

Apache ShardingSphere 与 openGauss 的生态对接

吴伟杰
May 12, 2022



吴伟杰

Apache ShardingSphere PMC
SphereEx 基础设施研发工程师

- 负责在 ShardingSphere 中实现 openGauss 协议
- 在 16 节点达成 1000 万 tpmC 的目标中参与 ShardingSphere 性能优化
- 目前专注于 ShardingSphere-Proxy 研发及 ShardingSphere 代码性能优化



目录

01

Apache ShardingSphere 介绍

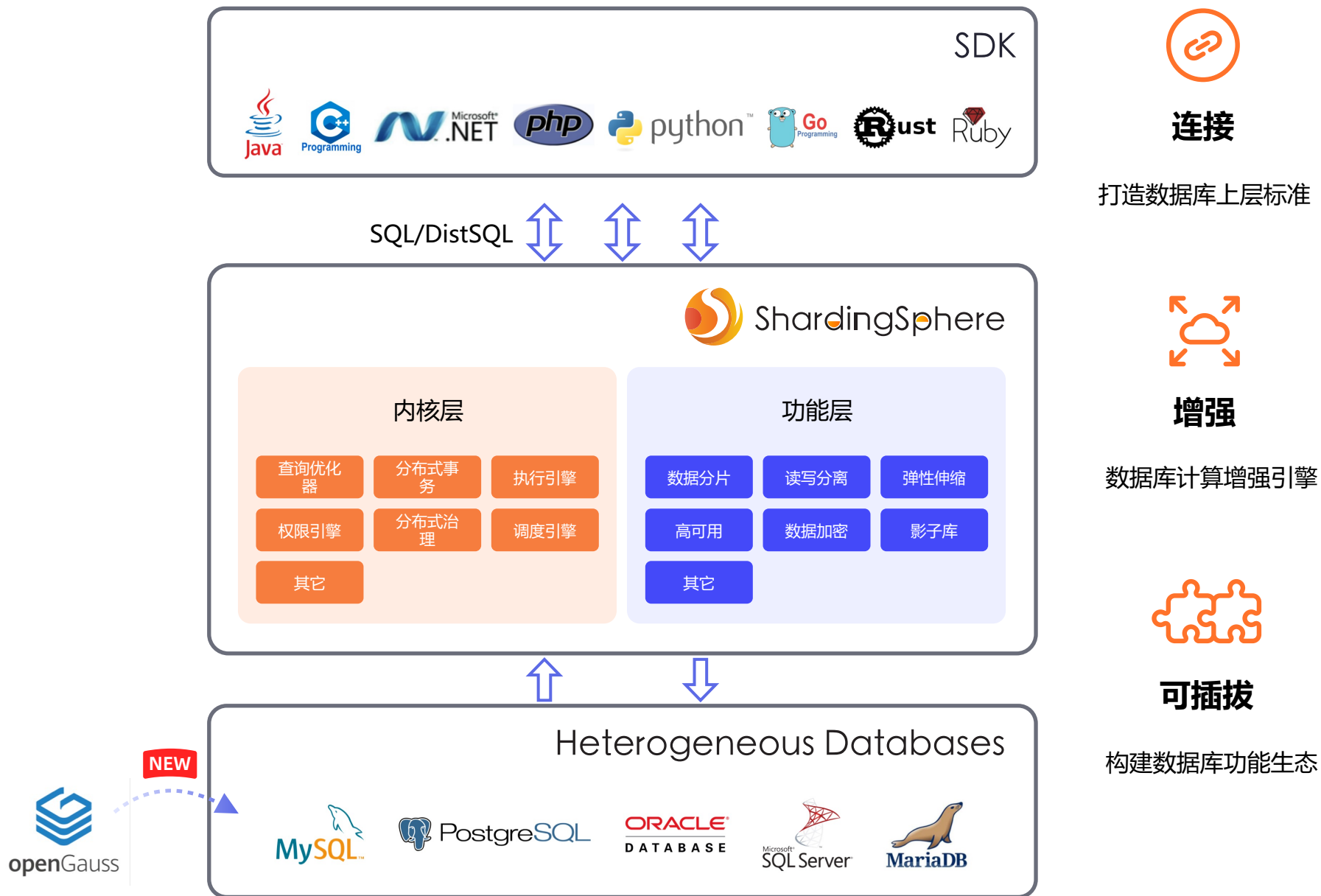
02

ShardingSphere 与 openGauss 的生态对接

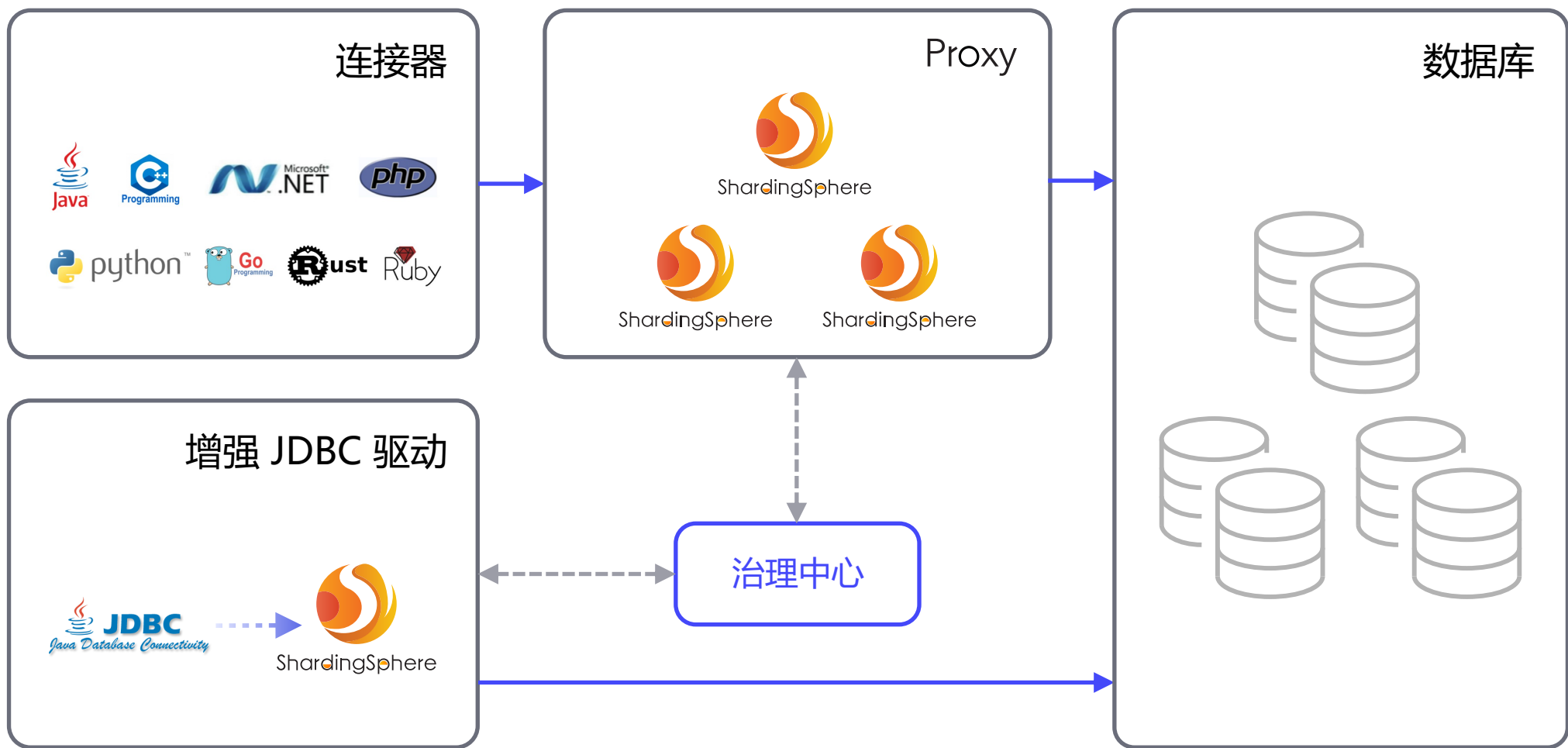
