

RedisJSON 横空出世！

程序员自修室 2021-12-22 11:30



程序员自修室

程序员的第一站！

47篇原创内容



公众号

近期官网给出了RedisJson (RedisSearch) 的性能测试报告，可谓碾压其他NoSQL，下面是核心的报告内容，先上结论：

- 对于隔离写入(isolated writes)，RedisJSON 比 MongoDB 快 5.4 倍，比 ElasticSearch 快 200 倍以上。
- 对于隔离读取(isolated reads)，RedisJSON 比 MongoDB 快 12.7 倍，比 ElasticSearch 快 500 倍以上。

在混合工作负载场景中，实时更新不会影响 RedisJSON 的搜索和读取性能，而 ElasticSearch 会受到影响。以下是具体的数据：

- RedisJSON* 支持的操作数/秒比 MongoDB 高约 50 倍，比 ElasticSearch 高 7 倍/秒。
- RedisJSON* 的延迟比 MongoDB 低约 90 倍，比 ElasticSearch 低 23.7 倍。

此外，RedisJSON 的读取、写入和负载搜索延迟在更高的百分位数中远比 ElasticSearch 和 MongoDB 稳定。当增加写入比率时，RedisJSON 还能处理越来越高的整体吞吐量，而当写入比率增加时，ElasticSearch 会降低它可以处理的整体吞吐量。

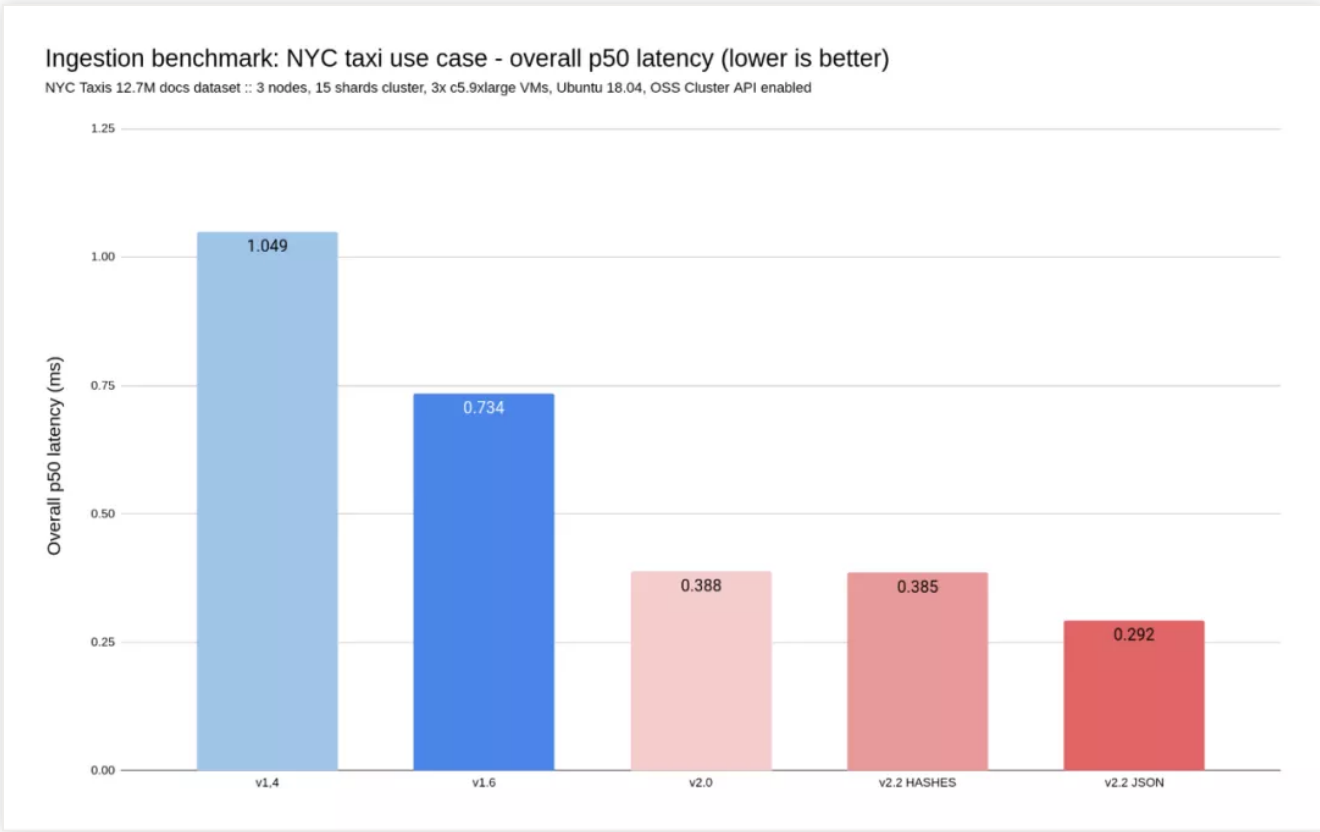
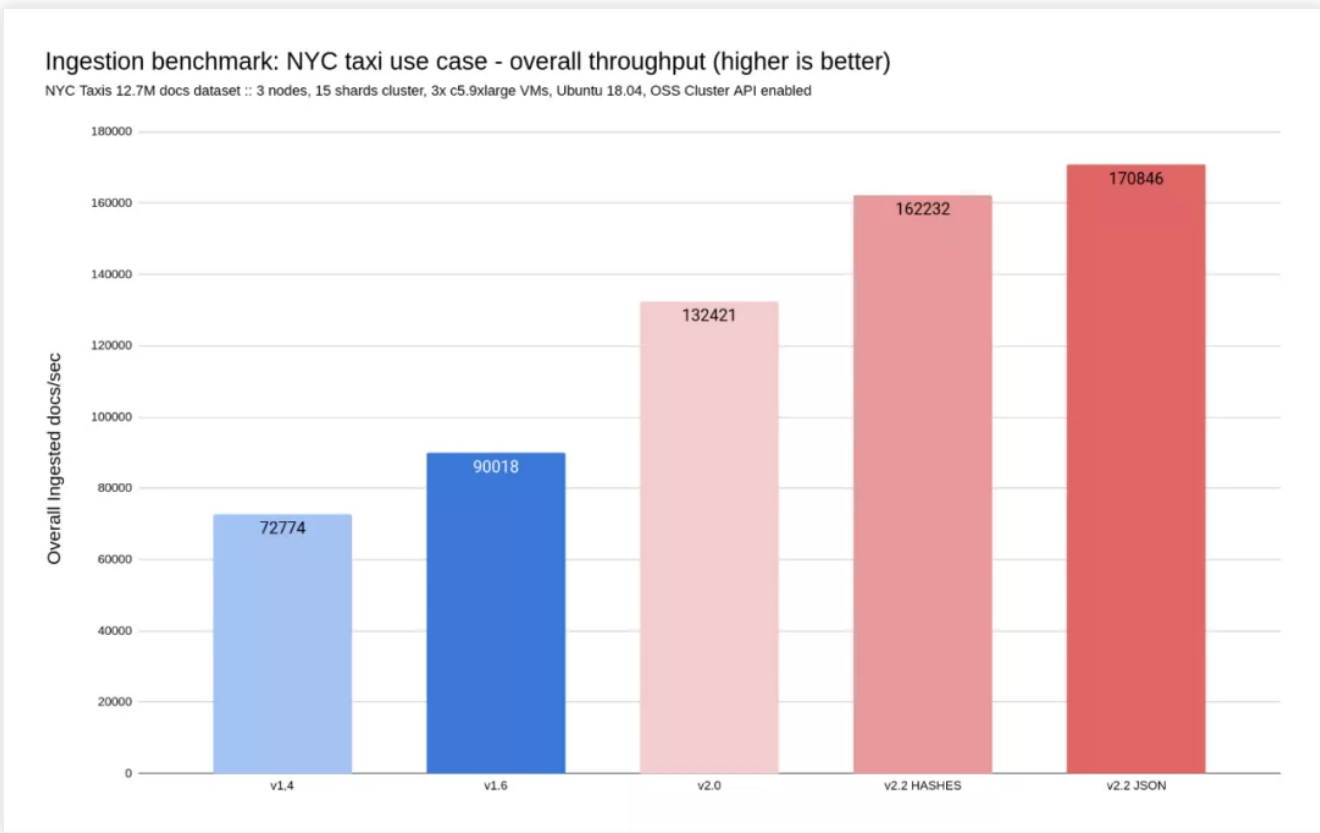
查询引擎

