New in v2.0） 本小节展示了CockroachDB的[TPC-C](http://www.tpc.org/tpcc/)基准性能测试，其基于2个TPC-C数据集合评价tpmC（新订单事务/分钟）指标：

* 1,000 个数据仓库，存储在3个节点上，总共200GB数据
* 10,000 个数据仓库，存储在30个节点上，总共2TB数据

以下将展示CockroachDB如何从适度规模的生产工作负载扩展到更大规模的部署，以及CockroachDB如何在大小超过2TB的TPC-C数据集上实现超过128,000 tpmC（新订单事务/分钟）的高OLTP性能。

小集群基准测试

创建3台GCE机器

* [为CockroachDB集群节点创建实例机器（3台）](https://cloud.google.com/compute/docs/instances/create-start-instance)
* 使用n1-highcpu-16机器类型 我们的TPC-C基准测试过程使用的是n1-highcpu-16类型的机器，目前认为是适用于高流量负载场景下的最佳配置。
* [创建和挂载本地SSD](https://cloud.google.com/compute/docs/disks/local-ssd#create_local_ssd) 为每个虚拟机器挂载一个本地SSD，减少读写延迟以最大化性能。其类似于裸机部署，机器会直接连接到一个物理磁盘。不建议使用网络附加块存储。
* [优化本地SSD的写性能](https://cloud.google.com/compute/docs/disks/performance#optimize_local_ssd) 查看链接网页中**Disable write cache flushing**章节。
* 尽早的应用已配置的防火墙规则，点击**Management** - **disk** - **networking** - **SSH keys**，选择**Networking**标签页、并在**Network tags**配置项填入cockroachdb。
* 获取每台n1-highcpu-16机器的内网IP地址，用于集群节点启动。
* 创建另一台用于运行TPC-C性能测试程序的机器。

**WARNING:** 此配置仅用于性能基准测试。对于生产部署，还有其他重要注意事项，例如确保数据在至少三个可用区域之间保持平衡。详见[生产清单](http://doc.cockroachchina.baidu.com/#deploy/production-checklist/)

启动3节点集群

对创建的3台n1-highcpu-16机器，进行以下操作：

* SSH登陆
* 下载CockraochDB Linux版本[CockroachDB压缩包](https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz)，解压并拷贝到PATH

wget -qO- https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz | tar xvz

sudo cp -i cockroach-v2.0.5.linux-amd64/cockroach /usr/local/bin

* 执行cockroach start命令

cockroach start --insecure --advertise-host=<node1 internal address> --join=<node1 internal address>:26257,<node2 internal address>:26257,<node3 internal address>:26257 --cache=.25 --max-sql-memory=.25 --background

对额外创建的第4台n1-highcpu-16机器，执行cockroach init命令。

cockroach init --insecure --host=localhost

每个节点的CockroachDB将输出详细的帮助信息到[标准输出](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/#standard-output)，例如CockroachDB版本、Web界面的URL地址、客户端SQL URL等。

导入测试数据

CockroachDB提供了linux下预编译的workload二进制，其中包含了多个用于模拟集群客户端流量的负载生成器。 - SSH登陆第四台机器（没有运行CockroachDB进程的节点），下载workload程序，添加执行权限

wget https://edge-binaries.cockroachdb.com/cockroach/workload.LATEST | chmod 755 workload.LATEST

* 重命名且拷贝workload到路径PATH

cp -i workload.LATEST /usr/local/bin/workload

* 指定集群任一节点的链接，生成TPC-C负载

./workload.LATEST fixtures load tpcc --warehouses=10000 "postgres://root@<node1 address>:26257?sslmode=disable"

该命令生成1,000个数据仓库的数据，持续1个小时。

**TIPS:** 更多关于tpcc的选项，可执行workload run tpcc --help查看。查看workload程序其他可选负载生成器，可执行workload run --help。

* 登陆[**Admin界面**](https://www.cockroachlabs.com/docs/stable/admin-ui-access-and-navigate.html)，查看**Jobs**表，了解负载生成器的数据生成进度。

运行测试程序

在第4台机器上，针对前3个节点执行workload程序，持续5分钟。

./workload.LATEST run tpcc --ramp=30s --warehouses=1000 --duration=300s --split --scatter "postgres://root@<node1 address>:26257?sslmode=disable postgres://root@<node2 address>:26257?sslmode=disable postgres://root@<node3 address>:26257?sslmode=disable [...space separated list]"

