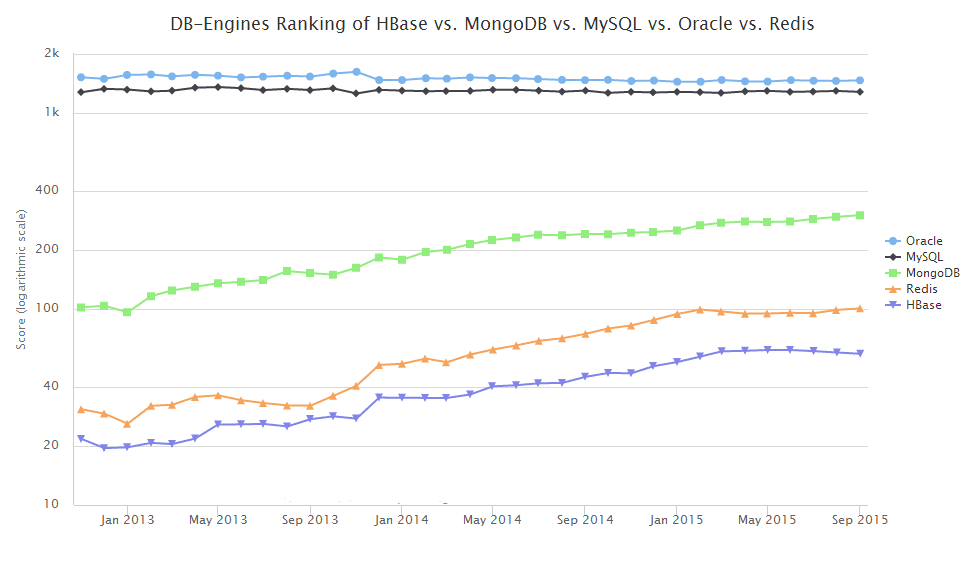
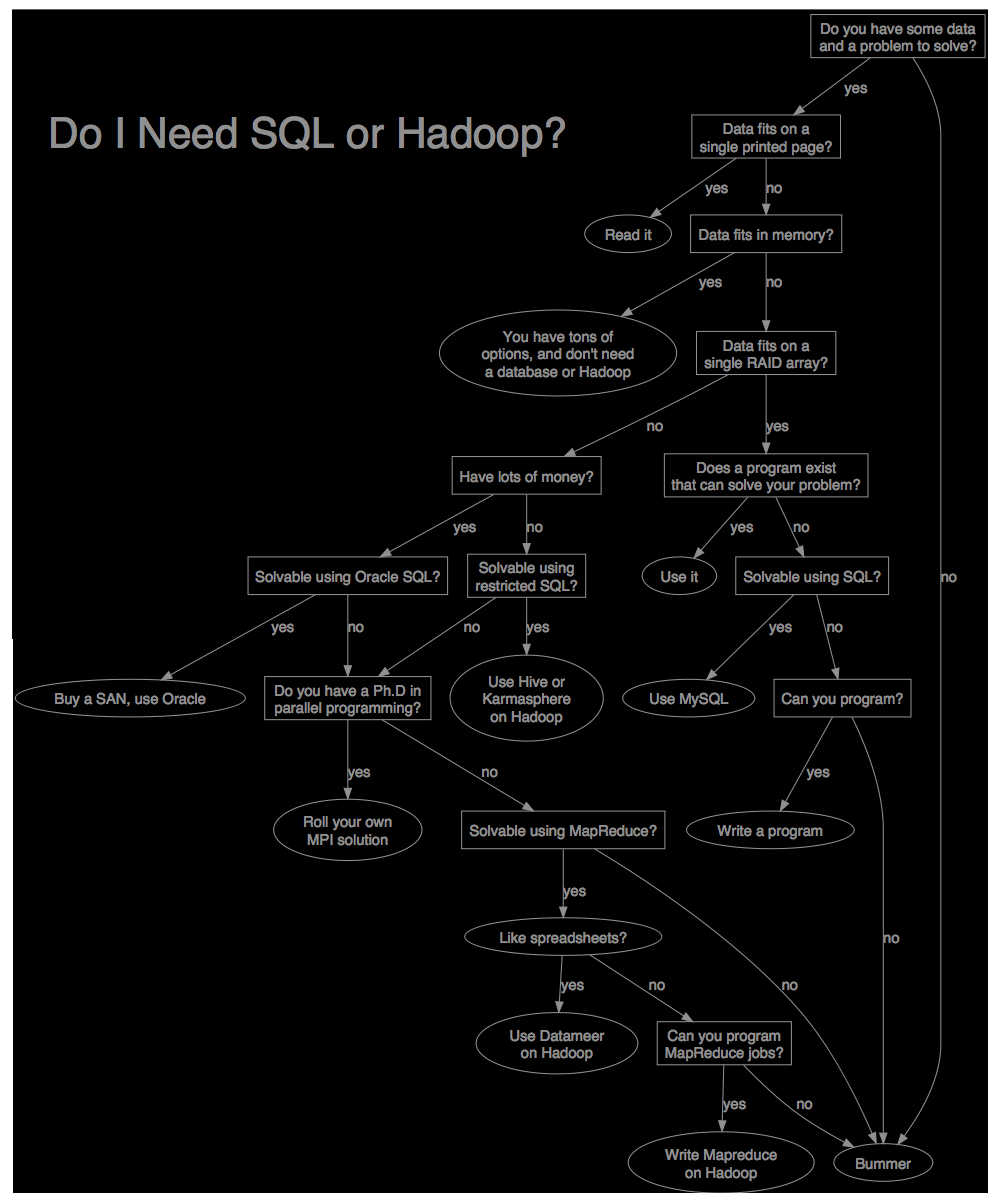
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HBase vs. MongoDB vs. MySQL vs. Oracle vs. Redis，三大主流开源 NoSQL 数据库的 PK 两大主流传统 SQL 数据库 | | | | | |
| **类别** | **HBase** | **MongoDB** | **MySQL** | **Oracle** | **Redis** |
| **描述** | 基于 Apache Hadoop 并提供 BigTable 能力的列存储 | 最受欢迎的文档存储数据库之一 | 广泛使用的开源 RDBMS | 广泛使用的 RDBMS | 内存中的数据结构存储，被用作数据库，缓存以及消息中间件 |
| **普及度等级(参考下文的趋势图)** | 15 级(59.03 分) | 4 级(300.57 分) | 2 级(1277.75 分) | 1 级(1463.37 分) | 10 级(100.65 分) |
| **数据库模型** | 列存储 | 文档存储 | 关系数据库系统 | 关系数据库系统 | 键-值存储 |
| **官网** | [hbase.apache.org](http://hbase.apache.org/) | [www.mongodb.org](http://www.mongodb.org/) | [www.mysql.com](http://www.mysql.com/) | [www.oracle.com/-us/-products/-database](http://www.oracle.com/us/products/database/) | [redis.io](http://redis.io/) |
| **技术文档** | [hbase.apache.org](http://hbase.apache.org/) | [docs.mongodb.org/-manual](http://docs.mongodb.org/manual/) | [dev.mysql.com/-doc](http://dev.mysql.com/doc/) | [www.oracle.com/-technetwork/-indexes/-documentation/-index.html](http://www.oracle.com/technetwork/indexes/documentation/index.html) | [redis.io/-documentation](http://redis.io/documentation) |
| **开发者** | Apache 软件基金会 | MongoDB 公司 | Oracle | Oracle | Salvatore Sanfilippo(Redis 之父) |
| **首次发行** | 2008 | 2009 | 1995 | 1980 | 2009 |
| **当前版本** | 1.1.0.1，2015 年 5 月 | 3.0.5，2015 年 7 月 | 5.6.26，2015 年 7 月 | 12 Release 1 (12.1.0.2)，2014 年 7 月 | 3.0.3，2015 年 6 月 |
| **许可** | 开源 | 开源 | 开源 | 商业 | 开源 |
| **是否数据库即服务(DBaaS)** | 否 | 否 | 否 | 否 | 否 |
| **实现语言** | Java | C++ | C 和 C++ | C 和 C++ | C |
| **服务器操作系统** | Linux Unix Windows | Linux OS X Solaris Windows | FreeBSD Linux OS X Solaris Windows | AIX HP-UX Linux OS X Solaris Windows z/OS | BSD Linux OS X Windows |