如前所述，reresearch和RedisJSON的开发非常强调性能。对于每一个版本，我们都想确保开发者可以体验到稳定和性能。为此，我们给出了一些分析工具、探测器来进行性能分析。

并且，我们每次发行新版本时时，也在不断的提升性能。特别是对于reresearch来说，2.2版本在加载和查询性能上都比2.0快了1.7倍，同时还改进了吞吐量和数据加载的延迟。

1.加载优化

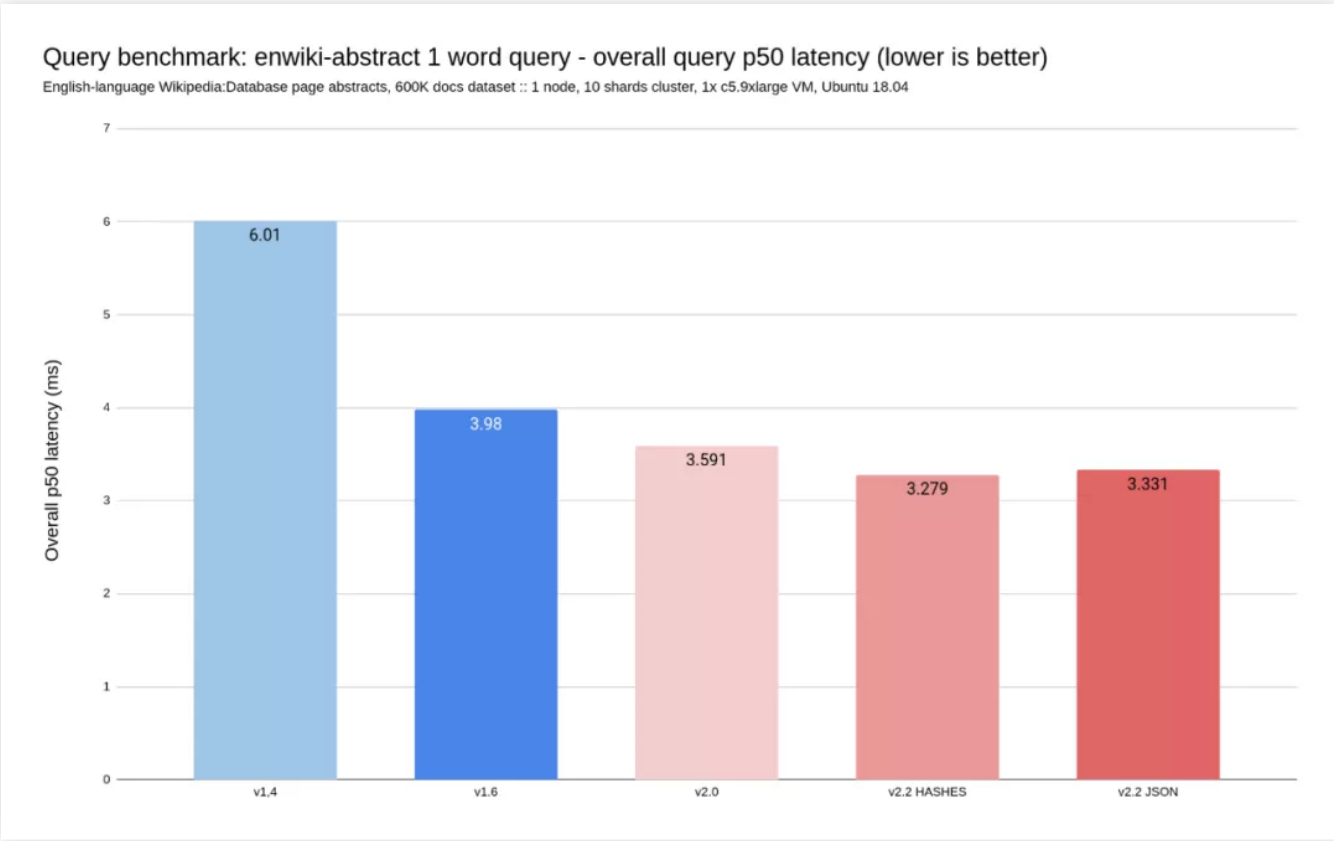
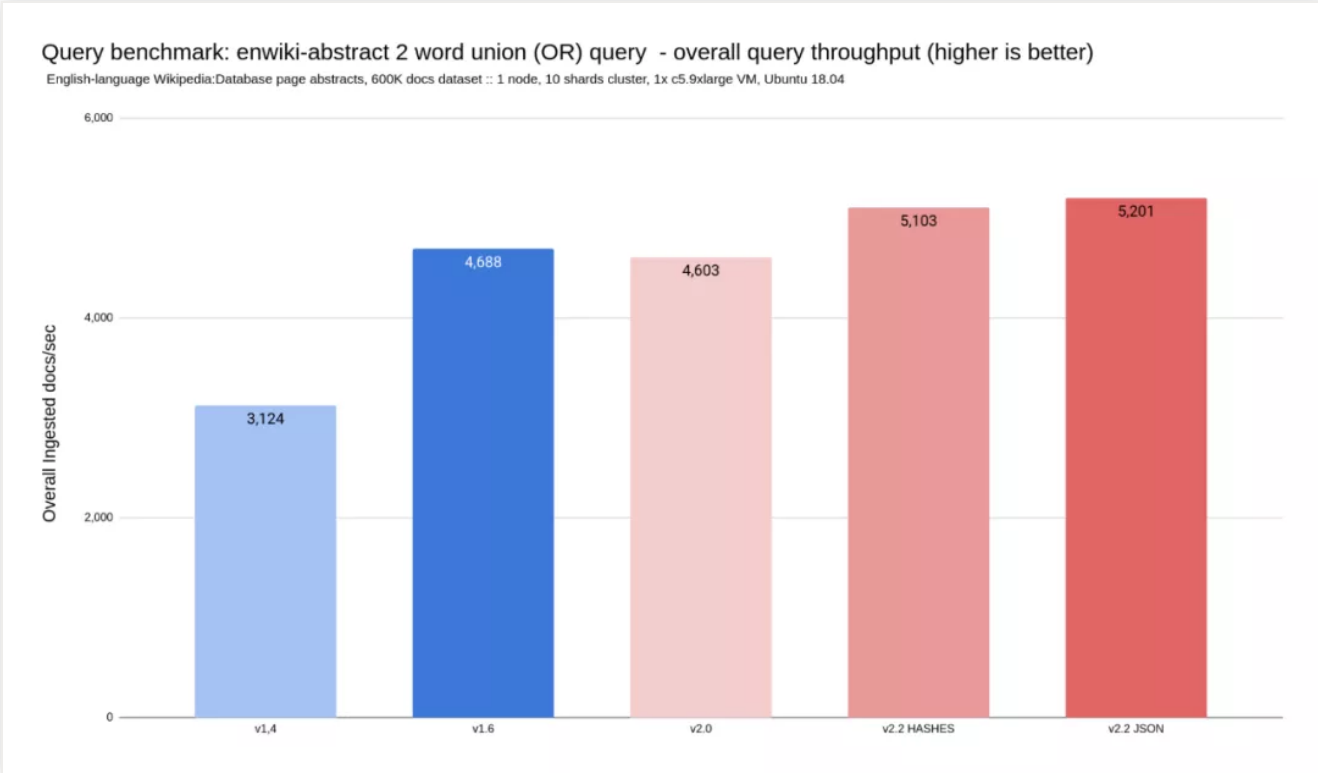
接下来的两个图显示了运行纽约市出租车基准测试的运行结果。该基准测试测量了吞吐量和加载耗时等基础数据。



