# The 10 Deadly Sins Against Scalability

Monday, June 10, 2013 at 8:27AM

Todd Hoff in Strategy


In the moral realm there may be 7 deadly sins, but scalability maven Sean Hull has come up Five More Things Deadly to Scalability that when added to his earlier 5 Things That are Toxic to Scalability, make for a numerologically satisfying 10 sins again scalability:

1. **Slow Disk I/O – RAID 5 – Multi-tenant EBS**. Use RAID 10, it provides good protection along with good read and write performance. The design of RAID 5 means poor performance and long repair times on failure. On AWS consider Provisioned IOPS as a way around IO bottlenecks.
2. **Using the database for Queuing.** The database may seem like the perfect place to keep work queues, but under load locking and scanning overhead kills performance. Use specialized products like RabbitMQ and SQS to remove this bottleneck.
3. **Using Database for full-text searching.** Search seems like another perfect database feature. At scale search doesn't perform well. Use specialized technologies like Solr or Sphinx.
4. **Insufficient Caching at all layers.** Use memcache between your application and the database. Use a page like cache like Varnish between users and your webserver. Select proper caching options for your html assets.
5. **Too much technical debt.** Rewrite problem code instead of continually paying a implementation tax for poorly written code. In the long run it pays off.
6. **Object Relational Mappers.** Create complex queries that hard to optimize and tweak.
7. **Synchronous, Serial, Coupled or Locking Processes.** Locks are like stop signs, traffic circles keep the traffic flowing. Row level locking is better than table level locking. Use async replication. Use eventual consistency for clusters.
8. **One Copy of Your Database.** A single database server is a choke point. Create parallel databases and let a driver select between them.
9. **Having No Metrics.** Visualize what's happening to your system using one of the many monitoring packages.
10. **Lack of Feature Flags.** Be able to turn off features via a flag so when a spike hits features can be turned off to reduce load.

---

Article originally appeared on (http://highscalability.com/).

See website for complete article licensing information.