03

ShardingSphere 与 openGauss 的性能磨合

Apache ShardingSphere 能力



Apache ShardingSphere 部署架构

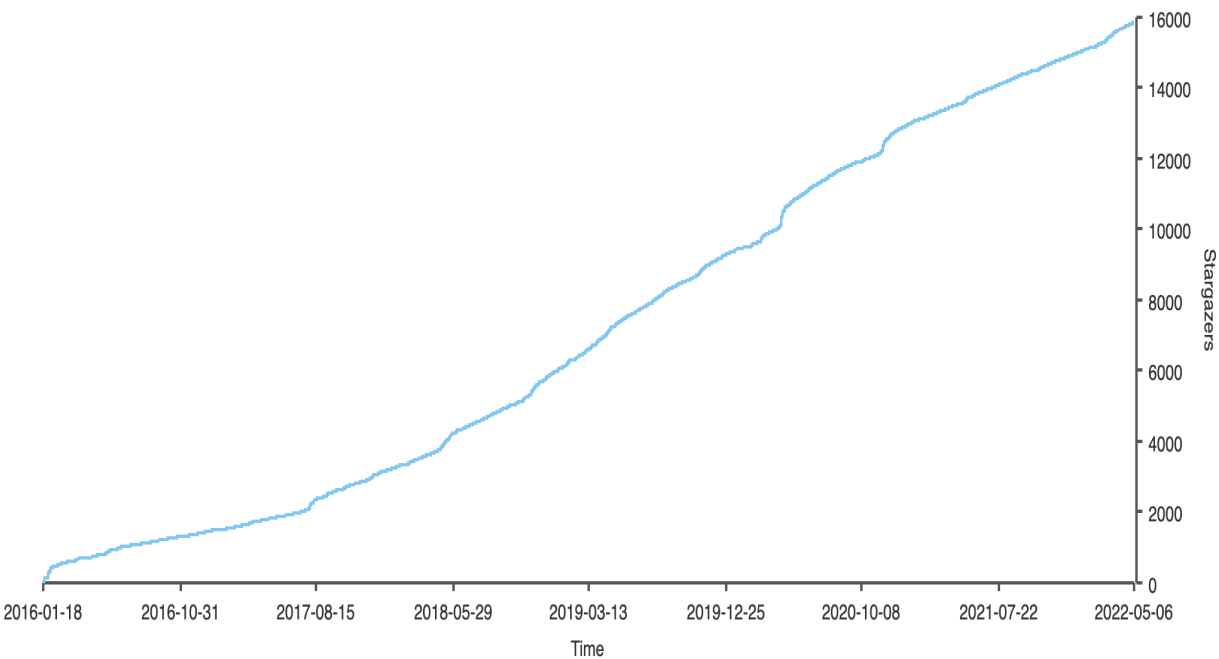


←---→ 代表控制流量

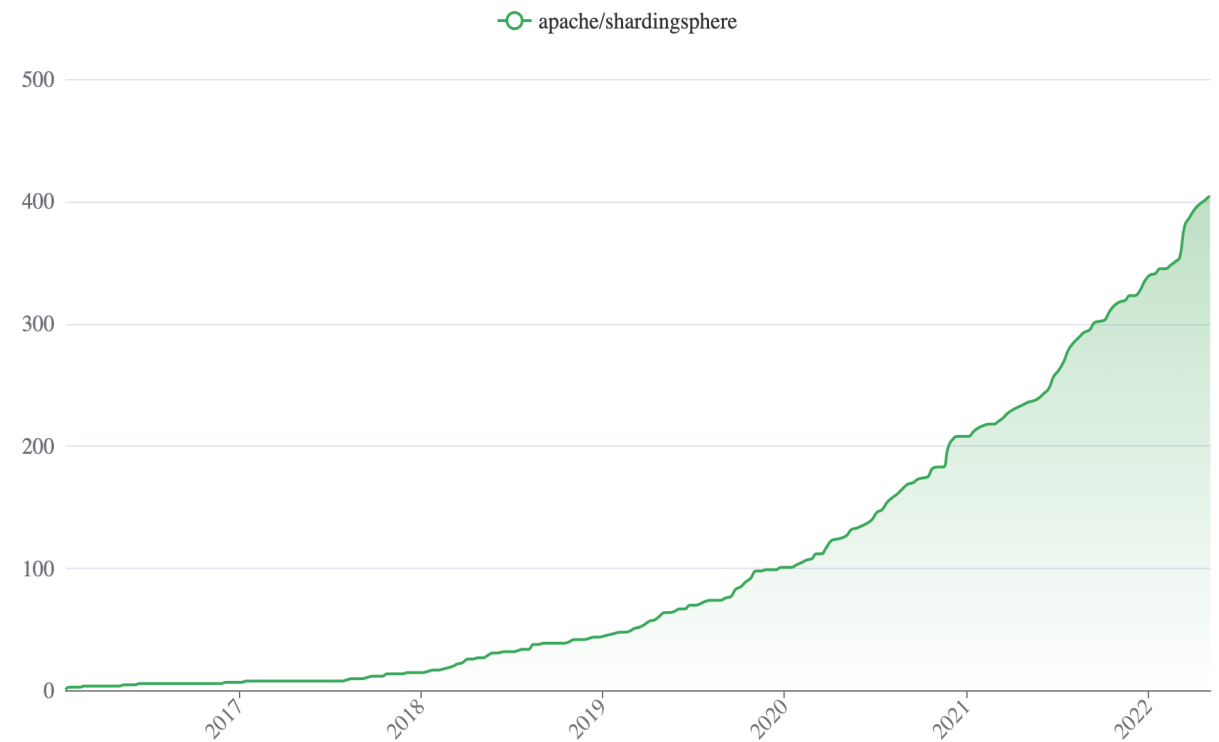
—→ 代表业务流量

Apache ShardingSphere 社区

- Stars 约 **16k**
- Contributors 超过 **400** 人



Contributor Over Time



Apache ShardingSphere Stars 与 Contributors 增长时间线

Apache ShardingSphere 合作伙伴&用户

