只是升级：PostgreSQL 12如何提升性能

译者: 陈雁飞，李红艳

### 作者简介

**Jonathan S. Katz**：crunchydata公司数据库工程师

### 译者简介

**陈雁飞**：开源PostgreSQL爱好者，一直从事PostgreSQL数据库运维工作

**李红艳，**衡数软件技术负责人，有百货公司（超过40家）、视光集团和知名电商新零售大数据分析经验。

PostgreSQL 12，“世界上最先进的开源关系型数据库”的最新版本，如果没有其他问题，将在接下来的几周时间内正式发布。该项目每年新增大量新数据库功能，坦白地说，这一点非常令人惊讶。同时也是我参与PostgreSQL社区的主要原因之一。

在我看来，今年和前几年有所不同，PostgreSQL 12并不仅仅是包含每个人都知道的一个或两个独立特性，并且说这个就是“特性”版本（我能想到的最近例子就是分区和并行查询）。我半开玩笑地说，这个版本的主题应该是“PostgreSQL 12：现在更稳定”——当然，当你为你的企业管理关键任务数据时候，这一点是非常重要的。

不过，我相信这个版本不仅仅是稳定：PostgreSQL 12中的很多功能和特性增强会让应用程序更块地运行，这只需要升级就可以，并不需要其它任何额外操作。（可能需要重建索引，但是在这个版本中重建索引不会像过去那样痛苦）

只需要执行升级操作，就能看到明显地提升，这一点对于升级PostgreSQL来说非常不错。几年前，当我分析从PostgreSQL 9.4升级到PostgreSQL 10的时候，我观测到升级后基础应用程序运行地更快了，主要是利用了在PostgreSQL10中引入的改进并行查询。我几乎不费吹灰之力就得到了性能提升（我设置了max\_parallel\_workers配置参数）。

通过简单地升级就可以让应用程序能够更好的运行，这一点对于用户来说是非常好的体验，并且对于维持现有的用户非常重要，也将有越来越多的人接受PostgreSQL。

那么，PostgreSQL 12如何仅仅通过升级就使得应用程序更好地运行？请继续阅读下面的内容！

# 索引重大改进

索引是任何数据库系统的关键部分；它有助于快速检索信息。PostgreSQL使用的基本索引系统叫作B-tree，这是一种针对存储系统优化的索引类型。

通常想当然地会认为语句CREATE INDEX ON some\_table (some\_column)非常简单，其实PostgreSQL完成了大量工作用来保证索引处于最新状态，因为它存储的值会不断地被插入、更新或者删除。通常，这个语句可以正常工作。

但是，PostgreSQL索引的一个问题是它们可能会[膨胀](https://info.crunchydata.com/blog/checking-for-postgresql-bloat)并占用额外的磁盘空间，这一问题也会影响到对数据的检索和更新性能。这里，我所说的“膨胀”是指索引结构维护方式的低效性，这可能与[VACUUM](https://www.postgresql.org/docs/current/sql-vacuum.html)移除垃圾元组有关，也可能没有关系（对于这个分析，要感谢[Peter Geoghegan](https://twitter.com/petervgeoghegan)）。在一个大量修改索引的场景下，索引膨胀表现地会更加明显。

PostgreSQL 12对B-tree索引做了显著的改进，并且从TPC-C测试实验的结果看，平均降低了40%的空间利用率。这不仅仅减少了花费在维护B-Tree索引（比如写）上的总时间，而且由于索引大小的减少，也有助于更快速的检索数据。

通常在OLTP（[在线事务处理](https://en.wikipedia.org/wiki/Online_transaction_processing)）系统中，应用程序会对表进行大量的更新，这些应用应该在磁盘利用率和查询性能方面看到明显的改进。同时，较少的磁盘利用率意味着在升级基础设施前，数据库将有更多空间用于业务扩展。

根据你的升级策略，可能需要重建B-tree索引来获得这些改进点（例如，[pg\_upgrade](https://www.postgresql.org/docs/current/pgupgrade.html)不会自动重建索引）。在之前的PostgreSQL版本中，如果表上有大的索引，重建可能会导致停机，因为索引重建将阻塞其它对表的修改操作。但是在PostgreSQL 12中有另外一个亮点：现在在PostgreSQL中，可以使用[REINDEX CONCURRENTLY](https://www.postgresql.org/docs/12/sql-reindex.html)命令进行索引的并发重建，因此现在执行索引重建不会导致潜在停机可能！

在PostgreSQL 12中对索引基础结构的其他部分也进行了改进。其中一个“重要可靠性”方面是[预写式日志](https://www.postgresql.org/docs/current/wal-intro.html)，也称为WAL。预写式日志发挥重要作用，因为它记录了PostgreSQL中发生的每个事务操作，它是安全崩溃和复制特性的基础，同时也被应用程序用于归档和[时间点恢复](https://info.crunchydata.com/blog/pgbackrest-point-in-time-recovery-using-crunchy-postgresql-operator)。预写式日志意味着需要将其他信息写入到磁盘中，着可能会对性能产生影响。

