SQL Server Cost Recovery White Paper

Date: 27 May 2011

Version: 1.0

Microsoft GOLD CERTIFIED Partner

Specialising in building Microsoft SQL Server Solutions

Document Revision History

Version	Revision date	Author	Description
1.0	27 May 2011	Peter Ward	Initial Release

Contents

Introduction

One of the challenges facing the IT departments of many organizations today is who pays for the operational support of the SQL Server databases that are used by the business? Organizations are continually introducing new applications that assist in improving and streamlining business processes. It was not that long ago that an office looked very similar to a classroom with long rows of desks and on each desk there was an in-tray and also an out-tray. Documents would move along the row of desks from one person's out-tray to another person's in-tray. These manual business processes and workflows have now been replaced with line-of-business applications such as CRM, ERP and HR systems. Which each new application that is adopted there is a corresponding database that also needs to be managed. Hence IT departments continually have to increase the number of databases that they support without having an easy way to recover the costs associated with this additional management. In other words IT departments are continually being asked to do more with less as a result of their inability to recover the costs associated with database management.

The Context for SQL Server as a Service

The IT industry is very much like the fashion industry in that both industries are based on trends that almost change with the seasons. What was considered to be best practice or fashionable last year is often no longer the case this year. One trend that is not going away anytime soon is the concept of virtualization. Virtualization is typically an initiative by IT departments to assist in reducing the costs associated with providing infrastructure to the business. However, the problem with many virtualization projects is that whilst they reduce the hardware footprint and help to reduce other infrastructure costs such as rack space, power and cooling, they often fail to deliver on consolidating the number of SQL Server instances in the environment.

In data centers today there is a phenomenon known as SQL Server sprawl. SQL Server sprawl is the uncontrolled and uncoordinated deployment of SQL Server instances. The root causes of this sprawl are often: 1) attempting to ensure application performance and 2) the way that a business procures infrastructure and software is on a project by project basis.

As mentioned, one of the reasons for SQL Server sprawl and also one of the reasons that consolidation is not performed is that application sociability is difficult to achieve. Application sociability is the concept of applications coexisting in an environment. One of the best ways to think about the challenge of application sociability is the analogy of two children playing in a sandpit with a toy truck. Both kids want to pay with the toy truck; however the truck is a finite resource that can only be used by one child at a time. Hence it is the bigger kid that muscles in and 'steals' the toy truck so that the smaller child is only able to utilize it once the big kid has finished with it. Applications often behave the same way in that there will be one application that wants to use all of the resources on a server to the detriment of any other applications on the server. Hence the introduction of SQL Server sprawl as new instances of SQL Server are deployed simply so that application A does not affect the performance of application B.

The second reason for SQL Server sprawl is that the introduction of a new application is typically funded as a project by a particular business unit. For example if a new Payroll application is implemented then the Human Resources business unit will fund the project. If new hardware is required then the project funds this purchase along with the software including the database platform. Hence the business unit sees the hardware as theirs, as they paid for it. As a result they are unwilling to share the infrastructure as they want to ensure that the Payroll application is not effected by sharing the infrastructure with anyone else.

If we look a little closer at this pattern of procurement what we will often find is that once the project has been implemented there is no longer any funding to provide ongoing support for the SQL Server instance and hardware, let alone someone to pay for upgrades. As a result, SQL Server instances are not upgraded and are run on legacy platforms that nobody wants to pay to upgrade.

SQL Server as a Service is designed to address these issues by providing a framework that allows the number of SQL Server instances to be reduced whilst increasing the elasticity in the SQL Server topology. It also provides a mechanism to recover the costs associated with the ongoing management and maintenance of the SQL Server topology.

What is SQL Server as a Service

SQL Server as a Service is a relatively new concept and is also known as SQL Server Private Cloud. I think Shashank Pawar, who is a Microsoft Data Platform Technology Solutions Professional, provides one of the best definitions of SQL Server as a Service "SQL Server as a service/private cloud helps in optimizing your existing SQL Server environment by reducing total costs of ownership and improves the agility for business solutions that require database platforms."