在 ShardingSphere 官网登记使用的公司超过 **190** 家，包括但不限于：





目录

01

Apache ShardingSphere 介绍

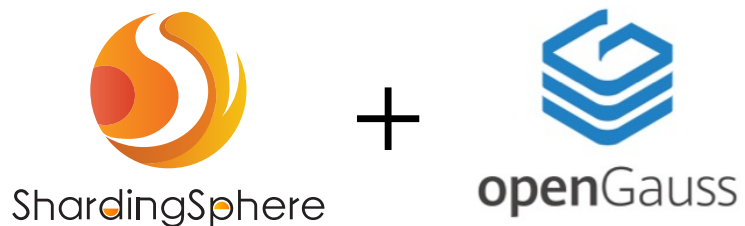
02

ShardingSphere 与 openGauss 的生态对接


03


ShardingSphere 与 openGauss 的性能磨合


openGauss 对接到 ShardingSphere 生态的价值




ShardingSphere 与 openGauss 生态对接历程

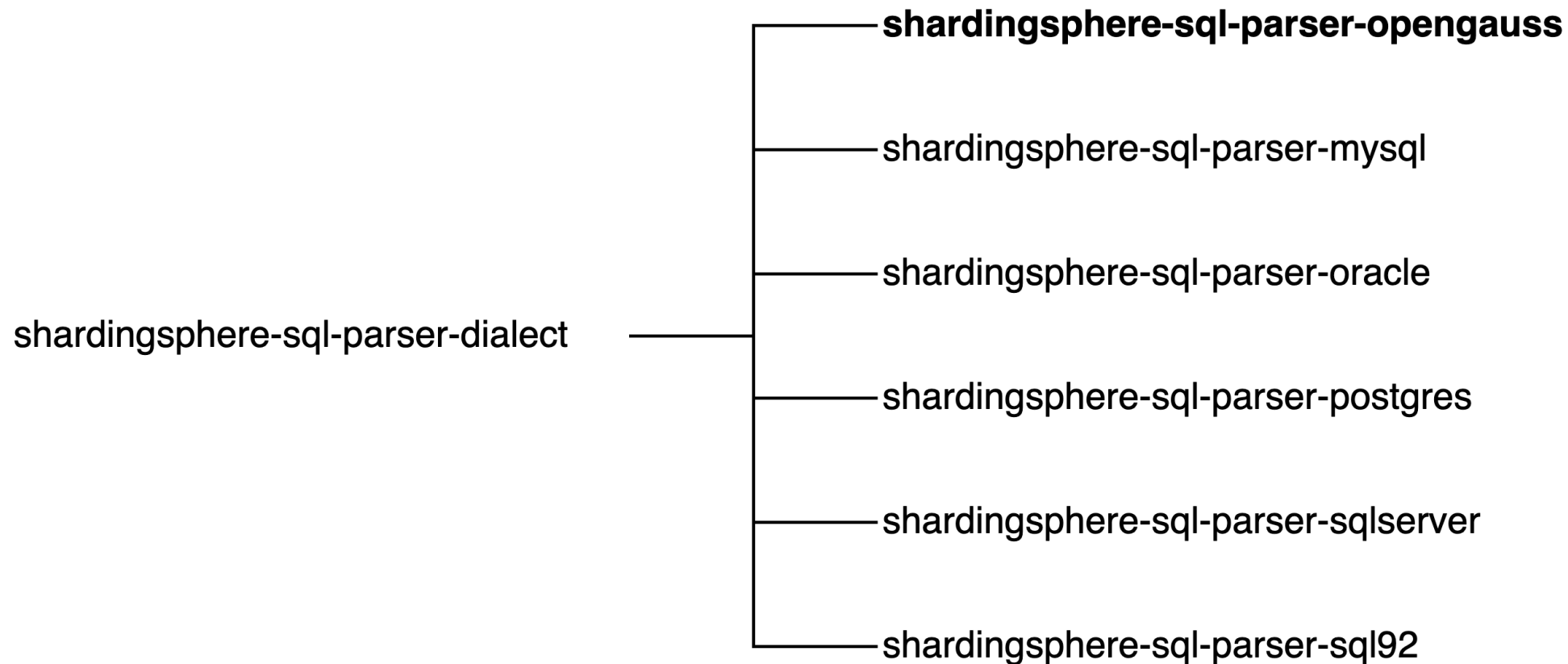
 **ShardingSphere 5.0.0**
支持 openGauss SQL 方言
支持 openGauss 协议
弹性伸缩支持 openGauss