| **是否结构化数据** | 自由 | 自由 | 是 | 是 | 自由 |
| **是否预定义数据类型** | 否 | 是 | 是 | 是 | 部分 |
| **是否支持 XML** | 否 |  | 是 | 是 | 否 |
| **是否支持二级索引** | 否 | 是 | 是 | 是 | 否 |
| **是否支持 SQL** | 否 | 否 | 是 | 是 | 否 |
| **API 以及其他访问方式** | Java API RESTful HTTP API Thrift | 使用 JSON 的专用协议 | ADO.NET JDBC ODBC | ODP.NET Oracle Call Interface (OCI) JDBC ODBC | 专用协议 |
| **支持的编程语言** | C C# C++ Groovy Java PHP Python Scala | Actionscript  C C# C++ Clojure  ColdFusion  D  Dart  Delphi  Erlang Go  Groovy  Haskell Java JavaScript Lisp  Lua  MatLab  Perl PHP PowerShell  Prolog  Python R  Ruby Scala Smalltalk | Ada C C# C++ D Eiffel Erlang Haskell Java Objective-C OCaml Perl PHP Python Ruby Scheme Tcl | C C# C++ Clojure Cobol Eiffel Erlang Fortran Groovy Haskell Java JavaScript Lisp Objective C OCaml Perl PHP Python R Ruby Scala Tcl Visual Basic | C C# C++ Clojure Crystal D Dart Elixir Erlang Fancy Go Haskell Haxe Java JavaScript (Node.js) Lisp Lua MatLab Objective-C OCaml  Perl PHP Prolog Pure Data Python R Rebol Ruby Rust Scala Scheme Smalltalk Tcl |
| **是否支持服务端脚本** | 是 | JavaScript | 是 | PL/SQL | Lua |
| **是否支持触发器** | 是 | 否 | 是 | 是 | 否 |
| **切分方式** | 分片 | 分片 | 水平切分，使用 MySQL Cluster 或者 MySQL Cluster 进行分片 | 水平切分 | 分片 |
| **主从复制方式** | 多种主从复制机制 | 主-从复制 | 主-主复制 主-从复制 | 主-主复制 主-从复制 | 主-从复制 |
| **是否支持 MapReduce** | 是 | 是 | 否 | 否 | 否 |
| **分布式场景下数据一致性方法** | 立即一致 | 最终一致 立即一致 |  | 立即一致 | 最终一致 |
| **是否支持外键** | 否 | 否 | 是 | 是 | 否 |
| **是否支持事务** | 否 | 否 | ACID | ACID | 乐观锁机制，原子性执行的命令块和脚本 |
| **是否支持并发** | 是 | 是 | 是 | 是 | 是 |
| **是否支持持久化存储** | 是 | 是 | 是 | 是 | 是 |
| **是否支持内存存储** | 否 |  | 是 | 是 | 是 |
| **访问控制** | 访问控制列表(ACL) | 基于用户和角色的访问权限 | 细粒度的用户访问权限 | 根据 SQL 标准细粒度的访问权限 | 简单的基于密码的访问控制 |