One of the terms that confuse the discussions around SQL Server as a Service is that it is also referred to as SQL Server Private Cloud. When you think of The Cloud typically you are thinking about Software as a Service (SaaS) models. SaaS is a software delivery model in which software and its associated data are hosted centrally in the Internet (cloud), in other words SaaS uses The Cloud as a delivery mechanism. However the term Cloud computing actually only refers to the provision of computational resources on demand via a computer network.

As Cloud computing has no dependency on the internet it is something that can be delivered using a Local Area Network (LAN) hence the term Private. Private Cloud is based on on-premises infrastructure. The key though is that the delivery of the infrastructure creates an agile, cost-effective factory that can quickly provision and expand or collapse capacity (elasticity) based on end-user (customer) demand. That all being said, what SQL Server as Service means is that a SQL Server topology satisfies the following characteristics:

1. Provide self-service functionality to allow end-users (customers) to request new, expanded or shrink the resources they have been allocated
2. Proactively monitor server and database utilization and reallocate resources as required
3. Elastic (expand or collapse capacity), with no downtime
4. Provide self-service provisioning of new resources
5. Metering of resource utilization for charge back

One of the key tenets of SQL Server as a Service is the metering of resource utilization and the charge back associated with it. This is because cost is often one of the main reasons that IT departments look to implement SQL Server as a Service or any Software as a Service solution. The reason that cost is such a big factor is that by using the service model, capital expenditure is converted to operational expenditure. It also means that IT departments do not need to provide infrastructure for one-time or infrequent intensive computing tasks and otherwise remains idle.

An Introduction to Chargebacks

We described earlier that trends in IT are very much like fashion in that what is trendy today is no longer the case next season. However just as in fashion what is old is also new again. The same way that flares and platform shoes came back into fashion, chargeback models for computing resources are also making a return.

Charging for resource utilization is not something that is a new concept. If you cast your mind back to about the same time that flares where originally fashionable, most computing was performed on Mainframe computers. These computers were very expensive to purchase and maintain and as a result there was a need to try and recover the associated costs. This cost recovery was done by charging for the computational units that were used. The process to perform this cost recovery was relatively easy as the mainframe operating systems natively provided logging mechanisms that identified who was using the resources, which applications they used along with the ability to show the amount of CPU, memory and storage they consumed executing the process. This information was then able to be used to recover the cost of the mainframe by charging the amount of resources consumed back to an individual business unit or costs center.

A lot has changed since then and now apart from the fashion. The biggest change to IT departments is the introduction of commodity computing and the ability to purchase a computer that has 100's of times the processioning power at 100th of the price of a mainframe. As a result of the reduction in the cost of delivering computational processing it was seen that there was no longer a need to recover the costs associated with procuring the IT resources. This was also coupled with the fact that the operating systems introduced did not provide an easy mechanism to track resource usage by individual users. However despite the reduction in the cost of modern server infrastructure it is still not something that is free. There is a cost associated with purchasing a server. There is also a cost associated with the ongoing operation and maintenance of the server which is often many times more than the original cost, let alone the cost of the software to run on the server.

The issue that most IT departments face today is how to introduce a paradigm shift in the business for how they charge for computing resources. Often you will hear an end user ask the question of "why does it cost the business unit $4,000 for a laptop? I can purchase the same one for less than $1,000". This question is raised as the IT executives for the organization have introduced a tax to recover the total cost of ownership that an end user does not normally think about, such as hardware and software maintenance as well as data consumption.

The issue with this model is that it is the same as taxes that governments force individuals to pay. These taxes pay for providing public facilities such as schools, roads and health facilities, but are not a "user pays" model. So someone that does not get sick or have any children going to school and walks everywhere is being taxed even though they do not use the resources they are paying for.