解析测试结果

workload程序执行结束后，可以看到最后的输出结果：

\_elapsed\_\_\_\_\_\_\_tpmC\_\_\_\_efc\_\_avg*(ms)*\_\_p50*(ms)*\_\_p90*(ms)*\_\_p95*(ms)*\_\_p99*(ms)*\_pMax*(ms)*

298.9s 13154.0 102.3**%** 75.1 71.3 113.2 130.0 184.5 436.2

也可以看到每个查询的一些审计检查和延迟统计信息。此处，部分检查可能由于数据不足而被SKIPPED。更全面的测试需要运行workload程序更长时间（例如两小时）。tpmC即新订单事务/分钟，其编号是首行的编号；efc即“效率”，指出CockroachDB与理论最大tpmC的接近程度。

[TPC-C规范](http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-c_v5.11.0.pdf)要求P90延迟在秒级别，结果中可以看出CockroachDB的P90延迟在数百毫秒范围内、远超过了规范要求。

大集群基准测试

重现CockroachDB的30节点10,000数据仓库的TPC-C测试结果，在方法上类似于上述的小集群测试。除了较大的节点数和数据集外，唯一的区别是用户需要使用CockroachDB的[分区功能](http://doc.cockroachchina.baidu.com/#deploy/define-table-partitions-(enterprise)/)来确保负载生成器执行查询与查询所需要的数据副本在同一个节点上。该分区功能有助于在整个群集中均匀分配工作负载。

在大集群测试中将使用的分区功能，需要用户具有有效的[企业版License](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/)。更多关于申请和设置试用版或完整企业版License的详细信息，可见[企业许可](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/)。

创建30台GCE机器

* [为CockroachDB集群节点创建实例机器（30台）](https://cloud.google.com/compute/docs/instances/create-start-instance)
* 使用n1-highcpu-16机器类型 测试使用的是n1-highcpu-16类型的机器，目前认为是适用于高流量负载场景下的最佳配置。
* [创建和挂载本地SSD](https://cloud.google.com/compute/docs/disks/local-ssd#create_local_ssd) 为每个虚拟机器挂载一个本地SSD，减少读写延迟以最大化性能。类似于裸机部署，机器会直接连接到一个物理磁盘。不建议使用网络附加块存储。
* [优化本地SSD的写性能](https://cloud.google.com/compute/docs/disks/performance#optimize_local_ssd) 查看链接网页中**Disable write cache flushing**章节。
* 尽早的应用已配置的防火墙规则，点击**Management** - **disk** - **networking** - **SSH keys**，选择**Networking**标签页、并在**Network tags**配置项填入cockroachdb。
* 获取每个n1-highcpu-16机器的内网IP地址，用于集群节点启动。
* 创建另一台用于运行TPC-C性能测试程序的机器。

**WARNING:** 此配置仅用于性能基准测试。对于生产部署，还有其他重要注意事项，例如确保数据在至少三个可用区域之间保持平衡。详见[生产清单](http://doc.cockroachchina.baidu.com/#deploy/production-checklist/)

添加企业版License

测试中将使用的分区功能，需要用户具有有效的[企业版License](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/)。更多关于申请和设置试用版或完整企业版License的详细信息，可见[企业许可](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/)。

使用内置的SQL客户端为集群添加企业版License

* SSH登陆第31台机器（未运行CockroachDB进程的节点），启动内置SQL客户端

cockroach sql --insecure

* 添加License

SET CLUSTER SETTING enterprise.license = '<secret>';

* 使用\q或CTRL+D退出交互式shell

启动30节点集群

对创建的30台n1-highcpu-16机器，进行以下操作： - SSH登陆 - 下载CockraochDB Linux版本[CockroachDB压缩包](https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz)，解压并拷贝到PATH

wget -qO- https://binaries.cockroachdb.com/cockroach-v2.0.5.linux-amd64.tgz | tar xvz

sudo cp -i cockroach-v2.0.5.linux-amd64/cockroach /usr/local/bin

如果遇到权限错误，在命令前增加sudo - 执行cockroach start命令

cockroach start --insecure --advertise-host=<node1 internal address> --join=<node1 internal address>:26257,<node2 internal address>:26257,<node3 internal address>:26257, [...] --cache=.25 --max-sql-memory=.25 --locality=rack=1 --background