 **ShardingSphere 5.0.0 之前的版本**
间接支持 openGauss

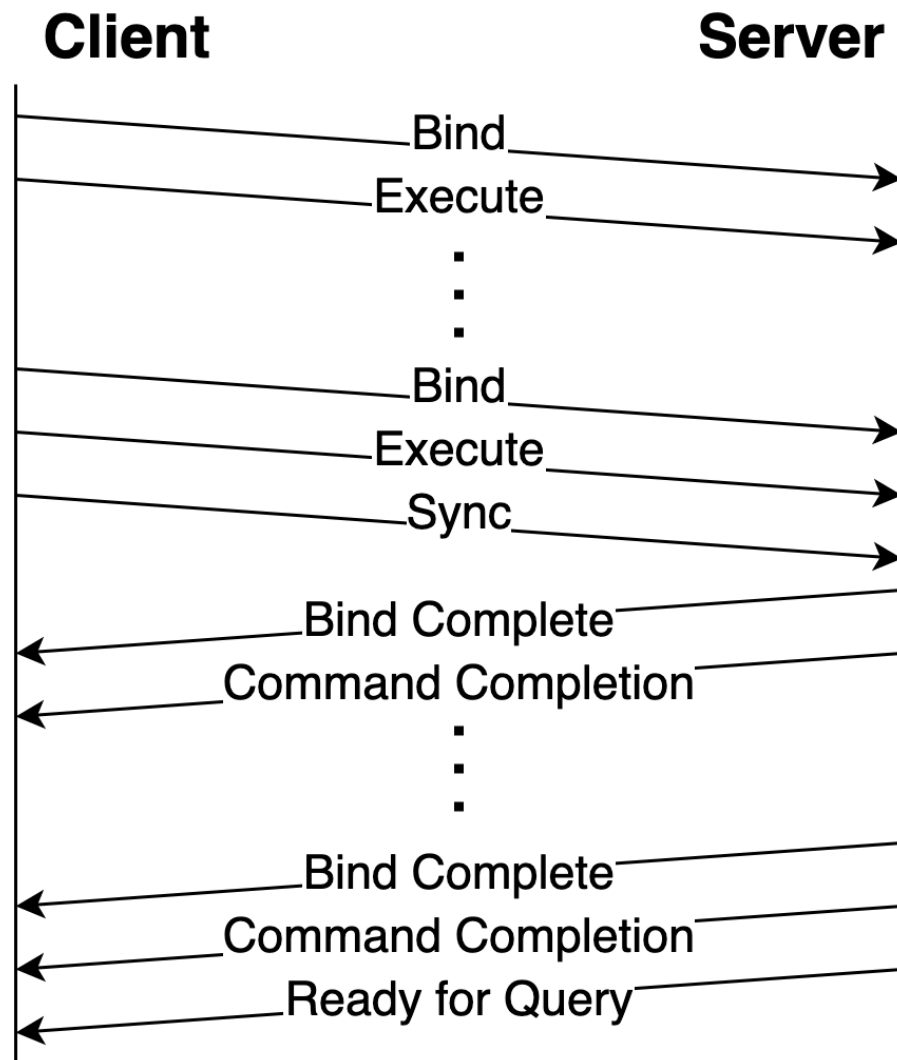
 **ShardingSphere 5.1.0**
支持 openGauss 数据库发现
优化 openGauss SQL 方言支持度
优化 openGauss 协议实现性能
优化 openGauss 弹性伸缩