从这些图表中可以看出，每一个rerearch的新版本都有一个实质性的性能改进。

2.全文搜索优化

为了评估搜索性能，我们索引了590万篇维基百科摘要。然后我们运行一个全文搜索查询面板，得到的结果如下图所示



从上面的图可以看出，通过从v2.0迁移到v2.2，同样的数据，在写、读、搜索(延迟图)方面都有了大幅度的改进，从而提高了运行Search和JSON的可实现吞吐量。

和其他框架的对比

为了评估RedisJSON的性能，我们决定将它与MongoDB和ElasticSearch进行比较。为了方便对比，我们会从文档存储、本地可用、云中可用、专业支持和提供可伸缩性、性能等方面进行全方位的对比。

我们使用了完善的YCSB标准来进行测试对比，它能够基于常见的工作负载来评估不同的产品，测量延迟、吞吐量曲线直到饱和。除了CRUD YCSB操作之外，我们还添加了一个两个字的搜索操作，专门帮助开发人员、系统架构师和DevOps从业者找到适合他们用例的最佳搜索引擎。

1. 基准测试

此次测试，我们使用了如下的一些软件环境：

- MongoDB v5.0.3
- ElasticSearch 7.15
- RedisJSON (RediSearch 2.2+RedisJSON 2.0)

此次是在Amazon Web Services 实例上运行基准测试，这三种解决方案都是分布式数据库，并且最常用于生产中的分布式方式。这就是为什么所有产品都使用相同的通用 m5d.8xlarge VM 和本地 SSD，并且每个设置由四个 VM 组成：一个客户端 + 三个数据库服务器。基准测试客户端和数据库服务器都在处于最佳网络条件下的单独 m5d.8xlarge 实例上运行，将实例紧密地打包在一个可用区内，实现稳态分析所需的低延迟和稳定的网络性能。

测试是在三节点集群上执行的，部署细节如下：

- MongoDB 5.0.3：三成员副本集（Primary-Secondary-Secondary）。副本用于增加读取容量并允许更低的延迟读取。为了支持对字符串内容的文本搜索查询，在搜索字段上创建了一个文本索引。
- ElasticSearch 7.15：15 个分片设置，启用查询缓存，并为 2 个基于 NVMe 的本地 SSD 提供 RAID 0 阵列，以实现更高级别的文件系统相关弹性操作性能。这 15 个分片为我们为 Elastic 所做的所有分片变体提供了可实现的最佳性能结果。
- RedisJSON*：RediSearch 2.2 and RedisJSON 2.0: OSS Redis Cluster v6.2.6，有27个分片，均匀分布在三个节点上，加载了RediSearch 2.2和RedisJSON 2.0 OSS模块。

除了这个主要的基准/性能分析场景之外，我们还在网络、内存、CPU 和 I/O 上运行基准基准测试，以了解底层网络和虚拟机特性。在整个基准测试集期间，网络性能保持在带宽和 PPS 的测量限制以下，以产生稳定稳定的超低延迟网络传输（每个数据包 $p99 < 100\text{micro}s$ ）。

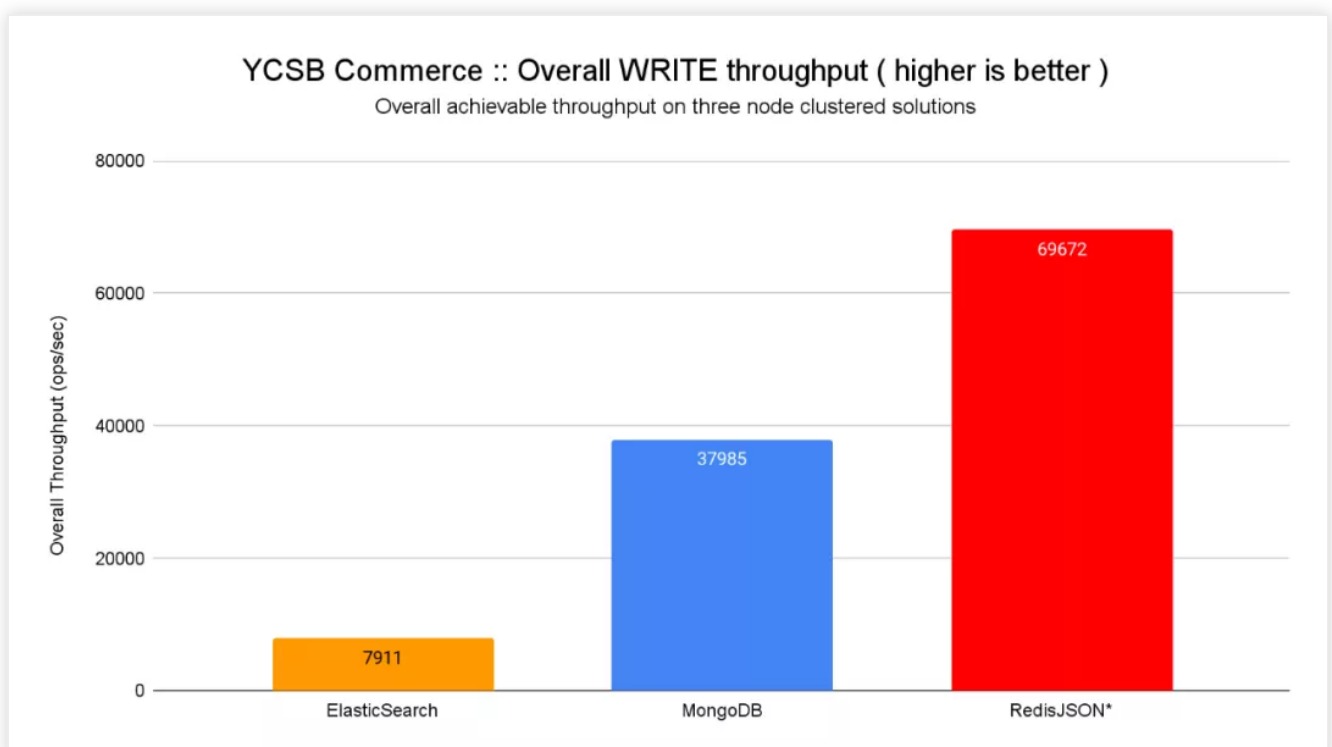
接下来，我们将从提供单独的操作性能 [100% 写入] 和 [100% 读取] 开始，并以一组混合工作负载结束以模拟现实工作中的应用程序场景。