每个节点启动的时候指定[locality](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/#locality)，代表节点对应一个用户自定义的"机架号码" ，例如--locality=rack=1。此处使用10个机架号码，则每3台机器视作属于同一个机架。

对额外创建的第31台n1-highcpu-16机器，执行cockroach init命令。

cockroach init --insecure --host=localhost

每个节点的CockroachDB将输出详细的帮助信息到[标准输出](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/#standard-output)，例如CockroachDB版本、Web界面的URL地址、客户端SQL URL等。

导入测试数据

CockroachDB提供了linux下预编译的workload二进制，其中包含了多个用于模拟集群客户端流量的负载生成器。 - SSH登陆第31台机器（没有运行CockroachDB进程的节点），下载workload程序，添加执行权限

wget https://edge-binaries.cockroachdb.com/cockroach/workload.LATEST | chmod 755 workload.LATEST

* 重命名且拷贝workload到路径PATH

cp -i workload.LATEST /usr/local/bin/workload

* 指定集群任一节点的链接，生成TPC-C负载

./workload.LATEST fixtures load tpcc --warehouses=100000 "postgres://root@<node1 address>:26257?sslmode=disable"

该命令生成10,000个数据仓库的数据，持续1个小时。

**TIPS:** 更多关于tpcc的选项，可执行workload run tpcc --help查看。查看workload程序其他可选负载生成器，可执行workload run --help。

* 登陆[**Admin界面**](https://www.cockroachlabs.com/docs/stable/admin-ui-access-and-navigate.html)，查看**Jobs**表，了解负载生成器的数据生成进度。

提高快照速率

为加速大量的数据迁移，需要提高快照的速率

* SSH登陆第31台机器，启动内置SQL客户端

cockroach sql --insecure

* 调整快照速率

SET CLUSTER SETTING kv.snapshot\_rebalance.max\_rate='64MiB';

* 使用\q或'CTRL+D'退出交互式shell

数据库分区

使用[数据库分区功能](http://doc.cockroachchina.baidu.com/#deploy/define-table-partitions-(enterprise)/)将TPC-C表和索引划分为10个分区，1个机架对应1个分区内容，使用[zone配置](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/)将这些分区固定到特定机架上。

* SSH登陆第31台机器，启动分区

ulimit -n 10000 && workload.LATEST run tpcc --partitions=10 --split --scatter --warehouses=10000 --duration=1s "postgres://root@<node31 address>:26257?sslmode=disable"

该命令生成持续1秒钟的TPC-C负载，并启动分区进程。 分区把TPC-C数据迁移到正确的位置，将持续12个小时。

* 登陆**Admin界面**，查看**Metrics** - **Queues** - **Replication Queue**图，查看分区进度。调整时间窗口到最后10分钟，可以观察到更细粒度的图。

当冗余队列的等待操作显示为0时，代表CockroachDB完成了数据迁移的工作，用户可以进行接下来的测试。

运行测试程序

在第31台机器上，针对前30个节点执行workload程序，将持续5分钟。

ulimit -n 10000 && ./workload.LATEST run tpcc \

--warehouses=10000 \

--ramp=30s \

--duration=300s \

--split \

--scatter \

"postgres://root@<node1 address>:26257?sslmode=disable postgres://root@<node2 address>:26257?sslmode=disable postgres://root@<node3 address>:26257?sslmode=disable [...space separated list]"

解析测试结果

workload程序执行结束后，可以看出最后的测试结果类似于小集群的测试结果。在增加数据仓库的数量以后，tpmC有了10倍的数值提升。详细结果如下：

\_elapsed\_\_\_\_\_\_\_tpmC\_\_\_\_efc\_\_avg*(ms)*\_\_p50*(ms)*\_\_p90*(ms)*\_\_p95*(ms)*\_\_p99*(ms)*\_pMax*(ms)*

291.6s 131109.8 102.0**%** 115.3 88.1 184.5 268.4 637.5 4295.0

也可以看到每个查询的一些审计检查和延迟统计信息。此处，部分检查可能由于数据不足而被SKIPPED。更全面的测试需要运行workload程序更长时间（例如两小时）。tpmC即新订单事务/分钟，其编号是首行的编号；efc即“效率”，指出CockroachDB与理论最大tpmC的接近程度。

[TPC-C规范](http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-c_v5.11.0.pdf)要求P90延迟在秒级别，结果中可以看出CockroachDB的P90延迟在数百毫秒范围内、远超过了规范要求。