在PostgreSQL 12中，当建立GiST、GIN和SP-GiST索引的时候，减少了生成WAL记录的开销。这对多个方面产生好处，包括更少的空间用于记录这些WAL记录，以及更快地完成崩溃恢复或者时间点恢复。如果你的应用程序使用到这些索引类型中的任何一种（例如，被PostGIS驱动的地理应用程序中大量使用GiST索引），这又是一个新的特性，在不需要额外操作的情况下对程序产生显著影响。

## 分区更大、更好、更快

PostgreSQL 10中引入了[声明式分区](https://www.postgresql.org/docs/current/ddl-partitioning.htm)。PostgreSQL 11使它更易于使用，PostgreSQL 12允许真正地扩展分区。

PostgreSQL 12对分区系统进行了显著地性能改进，特别是如何处理具有上千个分区的表。例如，一个查询仅涉及上千个分区中的几个分区，它的执行速度会大大加快。除了看到这些查询类型性能上的提升外，将同样看到在对具有大量分区的表上插入速度的改进。

在PostgreSQL 12中，同样提升了使用COPY命令将数据批量加载（[JSON提取](https://info.crunchydata.com/blog/fast-csv-and-json-ingestion-in-postgresql-with-copy)例子）到分区表中的性能。以前使用COPY已经很快了，PostgreSQL 12让这一操作变得更快。

所有上面的改变使PostgreSQL可以存储更大的数据集，同时可以更加容易地检索到数据，甚至可以更好地完成这一操作。对于使用很多分区的应用程序，例如记录时间序列数据，只需要通过升级就可以看到明显地性能提升。

虽然它可能不属于“只升级就更好”的范畴，但是PostgreSQL 12允许创建引用分区表的外键，从而消除了可能在分区中遇到的“陷阱”。

# WITH查询的巨大提升

当提交完[内联公共表达式补丁](https://git.postgresql.org/gitweb/?p=postgresql.git;a=commitdiff;h=608b167f9f9c4553c35bb1ec0eab9ddae643989b)后（也称为CTE，WITH查询），我就迫不及待地写了一篇文章介绍它对[PostgreSQL应用开发人员的帮助](https://info.crunchydata.com/blog/with-queries-present-future-common-table-expressions)。这是其中一项功能，如果使用到了CTEs，你将看到应用程序变得更快。

我经常发现对于初学SQL的开发人员喜欢使用CTEs：如果以某种方式编写它们，就会感觉像在编写命令程序一样。但我还是喜欢重写那些查询获得性能提升，而不是通过CTEs的方式。哎，现在这种情况已经一去不复返了。

PostgreSQL 12现在允许对某一类CTE进行内联操作，比如对于在查询中仅仅使用一次的查询，这一类是没有其他影响的（查询）。如果已经收集了使用CTE查询语句中有关查询数量的统计信息，那么我将查询它们，大部分语句都属于这种情况。这将帮助开发人员编写出可读性更高的代码，并且同样可以高效地运行。

更好的情况是PostgreSQL 12将优化这种SQL的执行，而我们不需要任何其他额外工作。虽然我可能不需要继续优化这种查询模式，但PostgreSQL持续改进查询优化显然更好。

# 默认采用Just-in-Time（JIT）

PostgreSQL 12系统支持[LLVM](https://llvm.org/)，默认将启用JIT（即时编译）。[JIT](https://www.postgresql.org/docs/current/jit.html)除了支持一些内部操作符外，查询选择列（写完“ELECT”之后的内容）有表达式（例如，类似“x+y“这样一个简单表达式）的查询语句、使用聚集函数、WHERE子句中包含有表达式以及类型子句，均可以利用JIT来提高性能。

由于在PostgreSQL 12中默认启用了JIT，因此无需执行任何操作就可以看到性能提升，但是我建议在引入JIT的PostgreSQL 11上测试应用程序，以评估查询的执行情况并查询是否需要其他调整操作。

# PostgreSQL 12中其他新特性怎么样？

PostgreSQL 12中有很多我所期待的新增加的特性，包括使用标准SQL/JSON路径表达式构建JSON数据、使用clientcert=verify-full设置项提供多因子的身份验证以及生成列等等，我对这些特性都特别感兴趣。这里有另外一篇博文介绍这些。

就像我过去使用PostgreSQL 10的经历一样，我相信PostgreSQL 12同样提供了类似的能力，只需要通过升级就提提高整体体验。当然，这可能因人而异：我再切换PostgreSQL 10的时候，首先在类似生产系统环境下测试应用程序，然后才进行切换。尽管PostgreSQL 12像我说的那样“现在更加稳定”，但是在应用程序投入生产环境前最好对其进行完整的测试。

# 原文地址

<https://info.crunchydata.com/blog/just-upgrade-how-postgresql-12-can-improve-your-performance>