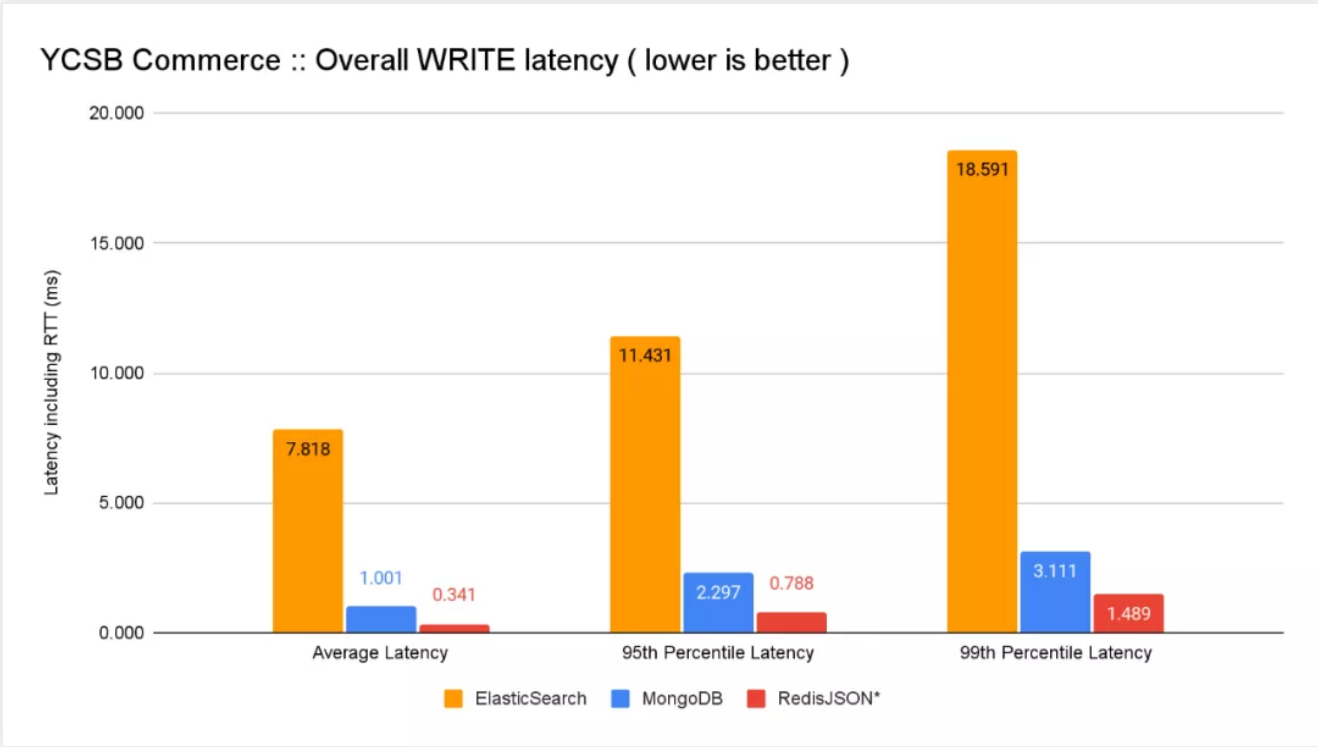
2. 100% 写入基准

如下图所示，该基准测试表明，RedisJSON* 的摄取速度比 Elasticsearch 快 8.8 倍，比 MongoDB 快 1.8 倍，同时保持每个操作的亚毫秒级延迟。值得注意的是，99% 的 Redis 请求在不到 1.5 毫秒的时间内完成。

此外，RedisJSON* 是我们测试过的唯一一种在每次写入时自动更新其索引的解决方案。这意味着任何后续的搜索查询都会找到更新的文档。ElasticSearch 没有这种细粒度的容量；它将摄取的文档放在一个内部队列中，并且该队列由服务器（不受客户端控制）每 N 个文档或每 M 秒刷新一次。他们称这种方法为近实时 (NRT)。Apache Lucene 库（它实现了 ElasticSearch 的全文功能）旨在快速搜索，但索引过程复杂且繁重。如这些 WRITE 基准测试图表所示，由于这种“设计”限制，ElasticSearch 付出了巨大的代价。

结合延迟和吞吐量改进，RedisJSON* 比 Mongoddb 快 5.4 倍，比 Elasticsearch 快 200 倍以上，用于隔离写入。

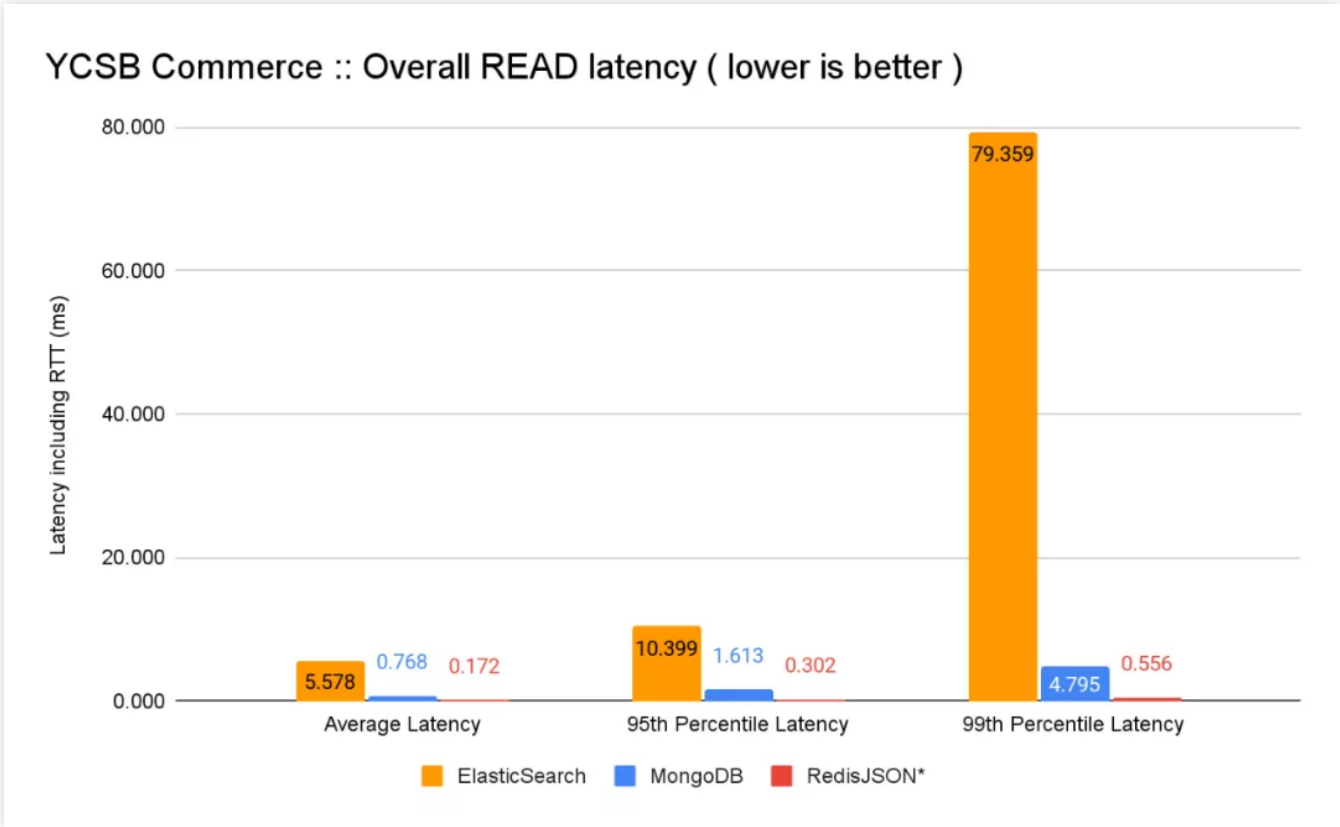
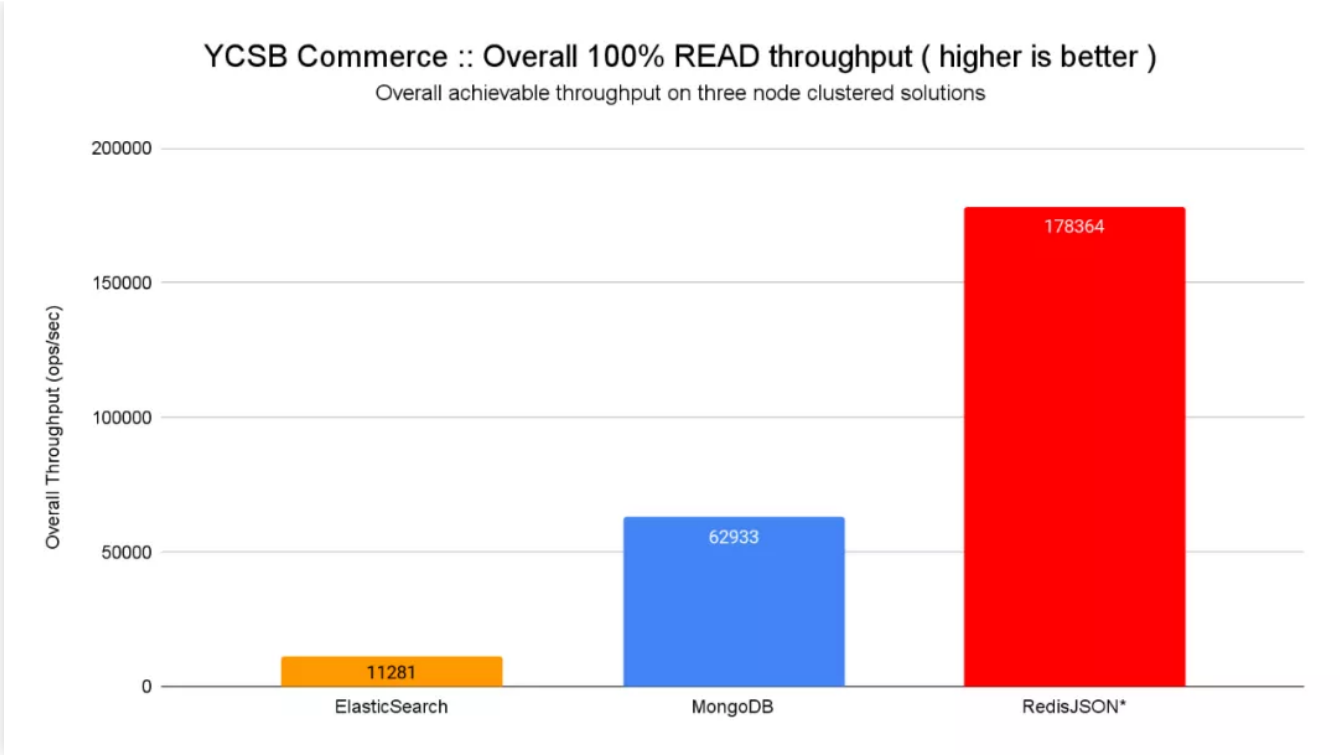




