OPINION

Counting database costs: inflexible databases are unaffordable databases

Elastic database systems address multiple needs with cloud-native and container-native designs that allow them to scale-out and scale-in dynamically.

Max Pixel (CC0)

> “Movement and change are the essence of our being; rigidity is death.” —Virginia Woolf

In the previous post we talked about the mega-trend towards elasticity in the enterprise software stack, and how the database layer is not keeping pace with developments in the web, application and storage layers. We also asserted that this limits business agility and constrains innovation. Most importantly we referred to the serious cost implications of an inflexible database layer. In this post I will set out some of the causes of these uncontrolled costs.

The unavoidable costs of inflexibility

Why are there radical economic benefits of elastic databases? There is a lot of detail we could talk about, but none of the headlines are surprising - they are the driving forces behind the growing trends such as microservices architectures and container-based deployments.

The main themes are as follows:

[Beware the 9 warning signs of bad IT architectureand see why these 10 old-school IT principles still rule. | Sign up for CIO newsletters.]

1. The **capacity-related costs** of inflexible database systems will always be higher than the capacity-related cost of elastic database systems.

2. The **downtime costs** of inflexible database systems will always be higher than the downtime costs of elastic database systems.
3. The **system-administration costs** of inflexible database systems will always be higher than the systems-administration costs of elastic database systems.

I will lay these out in greater detail below, but the TL;DR explanations of these three observations are:

1. That with elastic database systems you only run precisely the amount of capacity that you need at any moment,
2. That elastic databases systems are inclined to bend but don't break, and
3. That elastic database systems emphasize automated administration rather than manual and physical administration.

These combine to provide gigantic benefits in terms of lifecycle costs. And for organizations running large numbers of database systems there are further benefits of resource reusability, consistent service management, and standard tooling across a full multi-tenant database service.

Capacity-related costs of inflexible databases

Database servers in a traditional data center commonly run at about 5%-10% utilization. Evidence from cloud vendors indicates that the number of server units decreases by around 80% for workloads that move to the cloud. The decreases apply as much to on-premises elastic deployments as they do with respect to IaaS servers. These figures will become better when database provisioning is as elastic as it is for web servers, application servers, and storage servers.

The traditional challenge for database server provisioning is around how you ensure sufficient capacity for the 500-year flood while simultaneously maximizing utilization of the servers. For many businesses, the cost of failure under peak or emergency load situations is sufficiently high that servers are commonly enormously over-provisioned relative to average loads. This is how the 5%-10% utilization numbers come about.

The **over-capacity** of the servers of course results in direct server costs that can be ten times what is generally used. But there are many other aspects of database lifecycle costs that are provisioned-capacity related. These costs include database licenses, commonly modeled on the capacity of the machines on which the database runs, irrespective of the actual utilization of the servers in question. Larger servers also carry premium costs for add-on components, including memory and storage upgrades, add-on software tools and components, and general data center costs (e.g. energy usage, cooling requirements, rack space, etc.).

In addition to the costs of over-capacity inflexible database systems also involve costs of **under-capacity**. If your database system gets overwhelmed by Black Friday e-commerce transactions or by a rapid market reset in financial services, you lose money. In some cases, an overwhelmed database simply results in poor UI latency, user frustration and migration of customers to competitor offerings. Different businesses and database workloads will have different characteristics in this regard, but in almost all cases the costs of a database system not keeping up with the load requirements is a substantial direct or indirect expense.

What you really want is a database system in which you have only enough resources deployed to support the load at any given moment in time, but which can add capacity quickly and efficiently when required. The antidote to the capacity-related costs of inflexible databases is database elasticity - databases that can add and delete processing and storage nodes as needed.

Downtime costs of inflexible databases

Database downtime can be planned or unplanned. Regardless of whether it's planned or unplanned; however, downtime of any kind is undesirable, and avoiding it provides significant economic benefits.

Planned downtime is required by inflexible databases for such mundane tasks as upgrading software, changing database schemas or even doing backups. This obviously results in unavailability of the system, in premium and episodic people-related administration costs, and in risks of delays and the loss of data or transactions. For many, planned downtime can be arranged for times of low usage and is manageable, but in an always-on world, most modern applications are expected to run 24x7.

