

Elasticsearch 5.0.0版本新特性介绍

Elasticsearch 5.0.0在2016年10月26日发布，该版本基于Lucene 6.2.0，这是最新的稳定版本，并且已经在Elastic Cloud上完成了部署。Elasticsearch 5.0.0是目前最快、最安全、最具弹性、最易用的版本，此版本带来了一系列的新功能和性能优化。

[ElasticSearch 5.0.0 release Note](#) [点击下载ElasticSearch 5.0.0](#) [阅读最新文档](#)

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

索引性能

得益于多项改进，其中包括更好的数值型数据结构，索引吞吐量大幅提升。根据应用场景的不同，索引吞吐量提升在25%到80%之间。

Ingest节点

向Elasticsearch添加数据更简单了。Logstash是一个强大的工具，而一些较小的用户只需要过滤器，不需要它所提供的众多路由选项。因此，Elastic将一些最流行的Logstash过滤器（如grok、split）直接在Elasticsearch中实现为处理器。多个处理器可以组合成一个管道，在索引时应用到文档上。

Painless脚本

Elasticsearch中很多地方用到了脚本，而出于安全考虑，脚本在默认情况下是禁用的，这令人相当失望。为此，Elastic开发了一种新的脚本语言Painless。该语言更快、更安全，而且默认是启用的。不仅如此，它的执行速度是Groovy的4倍，而且正在变得更快。Painless已经成为默认脚本语言，而Groovy、Javascript和Python都遭到弃用。要了解有关这门新语言的更多信息，请点击[这里](#)。

新数据结构

Lucene 6带来了一个新的Points 数据结构K-D树，用于存储数值型和地理位置字段，彻底改变了数值型值的索引和搜索方式。基准测试表明，Points将查询速度提升了36%，将索引速度提升了71%，而占用的磁盘和内存空间分别减少了66%和85%（参见“在5.0中搜索数值”）。

搜索和聚合

借助即时聚合，Kibana图表生成速度显著提升。Elastic用一年的时间对搜索API进行了重构，Elasticsearch现在可以更巧妙地执行范围查询，只针对已经发生变化的索引重新计算聚合，而

不是针对每个查询从头开始重新计算。在搜索方面，默认的相关性计算已经由TF/IDF换成了更先进的BM25。补全建议程序经过了完全重写，将已删除的文档也考虑了进来。

更友好

Elasticsearch 5.0更安全、更易用。他们采用了“尽早提示”的方法。如果出现了问题，则新版本会及早给出提示。例如，Elasticsearch 5.0会严格验证设置。如果它不能识别某项设置的值，就会给出提示和建议。不仅如此，集群和索引设置现在可以通过null进行解除。此外，还有一些其他的改进，例如，rollover和shrink

API启用了一种新的模式来管理基于时间的索引，引入新的cluster-allocation-explain API，简化索引创建。

弹性

Elasticsearch分布式模型的每一部分都被分解、重构和简化，提升了可靠性。集群状态更新现在会等待集群中的所有节点确认。如果一个“复制片（replica shard）”被“主片（primary）”标记为失败，则主片会等待“主节点（master）”的响应。索引现在使用数据路径中的UUID，而不是索引名，避免了命名冲突。另外，Elasticsearch现在进行启动检查，确保系统配置没有问题。配置比较麻烦，但如果只是试用，开发人员也可以选择localhost-only模式，避免繁琐的配置。另外，新版本还增加了断路器及其他一些软限制，限制请求使用的内存大小，保护集群免受恶意用户攻击。

此外，该版本还提供了一个底层的Java REST/HTTP客户端，可以用于监听、日志记录、请求轮询、故障节点重试等。它使用Java 7，将依赖降到了最低，比Transport客户端的依赖冲突少。而在基准测试中，它的性能并不输于Transport客户端。不过，这是一个底层客户端，目前还没有提供任何查询构建器或辅助器。它的输入参数和输出结果都是JSON。

需要注意的是，该版本引入了许多破坏性更改，好在他们提供了一个迁移辅助插件，可以帮助开发人员从Elasticsearch 2.3.x/2.4.x迁移到Elasticsearch 5.0。如果是从更早的Elasticsearch版本向最新的5.0版本迁移，则请查阅升级文档。

本博客文章除特别声明，全部都是原创！
转载本文请加上：转载自过往记忆（<https://www.iteblog.com/>）
本文链接：【】（）