 **ShardingSphere 5.1.1**
与 openGauss 合作，在 TPC-C 测试
达成 **16 节点超过 1000 万 tpmC**

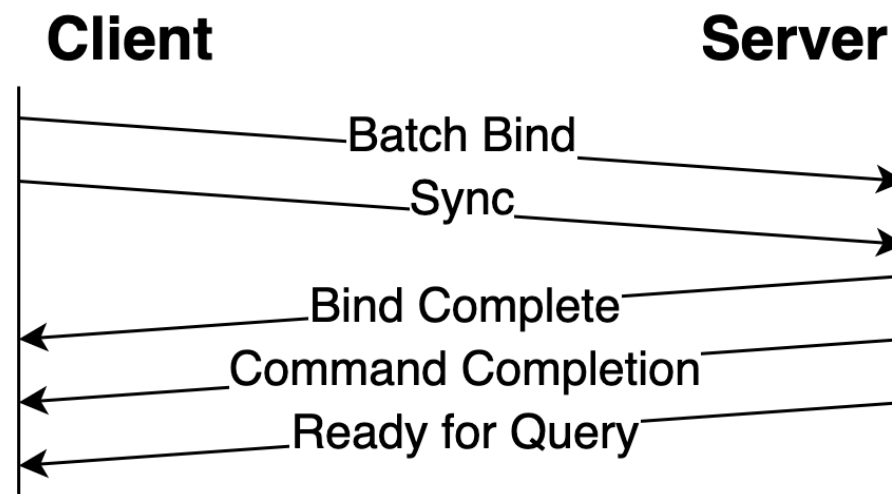
ShardingSphere SQL 解析支持 openGauss 方言



ShardingSphere 支持 openGauss 特有协议

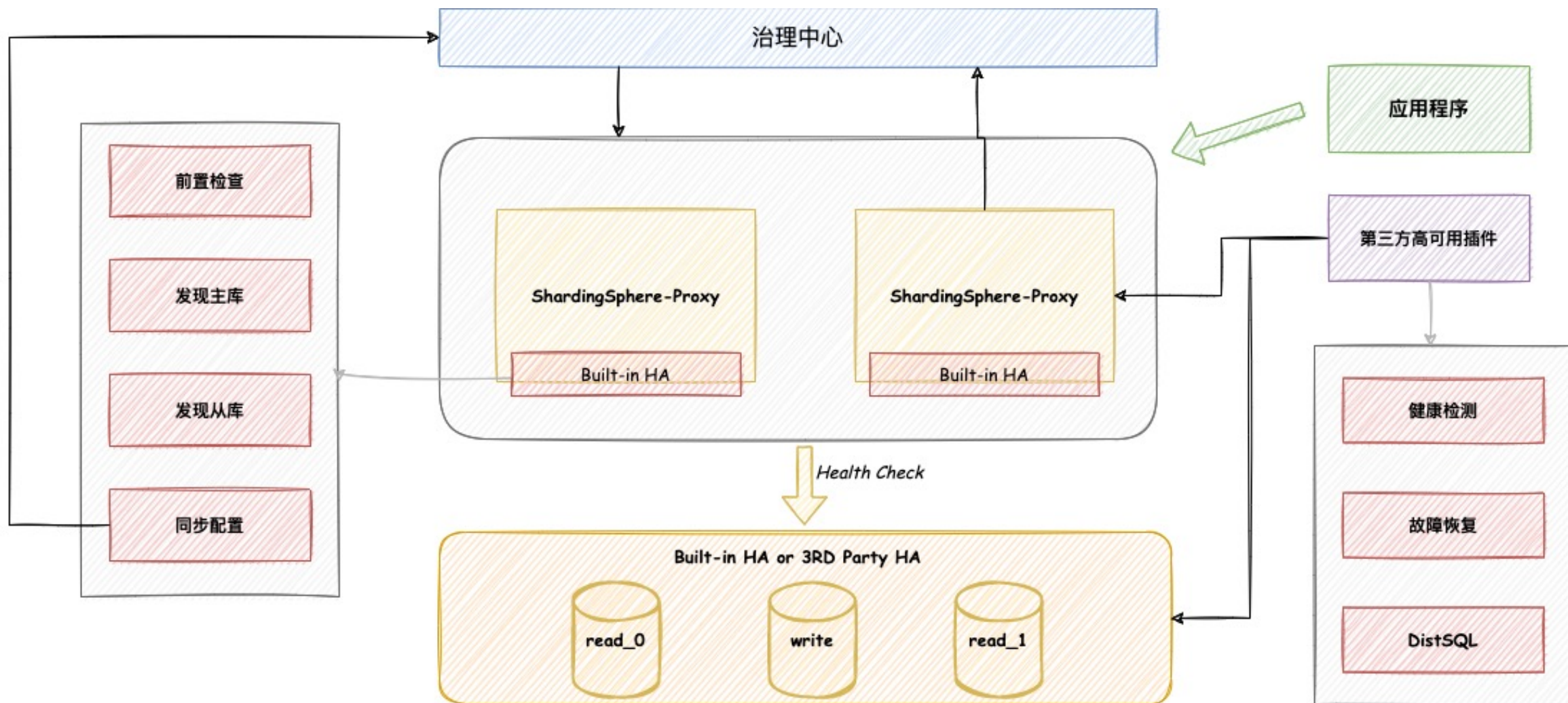


PostgreSQL 批量操作示意



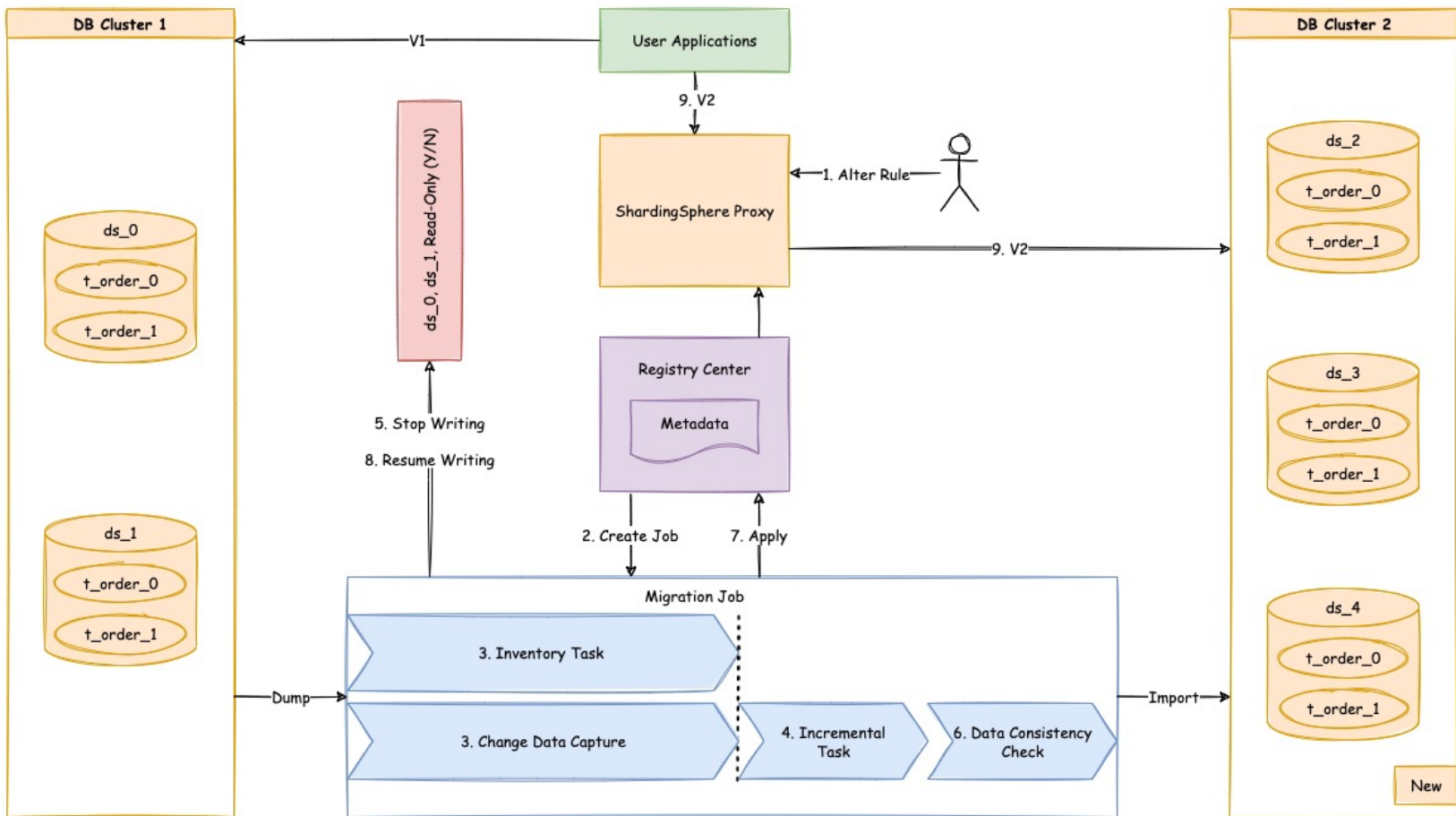
openGauss 批量协议示意

ShardingSphere 数据库发现支持 openGauss



ShardingSphere 弹性伸缩支持 openGauss

功能	MySQL	PostgreSQL	openGauss
全量迁移	支持	支持	支持
增量迁移	支持	支持	支持
自动建表	支持	支持	支持
DATA_MATCH 一致性校验	支持	支持	支持
CRC32_MATCH 一致性校验	支持	不支持	不支持