3. 100% 读取基准

与写类似，我们可以观察到 Redis 在读取方面表现最佳，允许读取比 ElasticSearch 多 15.8 倍，比 MongoDB 多 2.8 倍，同时在整个延迟范围内保持亚毫秒级延迟，如下表所示。

在结合延迟和吞吐量改进时，RedisJSON* 比 MongoDB 快 12.7 倍，比 ElasticSearch 快 500 倍以上，用于隔离读取。



4. 混合读/写/搜索基准

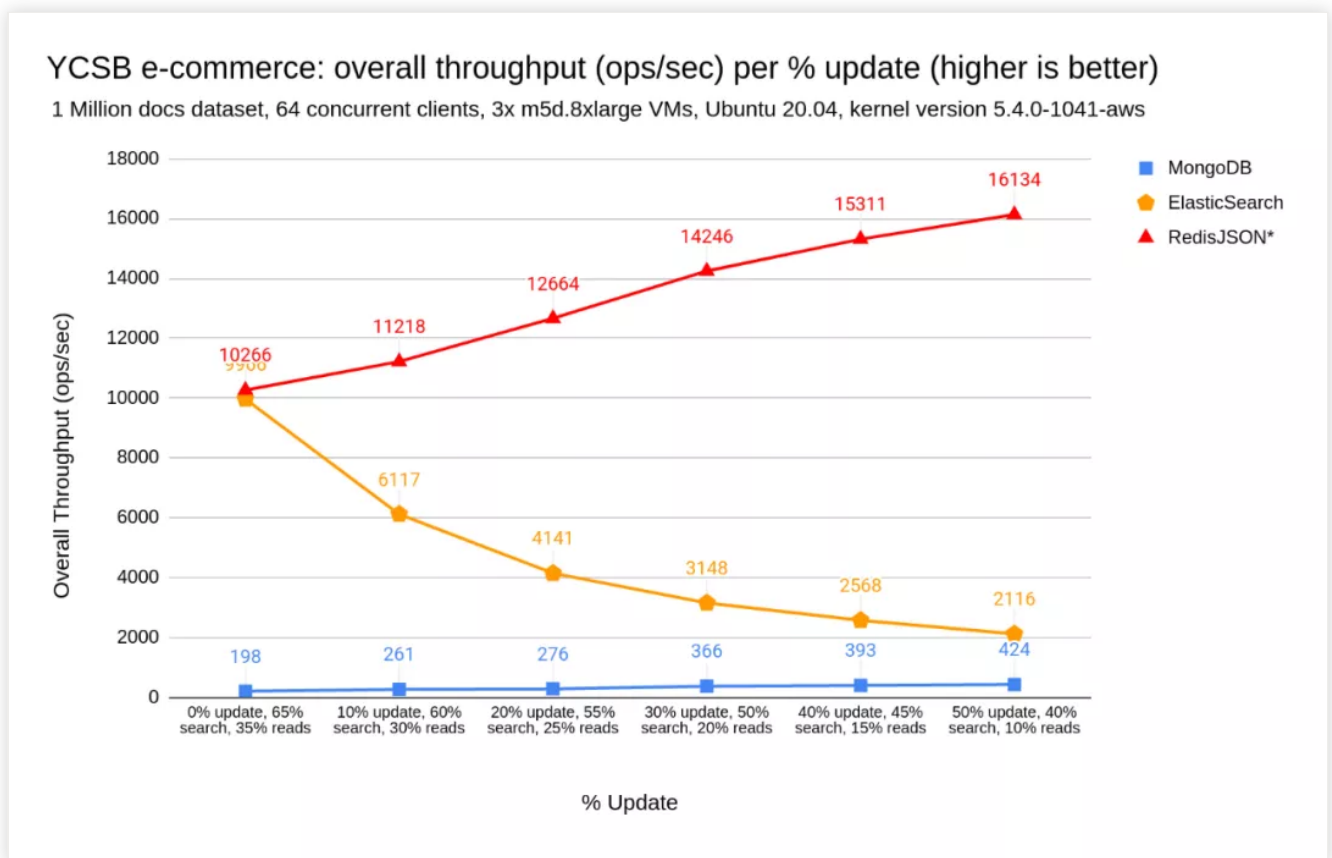
实际应用程序工作负载几乎总是读取、写入和搜索查询的混合。因此，在接近饱和时了解由此产生的混合工作负载吞吐量曲线更为重要。

作为起点，我们考虑了 65% 搜索和 35% 读取的场景，这代表了一个常见的现实世界场景，在该场景中，我们执行的搜索/查询比直接读取更多。65% 搜索、35% 读取和 0% 更新的初始组