Nobody likes paying taxes for things they don't use! A chargeback model is designed to provide transparency and control to end users regarding the resources they use. SQL Server as a Service introduces the concept of metering resource utilization. The implementation of this metering provides a mechanism to create a chargeback model that only taxes the end user for what they consume.

The concept of chargeback based on metering is designed so that a business unit is only charged based on what they use. This model is similar to tolling on a bridge or the tolling on a telephone call. If we think about the example of a bridge, someone makes an investment in the infrastructure to build the bridge along with capacity for future traffic needs. The cost of the bridge as well as the ongoing maintenance is then provided by the people that use the bridge. If you don't use the bridge you don't pay. The user is empowered as they know what the cost to use the bridge is and they can make decisions about when it is appropriate to use the bridge. They may make a decision to use the bridge when they are in a rush but if not, they can drive the long way to their destination.

A side effect of the tolling model is that if the end-user is aware of the real cost of what something is costing them then they can look to drive efficiencies and costs savings. For example if they are aware that the poor code in one of their reports is costing them money due to the additional computations that are required then they can choose to make an investment to have the poor code resolved and reduce the ongoing costs associated with running the report.

This level of reporting to provide a cost breakdown by process was not something that has been easily attainable in the past however a number of changes have been introduced in SQL Server 2008 R2 to assist to support chargeback mechanisms and the provision of SQL Server as a Service.

Implementing a Chargeback Model

The key thing to remember with a chargeback model, just like any solution, is that there is no one size that fits all. Each individual organization will have a slightly different way of determining what resources are charged for as well as how much to charge for them. The other key thing to remember when implementing a chargeback model is that just like building a bridge there will already be a road in place that can be used and you need to entice the users to utilize the new toll road in place of the existing road that they often see as free as they do not associate the taxes paid with the usage of the road.

The strategy for metering and charging based on consumption is to provide a cost-effective alternative to self-hosting SQL Server databases and attempt to entice business groups to migrate their current databases to a centrally managed service. In order to provide the cost benefits to the business groups it is critical that costs can be calculated and charged back to the business. There are four basic models of chargeback:

Approach	Description
Overhead	All costs are absorbed by the entire business.
Allocation of Expense	Component tracking is used to create a percentage of cost for each business unit.
Standard Resource Rates	Users are charged using a fixed rate schedule for a particular service, such as a search result or a report run
Fixed Rate	Users are charged by time on the system alone, regardless of other components.

The four basic models of chargeback

A typical chargeback model is a combination of the Allocation of Expense and Standard Resource Rates approaches. The challenge of the Allocation of Expense approach is to understand the expense of the service and its dependencies. To accurately recover costs it is important to understand the expense of offering the service. The following items are examples of costs that will be need to be defined for each organization:

- Capital expense such as hardware
- Ongoing expenses such as Software Assurance
- Support costs including both environmental and human resources
- Operational costs such as data centers, racking, power etc.
- Cost of the solution over time, what is the lifespan of the service

When defining the Standard Resource Rates there is a need to define a measurable unit of resources known as a Computational Unit (CU). For the sake of simplicity often this is defined as 1 processing unit and 1 GB of memory. A processioning unit represents what the operating system would see as a CPU, this could be a CPU core or a hyper-threaded CPU core. In addition to the Computational Unit costs the service is typically also charged based on storage usage.

Consider the following example where the Computational Unit is calculated at $1000 per Computational Unit per month and the SQL Server instance for the client has two Computational Unit assigned to it (2 CPUs and 2GB of memory).