目录

01

Apache ShardingSphere 介绍

02

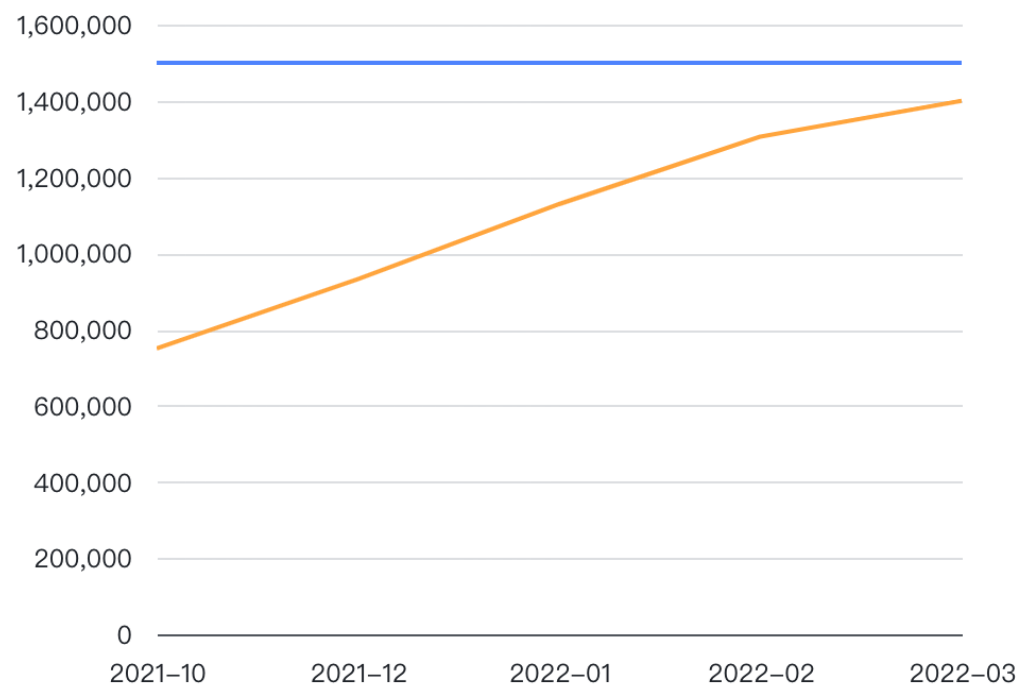
ShardingSphere 与 openGauss 的生态对接

03

ShardingSphere 与 openGauss 的性能磨合

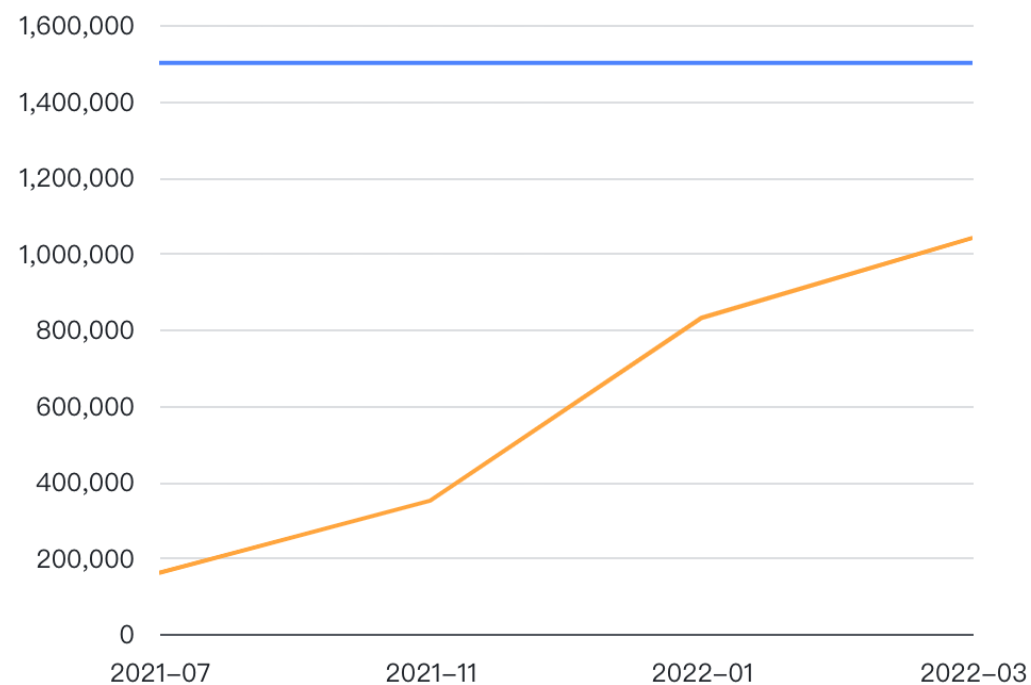
ShardingSphere 与 openGauss 的性能磨合

ShardingSphere-JDBC openGauss 单分片 tpmC



■ 直连 openGauss tpmC
■ ShardingSphere-JDBC openGauss 单分片 tpmC

ShardingSphere-Proxy openGauss 单分片 tpmC

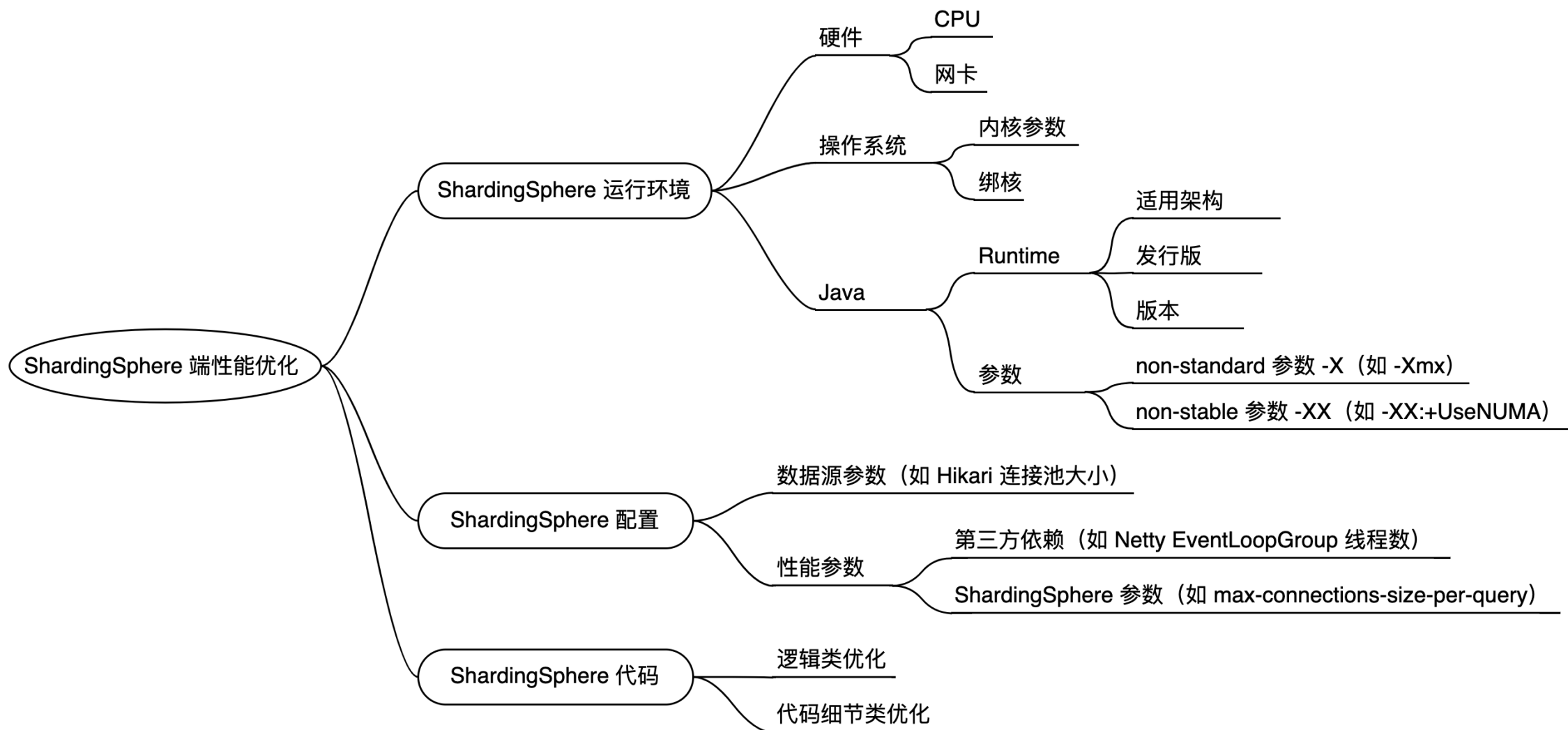


■ 直连 openGauss tpmC
■ ShardingSphere-Proxy openGauss 单分片 tpmC

* 使用 BenchmarkSQL 在 128 Cores 鲲鹏 920 环境测试，数据量 1000 Warehouses，openGauss 基准线 150 万 tpmC