合也导致 Elasticsearch 和 RedisJSON* 的吞吐量相等。尽管如此，YCSB 工作负载允许您指定搜索/读取/更新之间的比率以满足您的要求。

“搜索性能”可以指不同类型的搜索，例如“匹配查询搜索”、“分面搜索”、“模糊搜索”等等。我们所做的最初向 YCSB 增加的搜索工作负载仅专注于“匹配查询搜索”，模仿分页的两词查询匹配，按数字字段排序。“匹配查询搜索”是任何启用搜索功能的供应商进行搜索分析的起点，因此，每个支持 YCSB 的数据库/驱动程序都应该能够在其基准驱动程序上轻松启用此功能。

在每个测试变体中，我们添加了 10% 的写入，以按相同的比例混合和减少搜索和读取百分比。这些测试变体的目标是了解每个产品如何处理数据的实时更新，我们认为这是事实上的架构目标，即写入立即提交到索引，读取始终是最新的。



正如您在图表中所看到的，在 RedisJSON* 上不断更新数据和增加写入比例不会影响读取或搜索性能并提高整体吞吐量。对数据产生的更新越多，对 Elasticsearch 性能的影响就越大，最终导致读取和搜索速度变慢。

ElasticSearch 可实现的 ops/sec 从 0% 更新到 50% 的演变，我们注意到它在 0% 更新基准上以 10k Ops/sec 开始，并受到严重影响，减少了 5 倍的 ops/sec，在 50% 更新率基准。

与我们在上述单个操作基准中观察到的类似，MongoDB 搜索性能比 RedisJSON* 和 Elasticsearch 慢两个数量级，MongoDB 的最大总吞吐量为 424 ops/sec，而 RedisJSON* 为

16K 最大 ops/sec。

最后，对于混合工作负载，RedisJSON* 支持的操作数/秒比 MongoDB 高 50.8 倍，比 Elasticsearch 高 7 倍。如果我们将分析集中在混合工作负载期间的每种操作类型的延迟上，与 MongoDB 相比，RedisJSON* 可将延迟降低多达 91 倍，与 Elasticsearch 相比，延迟降低 23.7 倍。

5. 完整延迟分析

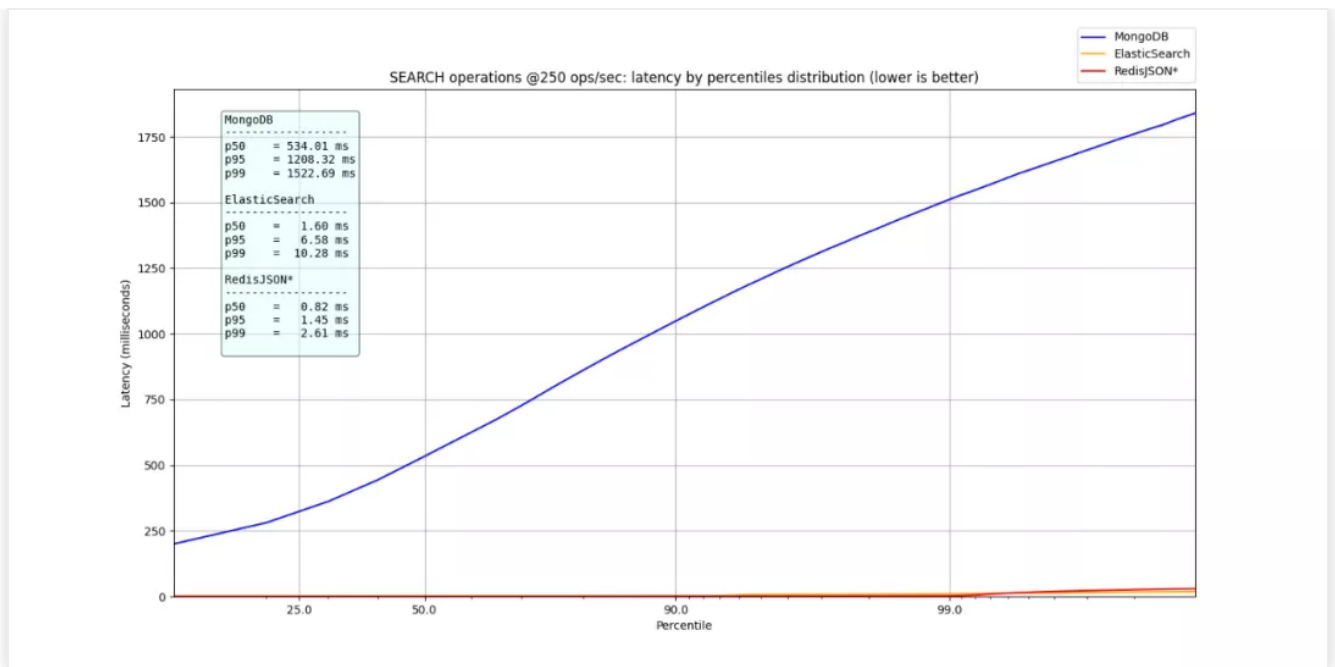
与测量每个解决方案饱和之前产生的吞吐量曲线类似，在所有解决方案通用的可持续负载下进行完整的延迟分析也很重要。这将使您能够了解对于所有已发布操作在延迟方面最稳定的解决方案是什么，以及哪种解决方案不易受到应用程序逻辑引发的延迟峰值的影响（例如，弹性查询缓存未命中）。如果您想更深入地了解我们为什么要这样做，Gil Tene 提供了延迟测量注意事项的深入概述。

- 查看上一节的吞吐量图表，并关注 10% 更新基准以包含所有三个操作，我们做了两种不同的可持续负载变化：
- 250 ops/sec：比较 MongoDB、ElasticSearch 和 RedisJSON*，低于 MongoDB 的压力率。
- 6000 ops/sec：比较 ElasticSearch 和 RedisJSON*，低于 ElasticSearch 压力率。

1 MongoDB 与 ElasticSearch 与 RedisJSON* 的延迟分析

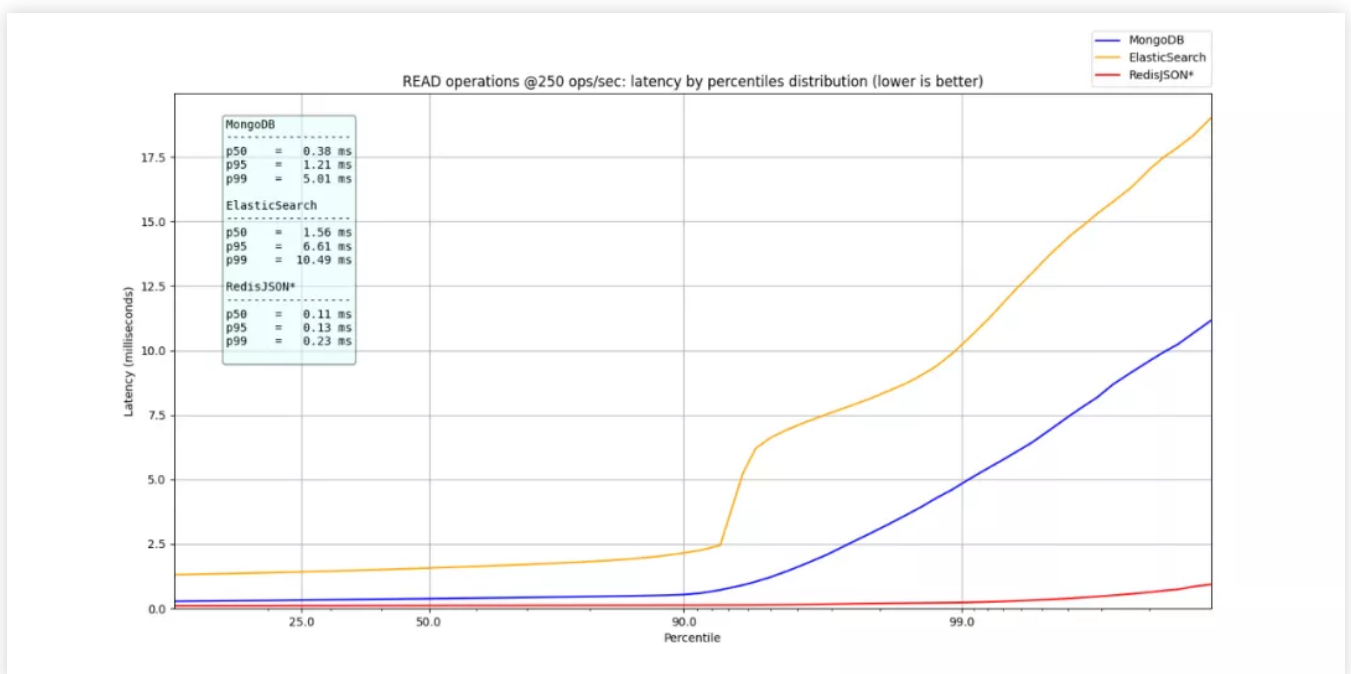
在下面的第一张图片中，展示了从 p0 到 p9999 的百分位数，很明显，在每次搜索时，MongoDB 的表现都远远优于 Elastic 和 RedisJSON*。此外，关注 ElasticSearch 与 RedisJSON*，很明显，ElasticSearch 容易受到较高延迟的影响，这很可能是由垃圾收集 (GC) 触发器或搜索查询缓存未命中引起的。