Each month $2000 will need to be recovered (excluding storage). As the business group consists of multiple business teams with individual cost centers the approach will be to measure proportional utilization of the Computational Unit assigned to the SQL Server instance. The following is an example of this scenario

The instance has four databases, two are from cost center A and one belongs to cost center B and also belongs to cost center C:

- Database 1 and 2 are owned by cost center A
- Database 3 is owned by cost center B
- Database 4 is owned by cost center C

Database	Cost Canter	CU Utilisation %	Storage (GB)
Database 1	A	34	10
Database 2	A	5	100
Database 3	B	15	1
Database 4	C	25	50
Internal	N/A	21	75

Internal resources include unused Computational Unit utilization during the billing cycle and the storage required to maintain the service but is not directly utilized by an end user. These costs will be equally allocated to all service users as they are required to maintain the service.

In the above example the chargeback would be:

- CU cost $1000, 2 CU Total
- Storage cost $2 per GB

Cost Centre 1 will be charged 39% of the UC + (the Internal CU cost /3) + 110GB of storage + (75 GB / 3).

Cost Centre	CU Cost	Storage Cost	Total
Cost Centre A	46% = $920	135 GB x $2 = $270	$1190
Cost Centre B	22% = $440	26 GB x $2 = $52	$492
Cost Centre C	32% = $640	75 GB x $2 = $150	$790

The above example is a simplified version of a chargeback model for SQL Server as a Service. There are a number of factors for the chargeback model that need to be considered, valued and then defined for the chargeback model to be effective:

- Does a CU cost in include DR servers
- Does a CU cost include supporting environments such as UAT, Development, Sandpit and Management environments or are they a separate service
- Does storage include backups and DR copies of a database
- Does storage include supporting environments such as UAT and Development

The success of the chargeback model is dependent on the accuracy of the utilization data and complete and accurate understanding of the costs to offer and maintain SQL Server as a Service.

WARDY IT Solutions Profile

WARDY IT Solutions is a specialist Microsoft SQL Server consulting, training and Business Intelligence provider. Since its establishment in 2004, WARDY IT Solutions has assisted customers to successfully deliver SQL Server solutions ranging from database administration to Business Intelligence reporting and training. We have a comprehensive range of consulting services designed to fulfil every Microsoft SQL Server requirement for our Clients and is in the unique position as a business to have direct access to the Microsoft SQL Server product team in Redmond – USA by being a member of the Technology Advisor Group for future releases of SQL Server.

WARDY IT Solutions has established itself amongst the top solution providers in SQL Server technologies and recently we have received the following industry accolades:

- 2009 Microsoft APAC Data Management Partner of the Year
- 2009 & 2010 Finalist for the QLD ACS Award for Service Delivery
- 2010 Finalist for the QLD AIIA Award

In the new 2010 Microsoft Partner Network, WARDY IT Solutions has achieved Gold Competency for Data Platform and Business Intelligence, and Silver for Learning, and prides itself on having the best SQL Server consultants and trainers. Within our team we have recognised industry experts in Microsoft SQL Server including; several Microsoft Certified Trainers (MCT's), the authors and editors of several SQL Server books and each of our consultants are Microsoft Certified IT Professionals (MCITP's).

WARDY IT Solutions has vast experience in private and government sectors, including: Manufacturing and Industrial, Retail and Distribution, Banking and Finance, Health, Transport, Utilities, Education, and Telecommunications. Our clients range from the largest banks and stockbrokers in Australia to Australia's largest retailers; our solutions deliver results for the largest Microsoft SQL Server implementations in Australia and we built a quality and lasting relationship based on trust and mutual respect with our Clients.

"I have been working with WARDY IT Solutions for over 18 months and their SQL Server skills have proved invaluable. I consider myself fortunate to work with such a talented and devoted organisation."

- Morty Douglass, RD8

"Our relationship with WARDY IT Solutions has become a crucial aspect to the success of our team. The level of excellence they provide in service and technical support has always been exemplary. "

- Roberto La Bozzetta, Coles Group

Microsoft Partner Program 2009 AWARD WINNER Data Management Solutions Partner of the Year

Microsoft Partner
Gold Data Platform
Gold Business Intelligence
Silver Learning