* ShardingSphere 使用数据分片规则与 MOD 算法，分 1 库，不分表

ShardingSphere 端性能优化思路



环境与参数调优带给 ShardingSphere 的性能收益

```
44 VERSION_OPTS=""
45 if [ $int_version = '8' ] ; then
46     VERSION_OPTS="-XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=70"
47 elif [ $int_version = '11' ] ; then
48     VERSION_OPTS="-XX:+SegmentedCodeCache -XX:+AggressiveHeap"
49     if $is_openjdk; then
50         VERSION_OPTS="$VERSION_OPTS -XX:+UnlockExperimentalVMOptions -XX:+UseJVMCICompiler"
51     fi
52 elif [ $int_version = '17' ] ; then
53     VERSION_OPTS="-XX:+SegmentedCodeCache -XX:+AggressiveHeap"
54 else
55     echo "unadapted java version, please notice..."
56 fi
57
58 JAVA_OPTS=" -Djava.awt.headless=true "
59
60 JAVA_MEM_OPTS=" -server -Xmx2g -Xms2g -Xmn1g -Xss1m -XX:AutoBoxCacheMax=4096 -XX:+UseNUMA -XX:+DisableExplicitGC -XX:LargePageSizeInBytes=128m ${VERSION_OPTS}"
```

* 在 128 Cores 鲲鹏 920 环境运行 ShardingSphere , 使用毕昇 JDK 17 相比毕昇 JDK 8 性能提升 5% ~ 10%

借助 async-profiler 发现 ShardingSphere 代码优化点

async-profiler 是针对 JVM 的采样分析工具。

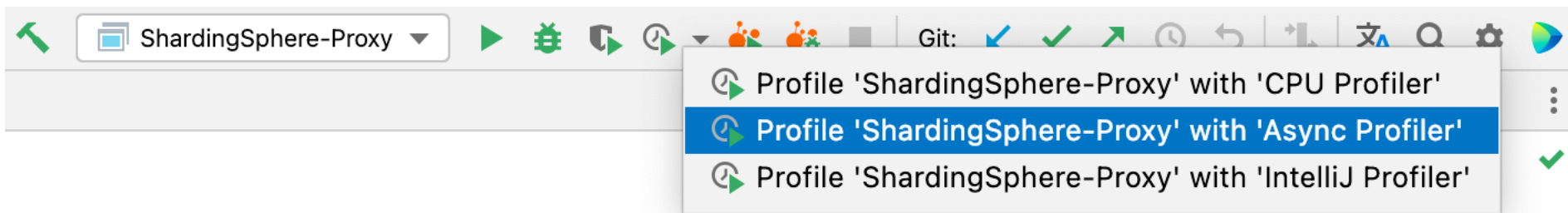
特点：

- ✓ 不受 Safepoint Bias 影响
- ✓ 性能开销低（采样频率可调整），生产可用
- ✓ 使用方便，可基于 Java Agent 启动，或指定 PID 连接已有 JVM
- ✓ 支持多种输出格式（HTML、SVG、JFR 等）及格式转换

支持事件：

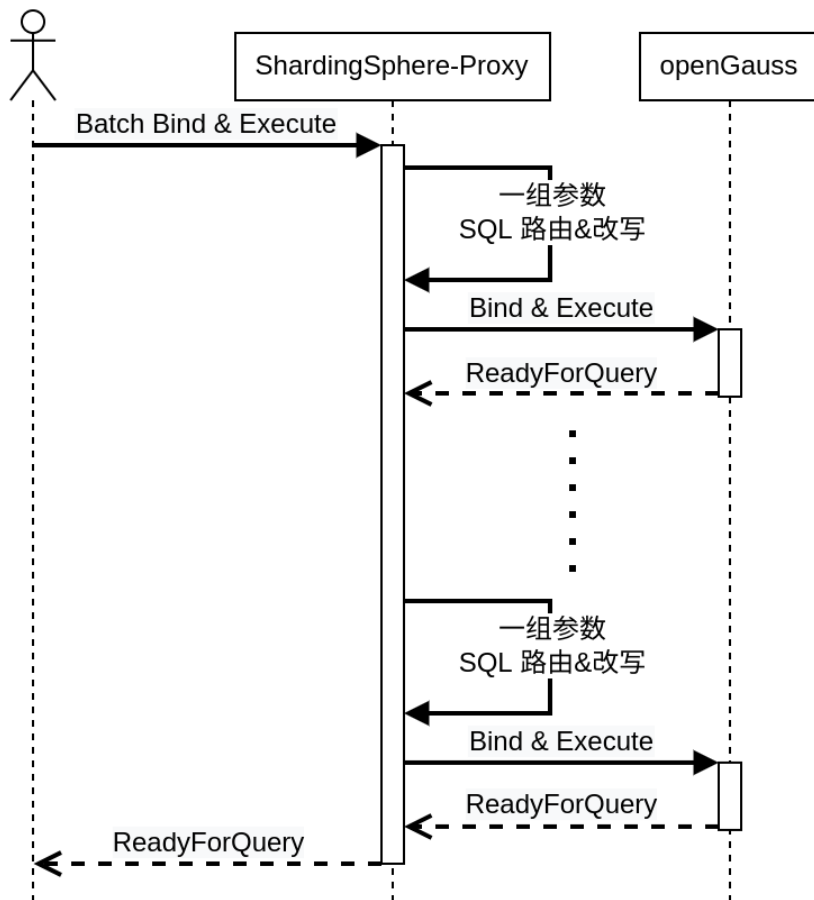
- ✓ CPU 周期
- ✓ 硬件和软件性能指标，如缓存未命中、分支未命中、缺页、上下文切换等
- ✓ Java 堆中的分配
- ✓ 锁，包括 Java 对象监视器和 ReentrantLock
- ✓

```
$ profiler list 163601
Basic events:
  cpu
  alloc
  lock
  wall
  itimer
Java method calls:
  ClassName.methodName
Perf events:
  page-faults
  context-switches
  cycles
  instructions
  cache-references
  cache-misses
  branch-instructions
  branch-misses
  bus-cycles
  L1-dcache-load-misses
  LLC-load-misses
  dTLB-load-misses
  rNNN
  pmu/event-descriptor/
  mem:breakpoint
  trace:tracepoint
  kprobe:func
  uprobe:path
```