RedisJSON* 的 p99 低于 2.61 毫秒，而 ElasticSearch p999 搜索达到 10.28 毫秒。

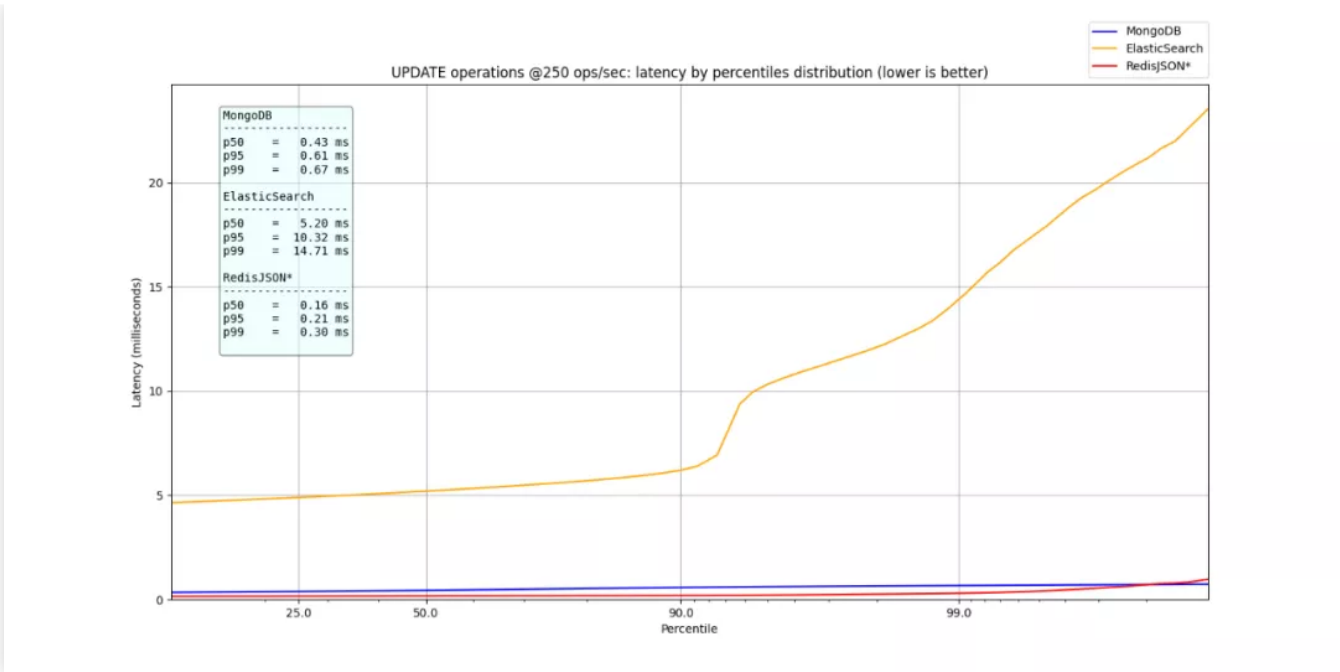


在下面的读取和更新图表中，我们可以看到 RedisJSON* 在所有延迟范围内表现最佳，其次是 MongoDB 和 ElasticSearch。

RedisJSON* 是在所有分析的延迟百分位数上保持亚毫秒级延迟的唯一解决方案。在 p99，RedisJSON* 的延迟为 0.23 毫秒，其次是 MongoDB 的 5.01 毫秒和 ElasticSearch 的 10.49 毫秒。



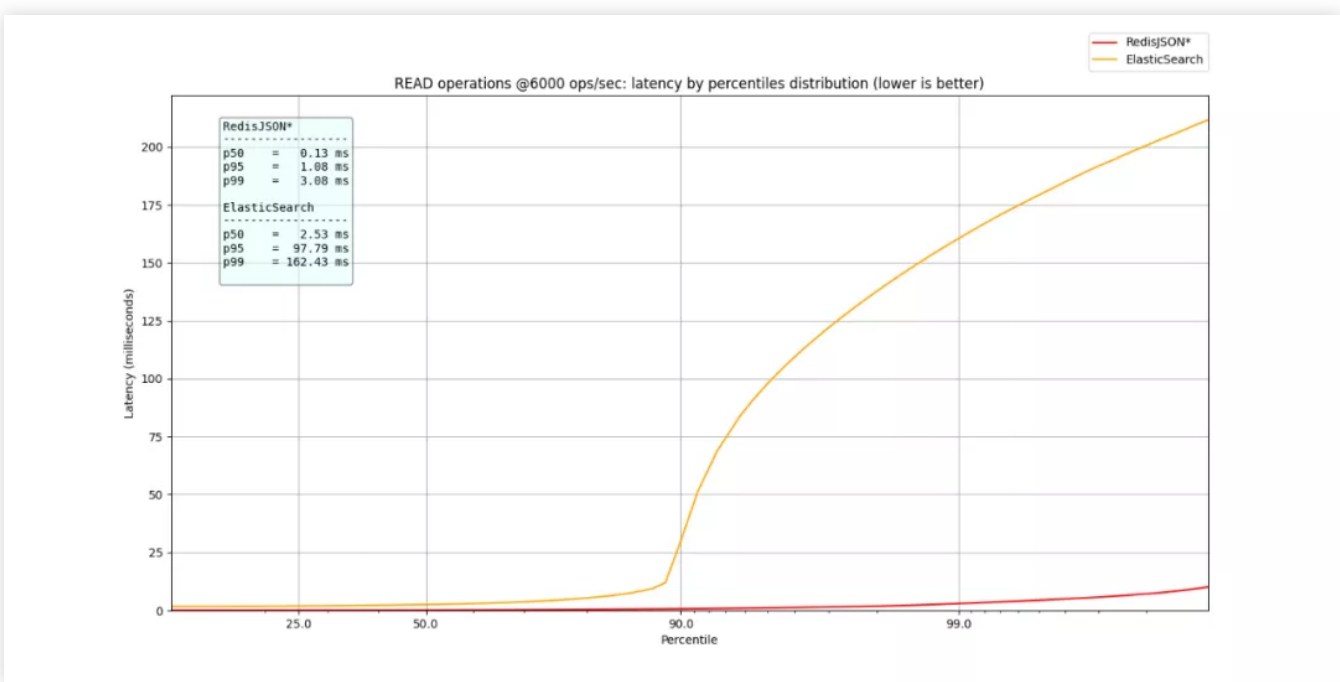
在写入时，MongoDB 和 RedisJSON* 即使在 p99 时也能保持亚毫秒级的延迟。另一方面，ElasticSearch 显示出高尾延迟（> 10 毫秒），这很可能与导致 ElasticSearch 搜索峰值的原因（GC）相同。

