## Cassandra系统简介

Cassandra是一套开源分布式NoSQL数据库系统。它最初由Facebook开发，用于储存收件箱等简单格式数据，集GoogleBigTable的数据模型与Amazon Dynamo的完全分布式的架构于一身Facebook于2008将 Cassandra 开源，此后，由于Cassandra良好的可扩展性，被Digg、Twitter等知名Web 2.0网站所采纳，成为了一种流行的分布式结构化数据存储方案。

Cassandra是一个混合型的非关系的数据库，类似于Google的BigTable。其主要功能比Dynamo （分布式的Key-Value存储系统）更丰富，但支持度却不如文档存储MongoDB（介于关系数据库和非关系数据库之间的开源产品，是非关系数据库当中功能最丰富，最像关系数据库的。支持的数据结构非常松散，是类似json的bjson格式，因此可以存储比较复杂的数据类型）。Cassandra最初由Facebook开发，后转变成了开源项目。它是一个网络社交云计算方面理想的数据库。以Amazon专有的完全分布式的Dynamo为基础，结合了Google BigTable基于列族（Column Family）的数据模型。P2P去中心化的存储。很多方面都可以称之为Dynamo 2.0。

Cassandra的主要特点就是它不是一个数据库，而是由一堆数据库节点共同构成的一个分布式网络服务，对Cassandra 的一个写操作，会被复制到其他节点上去，对Cassandra的读操作，也会被路由到某个节点上面去读取。对于一个Cassandra群集来说，扩展性能是比较简单的事情，只管在群集里面添加节点就可以了。

和其他数据库比较，有三个突出特点：

（1）模式灵活

使用Cassandra，像文档存储，你不必提前解决记录中的字段。你可以在系统运行时随意的添加或移除字段。这是一个惊人的效率提升，特别是在大型部署上。

（2）可扩展性

Cassandra是纯粹意义上的水平扩展。为给集群添加更多容量，可以指向另一台电脑。你不必重启任何进程，改变应用查询，或手动迁移任何数据。

（3）多数据中心

你可以调整你的节点布局来避免某一个数据中心起火，一个备用的数据中心将至少有每条记录的完全复制。

## Cassandra调优系统设计

Cassandra调优系统系统的设计理念为：generate and test（产生与测试），如图所示。在产生阶段，我们可以使用各种优化方法生成新的参数，比如使用模拟退火方法、遗传编程等等。在测试阶段，我们使用具体的测试方法对新参数进行验证，依据验证结果的反馈和记录，我们进行新一轮的产生和测试，从而不断的迭代。具体的实施方法我们在关键技术中进行介绍。



图 cassandra调优系统设计思想

## Cassandra调优系统关键技术

**Cassandra参数优化的难点**

Cassandra系统参数调优与hadoop和spark系统参数调优有所不同，cassandra系统参数修改要想生效，必须满足两个条件：

1. 必须修改conf目录下的cassandra.yaml文件，没有其他修改方式。
2. 每次修改必须重新启动cassandra系统。

因此，cassandra系统参数调优问题的一大难点就在于如何修改参数并使得参数生效。

**Cassandra参数调优算法**

针对cassandra系统的特点，我们给出了基于generate and test策略、结合迭代思想和模拟退火优化策略的算法，算法概括如下：

|  |
| --- |
| 输入：   1. 模拟退火算法参数（冷却温度等参数） 2. 用户设定的迭代次数阈值Iteration\_num   输出：优化参数 |
| 1. Counter = 0 2. 参数集合Ρ = {} 3. 产生优化初始参数p 4. 测试参数p的实际表现 5. **while** 模拟退火算法终止条件未满足 **and** Counter < Iteration\_num: 6. Counter += 1 7. 使用模拟退火算法得到修改后的参数p’,并保证p’之前不存在于集合Ρ中 8. 测试参数p’的实际表现 9. Ρ∪= p’ 10. **end while** 11. **if** 模拟退火算法终止 12. **return** 模拟退火算法选定的参数 13. **return** Ρ中表现最好的参数 |

第3行和第7行均为算法generate阶段，第4和第8行为算法test阶段。算法添加了迭代次数以满足实际的应用效率需求，如果模拟算法终止条件得到满足，则只需返回优化后的参数即可，如果算法迭代次数达到了用户设定迭代次数的阈值，则需要停止模拟退火算法并返回到目前为止表现最好的参数。

**调优设计框架**

在上一节中，我们介绍了调优算法的具体设计，并介绍了generate and test算法的思想，generate阶段可以由调优算法实现，本节详细介绍如何进行test阶段。

test阶段需要完成以下任务：

1. 修改cassandra配置文件cassandra.yaml
2. 重启cassandra系统
3. 使用特定任务进行测试

具体实现步骤如下：

|  |
| --- |
| 1. 使用程序修改cassandra.yaml中的相关参数 2. 调用脚本重启cassandra系统 3. 对调优系统进行sleep操作，以留给时间供cassandra重启，否则可能出现cassandra系统还没有恢复启动就进行测试导致的错误。 4. 使用特定任务进行测试，并将运行结果记录下来 |

具体操作可以用如下流程图表示：



图 cassandra调优系统运行流程

可以看出，generate and test形成了一个逐步迭代的闭环，用于优化参数的搜索。此外，generate阶段其实只依赖于具体的优化算法，我们可以使用不同的优化算法，不限于模拟退火。

## 优化工具与用户的交互形式

**Cassandra优化工具的部署**

赋予调优系统权限

（1）调优系统中自带几个.sh脚本文件，需要赋予应用权限

（2）若想修改cassandra.yaml，必须赋予用户修改此配置文件的权限

权限示例：

(ubuntu系统下) chmod -R xxx.sh

**用户的交互形式**

运行调优程序

在命令行中直接输入：python optimize\_wite\_iteration.py