Unplanned downtime is a major cost in enterprise database systems. When we talk to our customers and prospects about their pre-NuoDB experiences, they report downtime costs ranging from thousands to millions of dollars-per-hour. The latter is unusual of course - nevertheless for most businesses the numbers can be very large. The solutions typically include a mix of avoiding of downtime, minimizing restoration time, and providing best-effort service while the core system is down. Aside from the potentially very high costs of downtime there are also very high costs involved in High Availability strategies and technologies designed to reduce downtime in traditional databases.

Elastic database systems change the game on both planned and unplanned downtime.

Like most cloud-native systems, an elastic database is typically designed for zero planned downtime, with comprehensive online administration. You should be able to perform any administrative task without taking the system down. This includes changes to database structure, like schema changes or building indexes. It also includes upgrading servers, upgrading system software and upgrading the database software. Some elastic databases will allow you to move the database to new servers or even new data centers without taking the system down, without losing data or transactions, and without significant impact to application services or performance.

As relates to unplanned downtime, elastic databases have an unfair advantage. Redundancy is much easier when you have a system that allows arbitrary addition of both processing and storage nodes. And a system that allows nodes to be deleted could be expected to tolerate the failure of a node or group of nodes. Unlike an inflexible database system, a well-designed elastic database system has a natural resilience to failure. An elastic database that can run in multiple data centers simultaneously can also be resilient to data center failure - an Active/Active solution to DR challenges.

Elastic databases provide much stronger uptime guarantees than traditional inflexible databases, with potentially very large benefits in terms of direct and indirect costs of downtime.

System administration costs of inflexible databases

As general purpose, servers moved to the cloud, the ratio of servers to administrators expanded from 10:1 to 500:1 - more than an order-of magnitude increase. Databases need to follow suit. Large companies may be running thousands or tens of thousands of databases. And those databases have far too many individual needs, necessitating a great deal of DBA attention for each database.

What enables an order-of-magnitude improvement in the administrator to server ratio is breaking down the tight vertical integration of the entire database stack. If any database could be dynamically deployed or re-deployed on any server or set of servers at any time, without disrupting service, then admin tasks can be automated and involved via REST or other APIs from asingle pane of glass.

- Increasing performance is about adding machines to the running database.
- Reducing costs when database servers are under-utilized us about deleting machines from a running database.
- Moving a database to a larger or smaller machine is just about adding a bigger machine to the running database, and then shutting down the original machine.
- Changing a schema is about issuing some SQL commands in a running database
- Adding or deleting an index is about issuing SQL commands in a running database
- Upgrading the system is about adding new nodes and deleting old ones.

These are all software actions that can be provided as network APIs (e.g. REST) and can be scripted and automated using popular tools (e.g. Kubernetes, Mesos) or a scripting language of your choice.

An elastic database enables software defined administration models. With programmable administration an elastic database is, in effect, a software defined database system. This elevates the DBA task, automating the most mundane, time-consuming and error-prone tasks, while enabling the DBA to focus on rapid response to business requirements. As above there is an order-of-magnitude cost-reduction opportunity in the ratio of databases to DBAs. And there is a related benefit in enabling the DBA to respond to business needs much more rapidly.

Elastic databases provide an opportunity for much higher quality database administration at a fraction of the cost.

Economic inevitability of elasticity

The cost of running enterprise databases is typically the single biggest cost of running data centers. It is very expensive to meet database requirements for:

- Baseline Performance
- Peak Load Performance
- Uptime

- Business Continuity & Disaster Recovery
- Response to new Business Needs
- Multi-datacenter Operation

Elastic database systems address these needs with cloud-native and container-native designs that allow them to scale-out and scale-in dynamically.

In forthcoming posts, I will describe some of the design approaches that have been adopted to deliver on the promise. There are multiple solutions, all making different trade-offs in pursuit of the “Holy Grail” of the elastic database system.

This article is published as part of the IDG Contributor Network. Want to Join?

Next read this:

- *The secrets of highly successful data analytics teams*
- *The best ERP systems: 10 enterprise resource planning tools compared*
- *Hidden cloud migration gotchas — and how to avoid them*
- *IT Resume Makeover: Highlighting transformational leadership*
- *10 old-school IT principles that still rule*

Related: IT Budgeting CIO Role IT Leadership