**附录一：普及度等级趋势图**

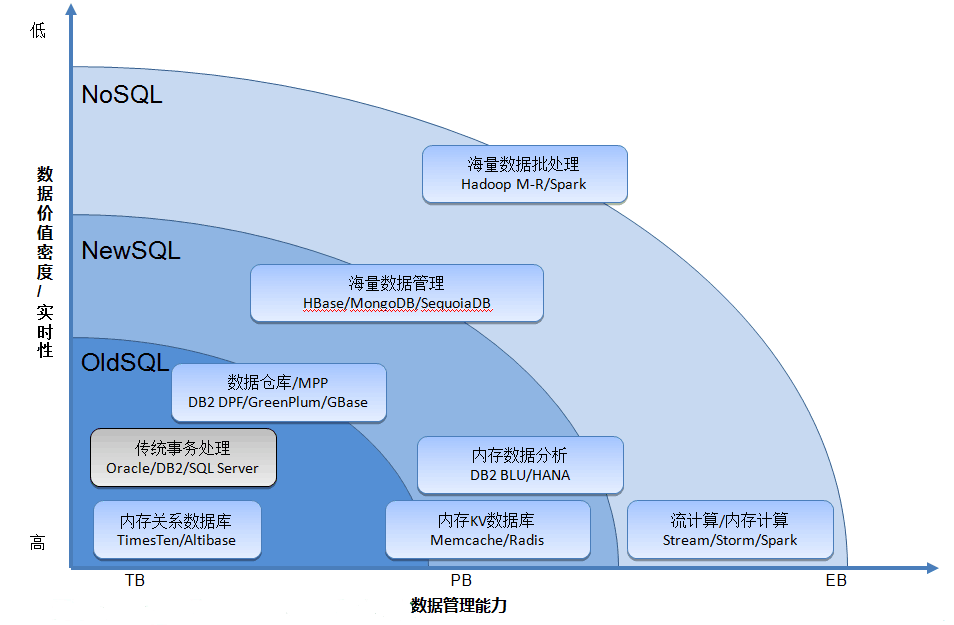


**附录二：两张图告诉你如何在 SQL、NewSQL、NoSQL 之间进行取舍**

图一：Do I Need SQL or Hadoop?



