

NoSQL Do's and Don'ts

Andrew J. Brust

<http://www.bluebadgeinsights.com>

andrew.brust@bluebadgeinsights.com



Performance



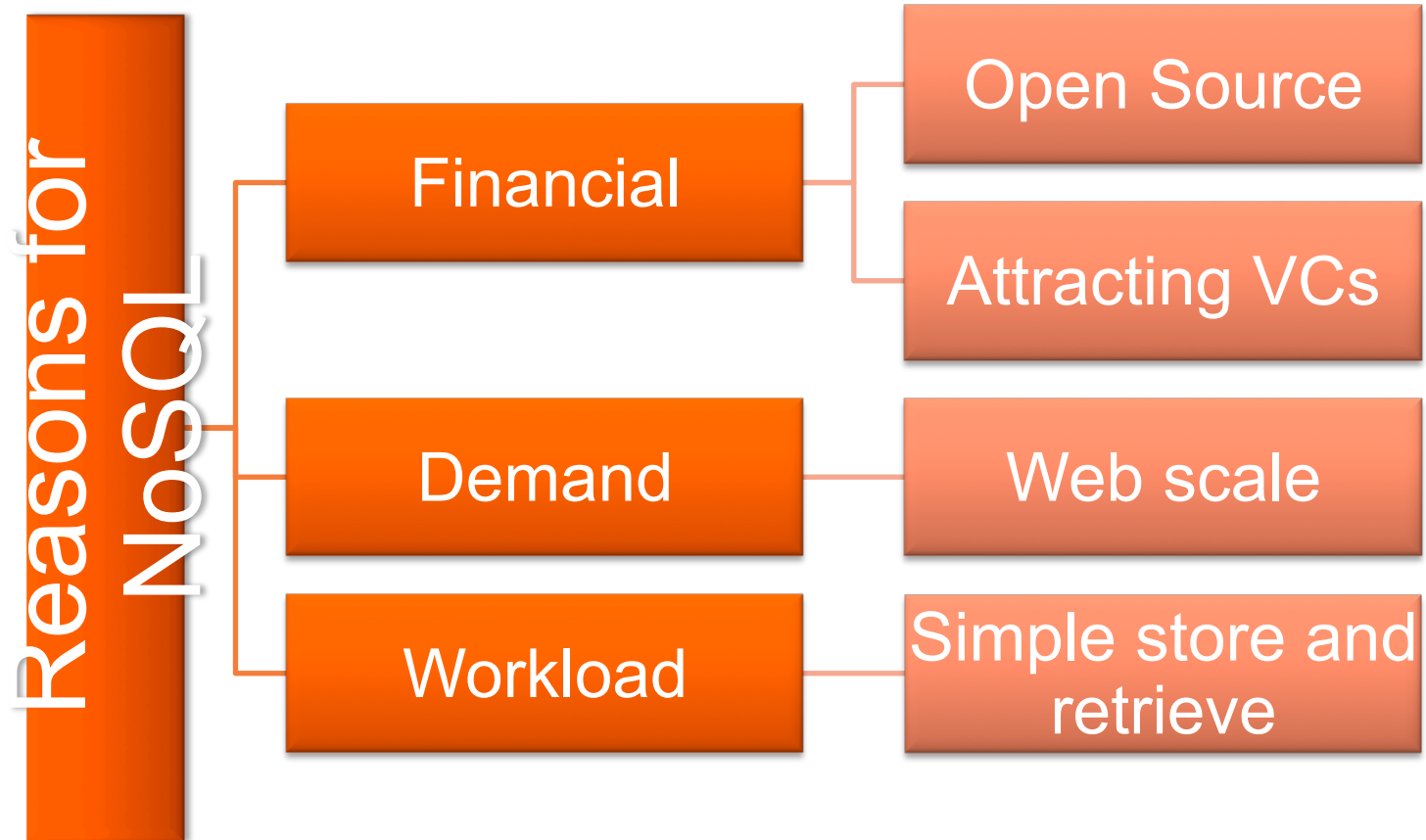
- **For certain types of queries, performance on NoSQL databases can be unacceptably slow**
- **For Line of Business application loads, relational databases will be just as fast, and far more robust/reliable**
- **But if you have just a few bits of information to fetch and enormous traffic and concurrency, NoSQL may be best**

