

MySQL and Memcached: End of an Era?

Friday, February 26, 2010 at 9:06AM

Todd Hoff in Memcached, MySQL, nosql

If you look at the early days of this blog, when web scalability was still in its heady bloom of youth, many of the articles had to do with leveraging [MySQL](#) and [memcached](#). Exciting times. Shard MySQL to handle high write loads, cache objects in memcached to handle high read loads, and then write a lot of glue code to make it all work together. That was state of the art, that was how it was done. The architecture of many major sites still follow this pattern today, largely because with enough elbow grease, it works.



This was a pre-cloud, relational database dominated world, built from parts scrounged from the remnants of enterprises and datacenters past. [Twitter](#) and [Digg](#) started in this era, but are evolving into something different, as scaling pressures increase and new purpose built technologies pop into being.

With a little perspective, it's clear the MySQL+memcached era is passing. It will stick around for a while. Old technologies seldom fade away completely. Some still ride horses. Some still use CDs. And the Internet will not completely replace that archaic electro-magnetic broadcast technology called TV, but the majority will move on into a new era.

LinkedIn has moved on with their [Project Voldemort](#). Amazon went there [a while ago](#).

Digg declared their entrance into a new era in a post on their blog titled [Looking to the future with Cassandra](#), saying:

The fundamental problem is endemic to the relational database mindset, which places the burden of computation on reads rather than writes. This is completely wrong for large-scale web applications, where response time is critical. It's made much worse by the serial nature of most applications. Each component of the page blocks on reads from the data store, as well as the completion of the operations that come before it. Non-relational data stores reverse this model completely, because they don't have the complex read operations of SQL.

Twitter has also declared their move in the article [Cassandra @ Twitter: An Interview with Ryan King](#). Their reason for changing is:

We have a lot of data, the growth factor in that data is huge and the rate of growth is accelerating. We have a system in place based on shared mysql + memcache but its quickly becoming prohibitively costly (in terms of manpower) to operate. We need a system that can grow in a more automated fashion and be highly available.

It's clear that many of the ideas behind MySQL+memcached were on the mark, we see them preserved in the new systems, it's just that the implementation was a bit clunky. Developers have moved in, filled the gaps, sanded the corners, and made a new sturdy platform which will itself form the basis for a new ecosystem and a new era.

It's always a bit sad to see an era pass, but it's not all that often we get to

notice as it's happening. We can enjoy what has gone before, but we can also get pumped to jump in with both feet and create the future. And excitingly, that's what many leading edge companies are doing today.

Related Articles

[Plays well with others](#) by Mark Callaghan. *I think that MySQL + memcached is still the default choice and I don't think it is going away in the high-scale market.* The death of Memcached is greatly exaggerated

[The death of Memcached is greatly exaggerated](#) by Mark Atwood. *Memcached is going to stick around.*

[Cassandra and the NoSQL scalable OLTP argument](#) by Curt Monash. Cool, Curt Monash commented on my article. He poked a little fun at me of course, but that's OK, it's Curt Monash. Love his blog.

[Getting Real about NoSQL and the SQL-Isn't-Scalable Lie](#) by Dennis Forbes.

[MySQL+Memcached is still the workhorse](#) by Mark Atwood. *Running a real live system is running a farm, ploughing a field. You want a workhorse that you know how to use, you know you can get gear for at the blacksmith and the tackle shop, and that you know you can hire field hands who know how to use it well and take of properly.*

[Building Scalable Databases: Are Relational Databases Compatible with Large Scale Websites?](#) by Dare Obasanjo. *I expect we'll see more large scale websites decide that instead of treating a SQL database as a denormalized key-value pair store that they would rather use a NoSQL database.*

Article originally appeared on High Scalability (<http://highscalability.com/>).

<http://highscalability.com/blog/2010/2/26/mysql-and-memcached-end-of-an-era.html>

See website for complete article licensing information.