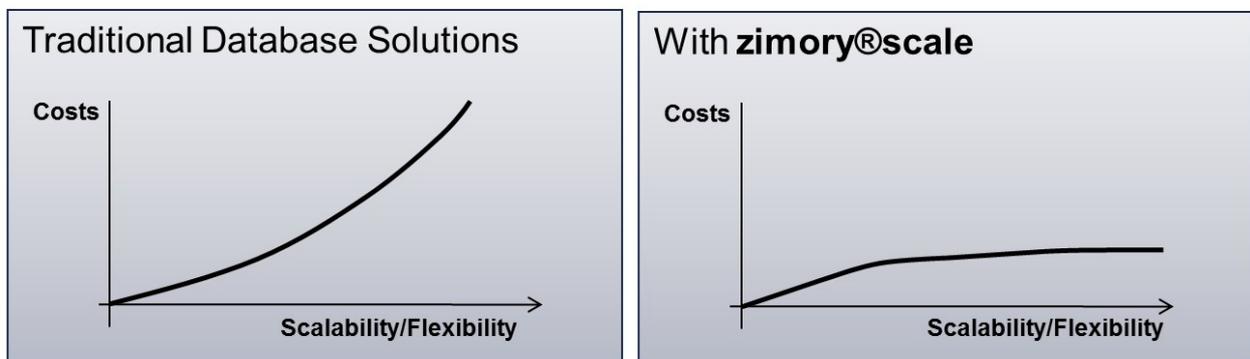
Such configuration allows the user to test VM and cloud deployments, zero-downtime migration across environments, and scalability at very low costs and with a great degree of flexibility.

Basic use cases

A number of use cases based on real deployments have shown the advantages attainable through **zimory@scale** can be significant, with an unprecedented gain in flexibility. This is in stark contrast to today's systems where customization leads to both high costs and a severe loss in generality. As systems evolve, the lack of flexibility in the deployed solutions and the underlying platforms result in even higher costs; a recurring situation during the life time of any infrastructure.

With **zimory@scale**, users can extend their legacy systems based on commercial database engines using low cost, open source databases. These databases act as satellites that can contain; complete copies of the original data, partial copies, or customized views of the data. The database hypervisor ensures that the data in the satellites is always in a consistent state and that applications see a single system image. Therefore, applications do not need to be modified or recompiled to be extended with the database hypervisor.

Figure 4: Evolution of cost as a function of scalability and flexibility of the database solution



Typical uses cases for **zimory@scale** includes:

- **Database agility** means that the database can scale on demand without significant lead time. The benefit of having database agility is twofold. On the one hand the administrator can better support applications with changing usage situations – e.g. peak loads on Monday morning. On the other hand the solution allows the administrator to reduce the over-provisioning of database resources. This leads to service improvements and cost reduction.

- **Database resource pooling** - **zimory@scale** technology supports administrators in the creation of shared pools of Database Systems that serve for a set of applications. An enterprise with multiple applications normally has dedicated database servers, each of these supporting a specific application. **zimory@scale** allows partial pooling of physical servers and shares them between all applications. This leads to significant cost reduction on licenses and hardware, as the peak-load demand can be covered by the pool and does not need to be provisioned on a per application basis.
- **Database high availability** - High and continuous availability are critical issues. **zimory@scale** allows users to create transparent failover systems with almost no performance overhead. Combining database agility with database resource pooling leads to high availability solutions where shared resources are used as standby nodes, significantly reducing the cost of such solutions when multiple databases are involved.

Conclusions

As the cornerstone of any IT architecture, relational databases offer a wide range of valuable features and functionality. However, due to both technical considerations as well as existing business models, relational engines have also become an obstacle in the race towards more flexible deployments and achieving the cost reductions promised by cloud computing and virtualization. **zimory@scale** offers a way to extend existing deployments of relational database engines with satellite databases. These satellite databases offer consistent copies of the master database for read-scalability and elasticity. With **zimory@scale**, users can leverage their investment in highly tuned and high performance relational database engines without renouncing the advantages of virtualization and cloud computing.

About Zimory

Based in Berlin, Zimory enables Enterprise Companies to transform their virtualized data centers into Cloud Services infrastructures. Zimory's "Carrier Grade Cloud Computing" product suite provides a secure, fully flexible, scalable and interoperable, end-to-end solution for private, public, hybrid and database cloud. For more information visit www.zimory.com