Appropriateness

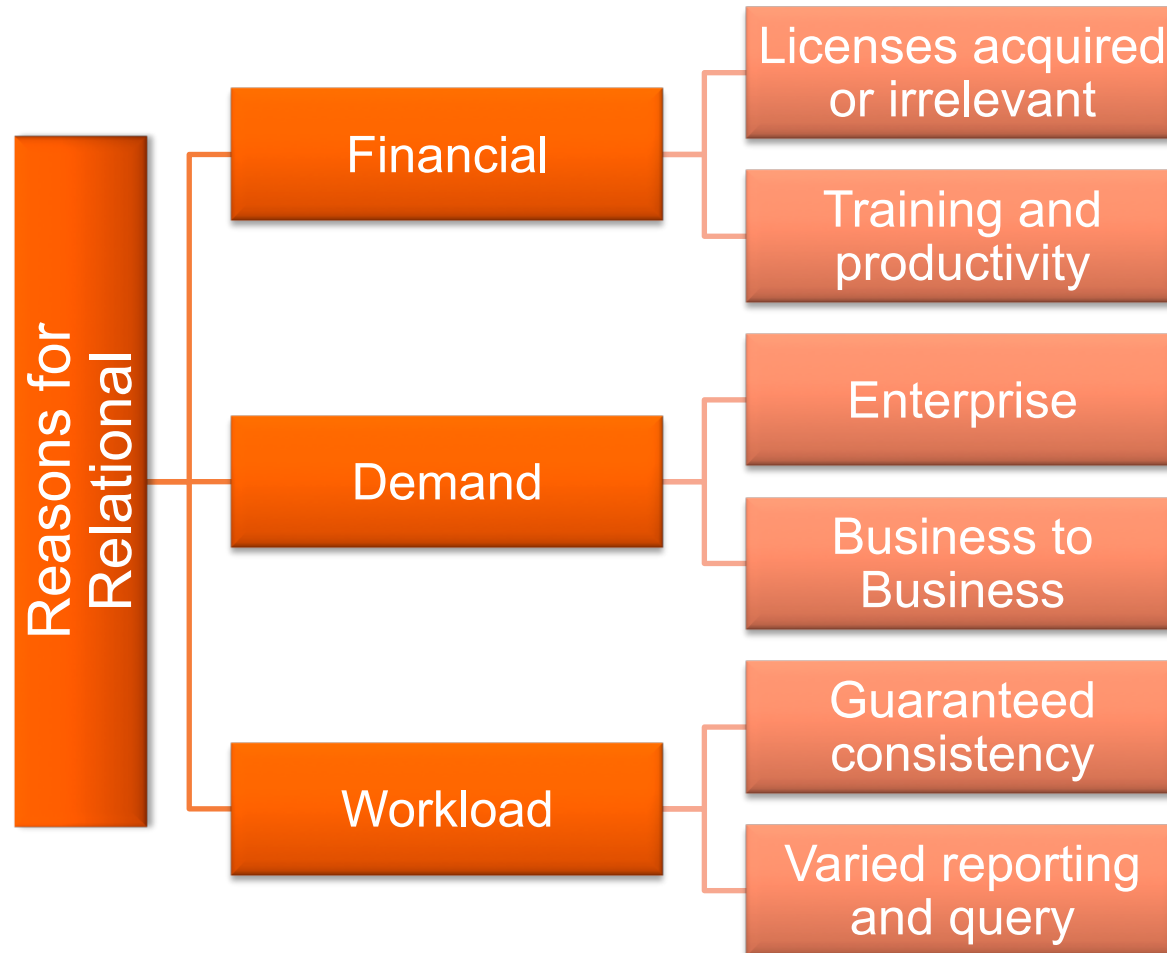


- **Using relational databases for the following may be imprudent:**
 - Configuration data
 - Environment data, settings, etc.
 - Log file, event data
- **Using NoSQL for these would be extreme:**
 - Stock trades
 - Accounting debits and credits
 - Credit card transactions
- **Don't be a partisan; use the right tool for the job.**

Supporting NoSQL



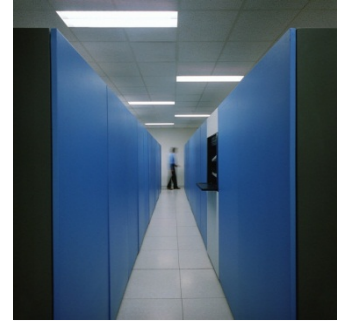
Supporting Relational



NoSQL + BI

- **NoSQL databases are not designed for query and data warehousing**
- **BI applications involve models; models rely on schema**
- **Extract, transform and load (ETL) may be your friend**
- **Wide-column stores, however are good for “Big Data”**
 - See next slide
- **Wide-column stores and column-oriented databases are similar technologically**

NoSQL + Big Data



- **Big Data and NoSQL are interrelated**
- **Typically, Wide-Column stores used in Big Data scenarios**
- **Prime example:**
 - HBase and Hadoop
- **Why?**
 - Lack of indexing not a problem
 - Consistency not an issue
 - Fast reads very important
 - Distributed files systems important too
 - Commodity hardware and disk assumptions also important
 - Not Web scale but massive scale-out, so similar concerns