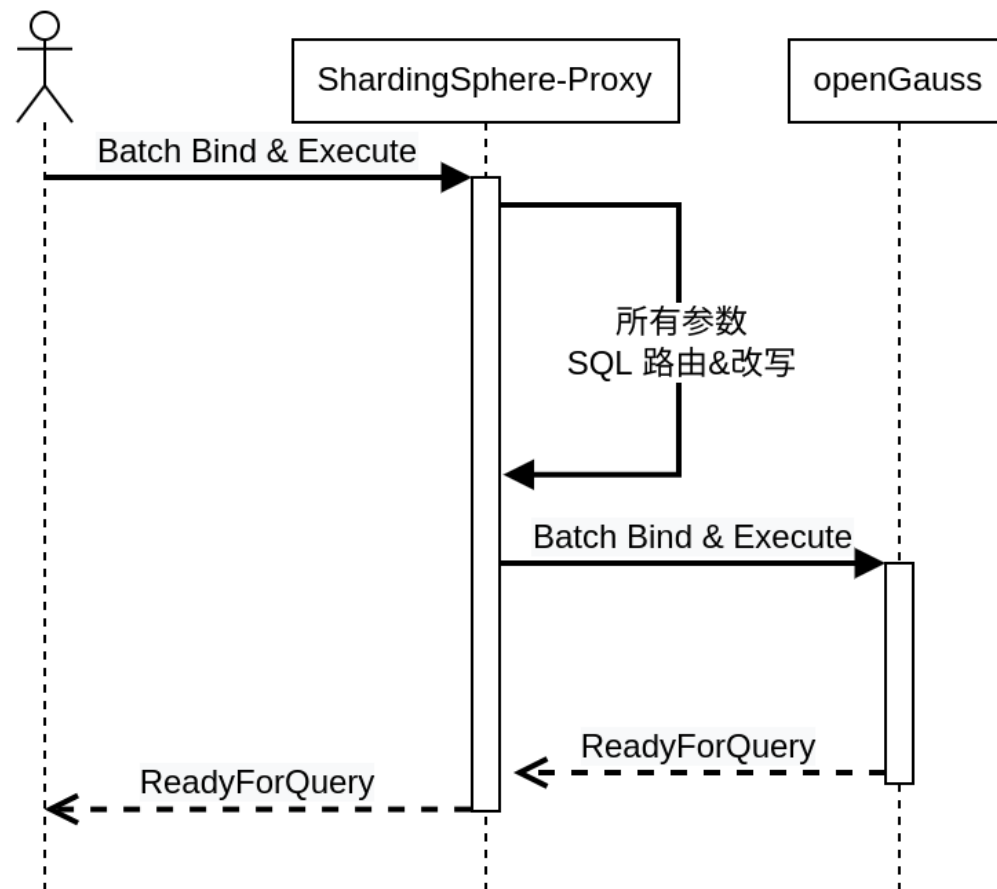
ShardingSphere 代码逻辑优化案例——充分利用 openGauss 批量协议

openGauss 客户端



优化前
ShardingSphere 5.0.0

openGauss 客户端

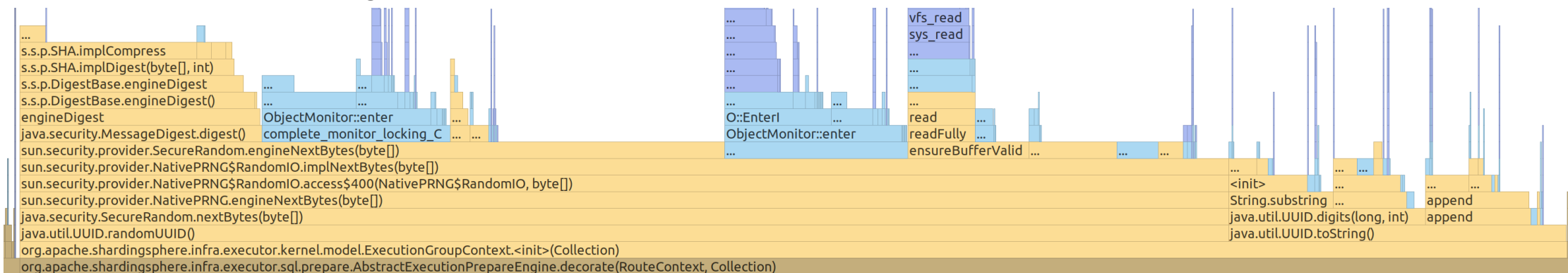


优化后
ShardingSphere 5.1.0+

ShardingSphere 代码细节优化案例——随机 UUID 生成性能优化

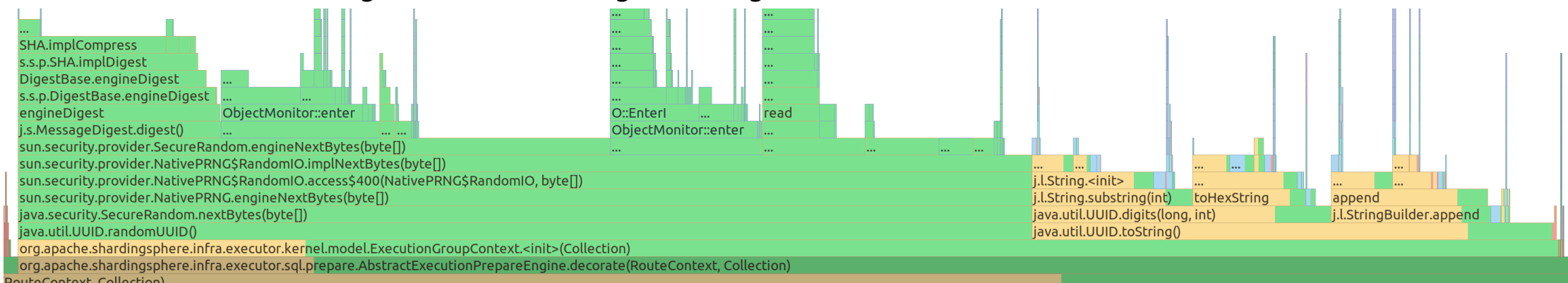
优化前：

UUID.randomUUID().toString()



优化后：

```
Random random = ThreadLocalRandom.current();
new UUID(random.nextLong(), random.nextLong()).toString();
```



欢迎关注 ShardingSphere



技术干货



加入交流群

Apache ShardingSphere Website : <https://shardingsphere.apache.org>

Apache ShardingSphere GitHub : <https://github.com/apache/shardingsphere>

Apache ShardingSphere Slack Channel : <https://apacheshardingsphere.slack.com>

谢谢观看