图二：SQL vs. NewSQL vs. NoSQL



**附录三：Redis 之父 Salvatore Sanfilippo 访谈节选**

Redis 是一个开源的、高级键值数据库和数据结构服务程序，其中键可以保存字符串、散列、列表、集合和有序集合。

Redis 的内核是用标准 ANSI C 写成的，基于一种事件模型。

非阻塞复制是 Redis 的设计目标。

Redis 中的复制是异步的。

目前已经为许多语言（包括大部分常用语言）提供了 Redis 库。

C client 是唯一官方支持的包装。

Redis Pub/Sub 对实时应用非常适合。

Redis 用户有的将它用作数据库，有的用作消息总线，也有的用来做 cache。

**附录四：NoSQL 数据库的类型一览表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NoSQL 数据库的类型一览表 | | | | | |
| **数据库类型** | **描述** | **主流产品** | **有谁在用** | **适用场景** | **不适用场景** |
| **键值（Key-Value）数据库** | 键值数据库就像在传统语言中使用的哈希表。你可以通过 key 来添加、查询或者删除数据，鉴于使用主键访问，所以会获得不错的性能及扩展性。 | Riak、Redis、Memcached、Amazon’s Dynamo、Project Voldemort | GitHub （Riak）、BestBuy （Riak）、Twitter （Redis和Memcached）、StackOverFlow （Redis）、 Instagram （Redis）、Youtube （Memcached）、Wikipedia（Memcached） | 储存用户信息，比如会话、配置文件、参数、购物车等等。这些信息一般都和 ID（键）挂钩，这种情景下键值数据库是个很好的选择。 | 1. 取代通过键查询，而是通过值来查询。Key-Value 数据库中根本没有通过值查询的途径。 2. 需要储存数据之间的关系。在 Key-Value 数据库中不能通过两个或以上的键来关联数据。 3. 事务的支持。在 Key-Value 数据库中故障产生时不可以进行回滚。 |
| **面向文档（Document-Oriented）数据库** | 面向文档数据库会将数据以文档的形式储存。每个文档都是自包含的数据单元，是一系列数据项的集合。每个数据项都有一个名称与对应的值，值既可以是简单的数据类型，如字符串、数字和日期等；也可以是复杂的类型，如有序列表和关联对象。数据存储的最小单位是文档，同一个表中存储的文档属性可以是不同的，数据可以使用 XML、JSON 或者 JSONB 等多种形式存储。 | MongoDB、CouchDB、RavenDB | SAP （MongoDB）、Codecademy （MongoDB）、Foursquare （MongoDB）、NBC News （RavenDB） | 1. 日志。企业环境下，每个应用程序都有不同的日志信息。Document-Oriented 数据库并没有固定的模式，所以我们可以使用它储存不同的信息。 2. 分析。鉴于它的弱模式结构，不改变模式下就可以储存不同的度量方法及添加新的度量。 | 在不同的文档上添加事务。Document-Oriented 数据库并不支持文档间的事务，如果对这方面有需求则不应该选用这个解决方案。 |
| **列存储（Wide Column Store/Column-Family）数据库** | 列存储数据库将数据储存在列族（column family）中，一个列族存储经常被一起查询的相关数据。举个例子，如果我们有一个 Person 类，我们通常会一起查询他们的姓名和年龄而不是薪资。这种情况下，姓名和年龄就会被放入一个列族中，而薪资则在另一个列族中。 | Cassandra、HBase | Ebay （Cassandra）、Instagram （Cassandra）、NASA （Cassandra）、Twitter （Cassandra and HBase）、Facebook （HBase）、Yahoo!（HBase） | 1. 日志。因为我们可以将数据储存在不同的列中，每个应用程序可以将信息写入自己的列族中。 2. 博客平台。我们储存每个信息到不同的列族中。举个例子，标签可以储存在一个，类别可以在一个，而文章则在另一个。 | 1. 如果我们需要 ACID 事务。Vassandra 就不支持事务。 2. 原型设计。如果我们分析Cassandra 的数据结构，我们就会发现结构是基于我们期望的数据查询方式而定。在模型设计之初，我们根本不可能去预测它的查询方式，而一旦查询方式改变，我们就必须重新设计列族。 |
| **图（Graph-Oriented）数据库** | 图数据库允许我们将数据以图的方式储存。实体会被作为顶点，而实体之间的关系则会被作为边。比如我们有三个实体，Steve Jobs、Apple 和 Next，则会有两个“Founded by”的边将 Apple 和 Next 连接到 Steve Jobs。 | Neo4J、Infinite Graph、OrientDB | Adobe （Neo4J）、Cisco （Neo4J）、T-Mobile （Neo4J） | 1. 在一些关系性强的数据中 2. 推荐引擎。如果我们将数据以图的形式表现，那么将会非常有益于推荐的制定 | 不适合的数据模型。图数据库的适用范围很小，因为很少有操作涉及到整个图。 |

**参考资料**

* [HBase vs. MongoDB vs. MySQL vs. Oracle vs. Redis Popularity](http://db-engines.com/en/system/HBase%3BMongoDB%3BMySQL%3BOracle%3BRedis)
* [一网打尽当下NoSQL类型、适用场景及使用公司](http://www.csdn.net/article/2013-07-24/2816330-how-to-choose-nosql-db)
* [Redis之父Salvatore Sanfilippo访谈](http://www.csdn.net/article/2011-01-18/290347)
* [一张图告诉你是需要 SQL 还是 Hadoop](http://blog.csdn.net/defonds/article/details/47005827)