A periodic newsletter with links to blogs and industry news for subscribers to the Architecting IT blog.

View this email in your browser

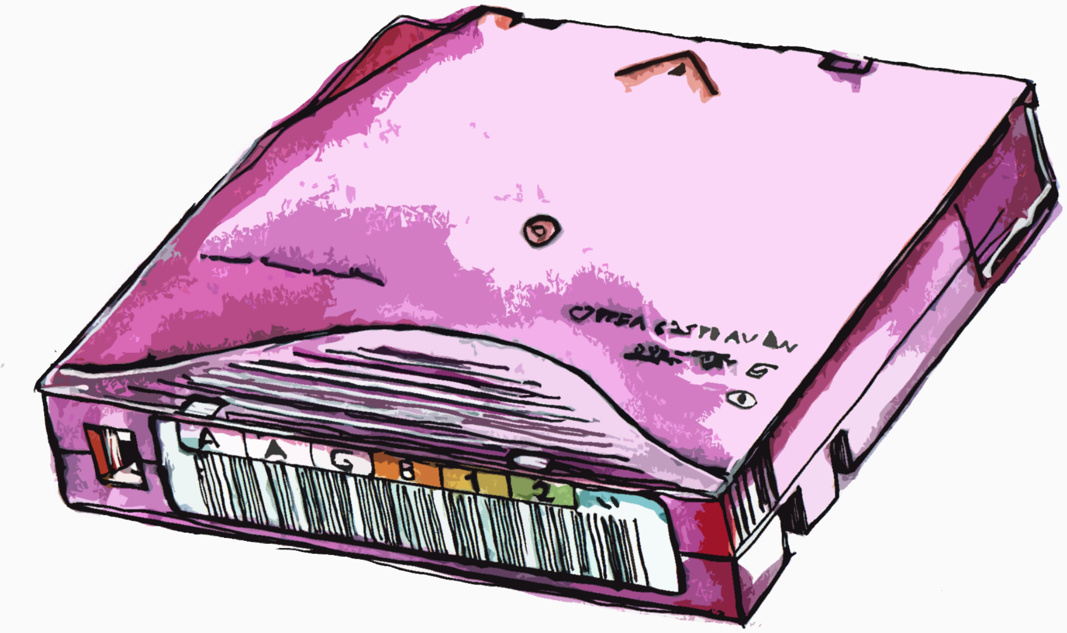
TECH.UNPLUGGED

TECHunplugged will return to London on Thursday 12 May 2016. The venue is The Kensington Close Hotel, Wright's Lane, London W8 5SP.

The event is free for end users (fee for vendors). Details can be found here (EventBrite registration page) with the ability to register and check out the agenda.

This is newsletter #23, up to 30 April 2016.

architectingIT



The Rise of Object Storage

For the last 30 years, block and file have been the dominant protocols in the data centre. Block is dumb, but fast - you can reference sections of a LUN or volume very quickly, with flash systems providing sub-millisecond and even a few hundred microseconds of latency. File-based systems add a level of intelligence to raw storage, operating (naturally) at the file level, providing structure to data and locking/serialisation capabilities. However neither block nor file are particularly practical at scale. File systems become more difficult to manage with millions of entries and can be problematic to scan and fix at boot time. Block requires either very big or lots of LUNs/volumes, each of which somehow still has to be mapped to content and kept consistent.

Object storage fills the need for the ability to store large quantities of binary data - objects that have no inherent relationship to the system in which they are being stored. Object stores can be used to store files, emails, audio, video and other proprietary formats that an accessing application can understand. The function of the object store is to simply store and retrieve content in a reliable fashion. That is, until now.

Multi-Protocol and Performance

Object storage is maturing and moving to a new level. The ability to store and retrieve data in an object store system (like Amazon's Simple Storage Service) requires the use of web-based protocols, such as the S3 API, or another proprietary format. Having to add code to an application in order to store and retrieve objects is a bit of a pain, so many on-premises object storage systems (like Scality Ring or Cloudian HyperStore) build in native support for other protocols such as file (SMB and NFS). This allows data to be stored in traditional formats, while retaining the benefit of having data in an object format.

The second feature starting to emerge is that of performance. Object stores were traditionally seen as having lower performance than file and block-based systems. This was naturally to be expected; block-based access can be measured in milliseconds (or less) because the volume of data transferred is low. Object stores can retrieve and store objects of megabytes or gigabytes in size, so latency has little meaning (other than to measure time to “first data”). Instead performance needs to be measured in terms of throughput, i.e. the number of MB/s transferred in/out of the system. High performance object stores fit the need for media content delivery and editing, as well as other object-based applications that need to store and retrieve data quickly.

The idea of referencing an object rather than writing to dumb storage does have a certain appeal. Platforms like Tintri's VMstore and others that support VVOLs can identify and encapsulate a virtual machine, allowing policies covering backup, quality of service and replication to be applied at the VM (object) level. Imagine extending this paradigm to databases. Today we already see support for Hadoop in many storage platforms. Why not allow storage to natively support a range of database protocols and implementations, pushing the functionality away from the operating system? This is what companies like Teradata are already doing.

The ultimate development of this paradigm sees storage evolve to a system that understands and manages content, rather than being unaware of the application being stored. Again we see the job of the storage administrator evolving into one of data guardian, managing the asset (data) rather than the infrastructure.

What do you think?

Chris Evans

New Architecting IT Blog Posts - April 2016

- IBM Reuses Existing Tech in Latest FlashSystem Release (29 April 2016)
- The Four Stages of All-Flash Storage (28 April 2016)
- When Did Hard Drives Get Workload Limits? (26 April 2016)
- A Quick Comment on IBM FlashSystem (7 April 2016)
- Revisiting Seagate Kinetic Drives (6 April 2016)

Tweet Share Forward Read Later