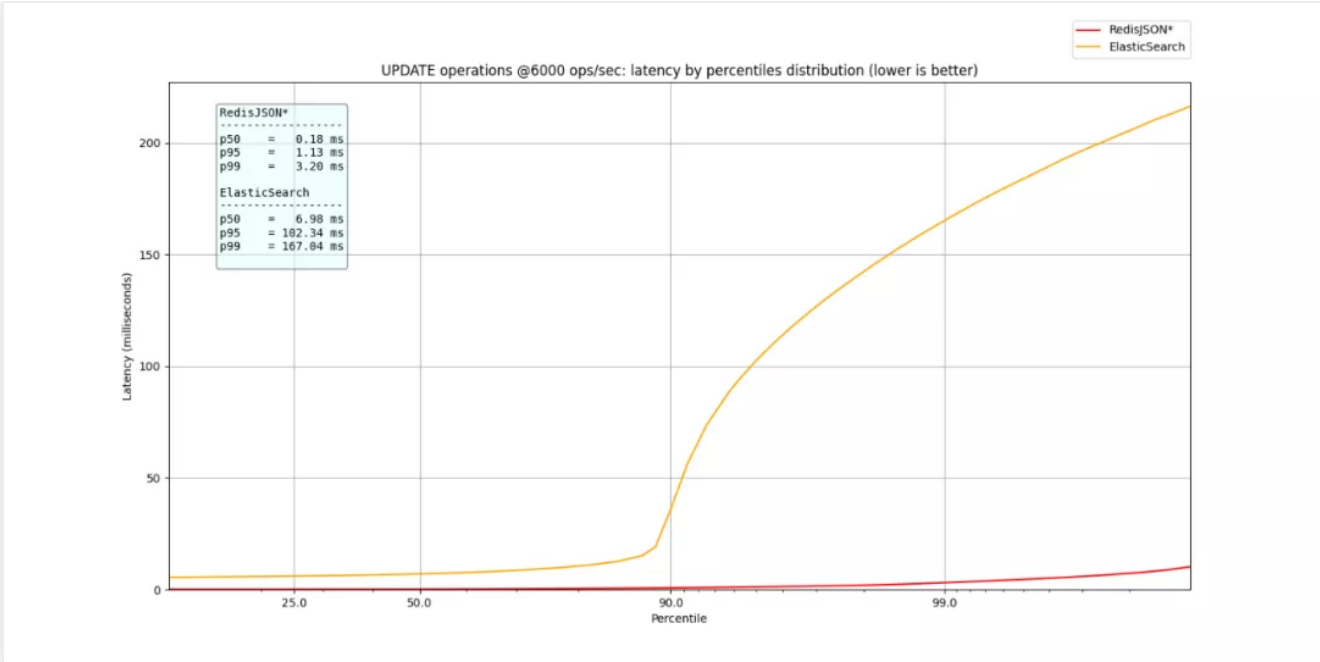


2. ElasticSearch 与 RedisJSON 的延迟分析

仅关注 ElasticSearch 和 RedisJSON*，在保持 6K ops/sec 的可持续负载的同时，我们可以观察到 Elastic 和 RedisJSON* 的读取和更新模式与以 250 ops/sec 进行的分析保持一致。RedisJSON* 是更稳定的解决方案，其 p99 读取时间为 3 毫秒，而 Elastic 的 p99 读取时间为 162 毫秒。

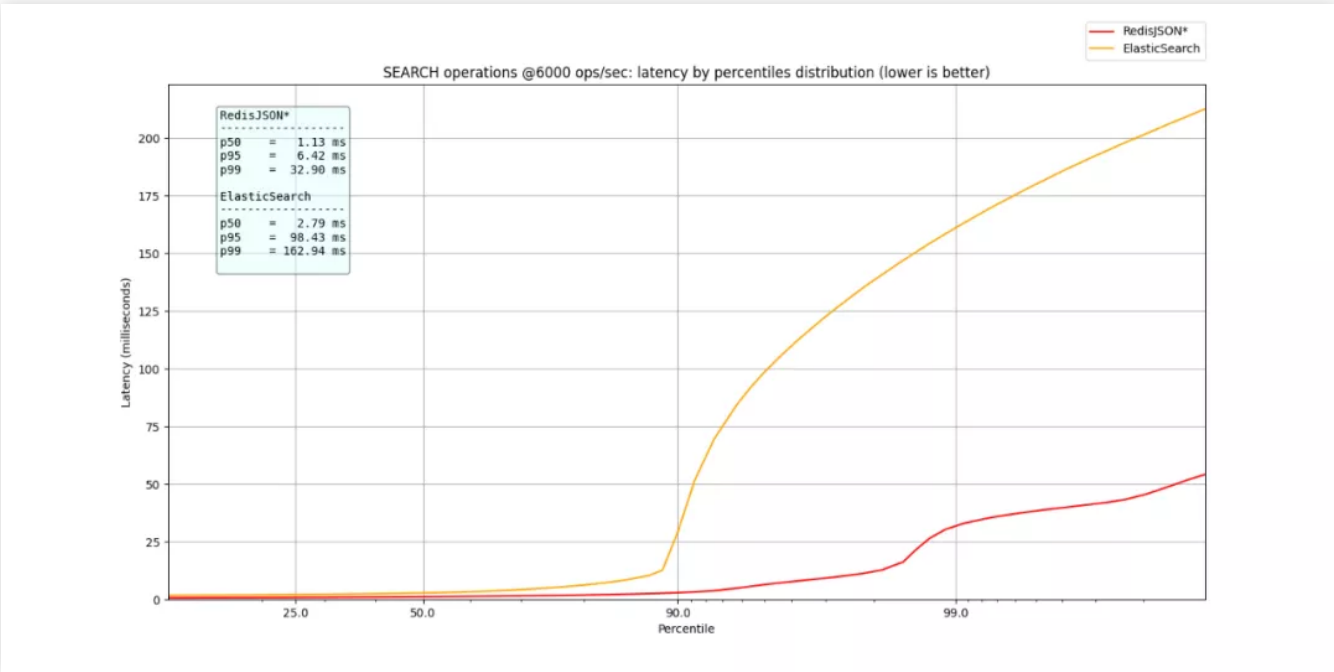
在更新时，RedisJSON* 保留了 3 毫秒的 p99，而 ElasticSearch 则保留了 167 毫秒的 p99。





专注于搜索操作，ElasticSearch 和 RedisJSON* 以个位数 p50 延迟开始（p50 RedisJSON* 为 1.13 毫秒，而 ElasticSearch 的 p50 为 2.79 毫秒），其中 ElasticSearch 付出了 GC 触发和查询缓存未命中的代价在较高的百分位数上，在 $\geq p90$ 百分位数上清晰可见。

RedisJSON* 将 p99 保持在 33 毫秒以下，而 ElasticSearch 上的 p99 百分位数为 163 毫秒，高出 5 倍。



来源：

来源：xiangzhihong8
链接：<https://blog.csdn.net/xiangzhihong8/article/details/121530019>

编程资源库



专注于分享黑科技、黑教程、黑项目...



ID: coderesource



按一下就知道 编程资源库网

喜欢此内容的人还喜欢

为了拿捏 Redis 数据结构，我画了 20 张图
小林coding



Redlock——Redis集群分布式锁
程序员大白

