Memory-mapped Database for OpenLDAP

by Howard Chu

While OpenLDAP already provides a reliable high performance transactional backend database (using BerkeleyDB), it requires careful tuning to get good results and the tuning aspects can be quite complex. Data comes through three separate layers of caches before it may be used, and each cache layer has a significant footprint. Balancing the three layers against each other can be a difficult juggling act.

This talk presents the design and implementation of a new "back-mdb" memory-mapped database backend for OpenLDAP. This is built on top of a new mdb library written from scratch for the purpose. The library implements B-trees with multi-version concurrency support, and all "reads" are performed by mapping the entire database into virtual memory.

There are many benefits to the mdb design:

1. Multi-version concurrency allows reads to be performed with essentially no locking. This allows reads to scale across as many CPU cores as desired, with no synchronization bottlenecks.
2. data fetches are extremely fast - there are no mallocs or memcpy's anywhere in the data read path.
3. The database is operated with a copy-on-write style; active data pages are never overwritten. This approach eliminates any potential for corruption on-disk and eliminates the need for write-ahead transaction logs and their associated housekeeping.
4. The memory map is read-only, so stray writes from buggy code also cannot corrupt any in-memory database structures.
5. Trivial configuration - DB configuration is far simpler than for BerkeleyDB. It requires only a pathname and a maximum size for the memory map.
6. Cloud-friendly, virtual machine-friendly. It will only use as much RAM as the host provides; it will never drive into swap space.
7. Simpler code - back-bdb/cache.c is the single largest source file in that backend. Managing the cache has always required the greatest investment of labor.
8. Memory efficiency - instead of data residing in the filesystem cache, the BerkeleyDB cache, and the slapd entry cache, it only resides in the filesystem cache. This increases the potential volume of data that can be served by 2-3x compared to BerkeleyDB in the same amount of RAM. back-mdb and the underlying mdb library are a work-in-progress. We anticipate it will be operational by September but its characteristics may change between now and then.