Greenplum 集群之间同步数据方法及性能

Greenplum 集群之间同步数据方法及性能 1
1 概述 2
2 相同集群相同数据库不同 SCHEMA 之间同步数据 2
2.1 查看原始表的大小行数与结构 2
2.2 同步语句 2
2.3 查看 cpu 与内存的使用情况 3
2.3.1 查看 Master CPU 与内存使用情况 3
2.3.2 查看数据节点的 CPU 使用情况 3
2.3.3 查看数据节点的内存与磁盘使用情况 6
2.4 查看耗时与表的大小 7
3 相同集群不同数据库之间同步数据 8
3.1 使用 pg_dum 方式同步数据 8
3.1.1 查看原始表的信息 8
3.1.2 把数据下载到磁盘 9
3.1.2.1 下载数据语句 9
3.1.2.2 查看 Master 节点详细信息 9
3.1.3 把数据导入到数据库中 10
3.1.3.1 导入数据库语句 10
3.1.3.2 查看 Master 节点的详细信息 11
3.1.3.3 查看数据节点的详细信息 11
3.1.4 验证数据的准确性 13
3.1.5 pg_dump 同步数据总结 13
3.2 使用 dblink 同步数据 13
3.2.1 dblink 同步相同集群不同数据库的数据 13
3.2.1.1 同步语句 13
3.2.1.2 查看 Master 节点的详细信息 15
3.2.1.3 查看数据节点的详细信息 15
3.2.2 验证数据的准确性 17
3.2.3 dblink 使用总结 17
3.3 使用 gptransfer 同步数据 17
3.3.1 gptransfer 介绍 17
3.3.2 gptransfer 命令参数介绍 18
3.3.3 查看集群及硬件信息 19
3.3.4 同集群之间同步数据 19
3.3.4.1 查看表的详细信息 19
3.3.4.2 进行表数据同步 19
3.3.4.3 查看硬件详细信息 21
3.3.4.4 查看数据的准确性 23
3.3.4.5 使用 gptransfer 总结 23
3.3.5 不同集群之间同步数据 24

1 概述

Greenplum 数据同步包括相同集群不同 schema 之间的数据同步，包括相同集群不同数据库之间的同步，也包括不同数据库之间的数据，同步包括数据同步，函数同步，用户同步等，接下来不要太多的语言就要干，不要怂、、、、、

2 相同集群相同数据库不同 SCHEMA 之间同步数据

2.1 查看原始表的大小行数与结构

原始数据的大小

```
select pg_size_pretty(pg_relation_size('dim.test1'));
-- 43 GB
```

原始数据的行数

```
select count(*) from dim.test1;
-- 182,683,056
```

原始数据的表结构

```
stagging=# \d    dim.test1;
                        Table "dim.test1"
   Column      |                 Type                 | Modifiers
-----------+------------------------+-----------
  zspid          | character varying(500) |
  encode_v1 | character varying(500) |
  encode_v2 | character varying(500) |
Distributed by: (zspid)
```

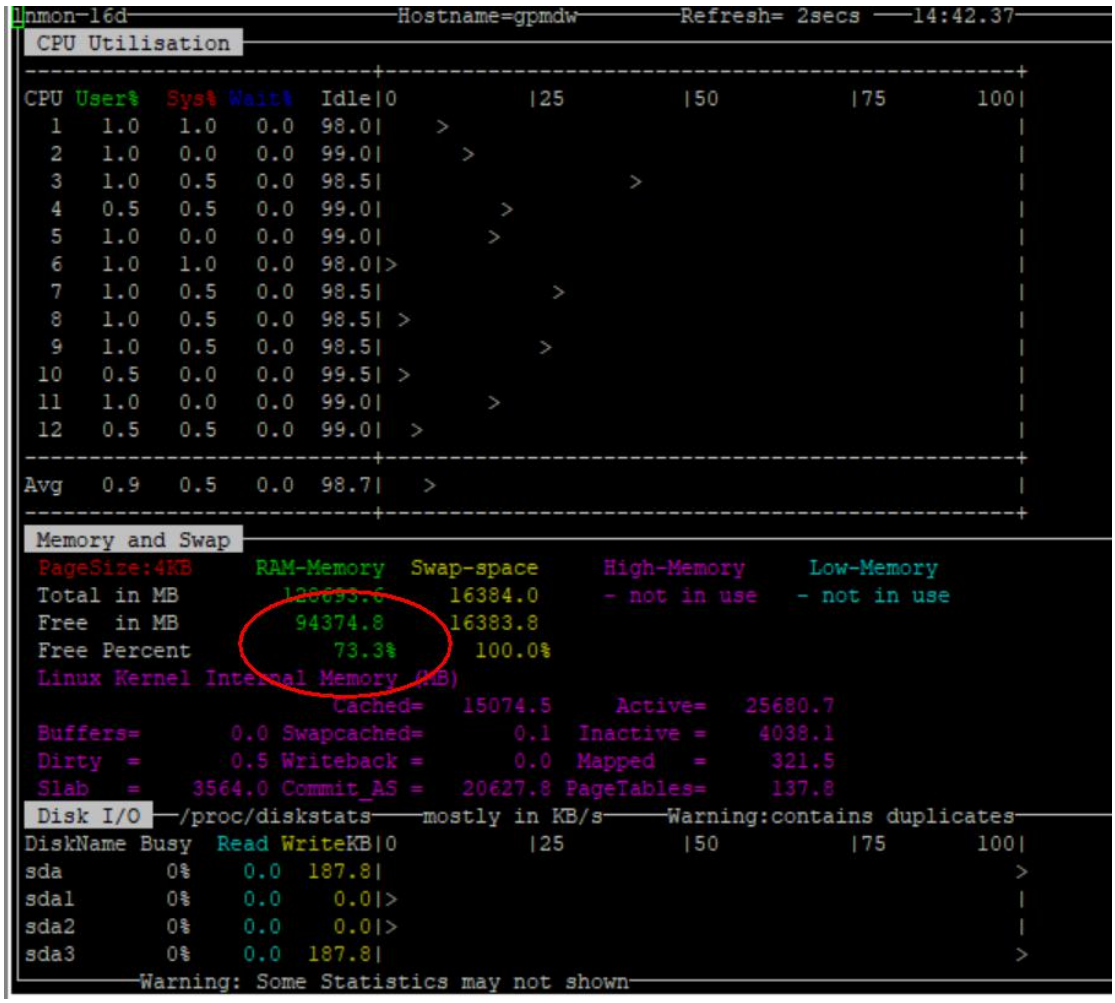
2.2 同步语句

```
create table ods.test1 with (appendonly = true, compresstype = zlib, compresslevel = 5
,orientation=column, checksum = false,blocksize = 2097152)    as
select zspid,encode_v1,encode_v2 from dim.test1
Distributed by (zspid);
```

参数的含义请查看:
https://mp.weixin.qq.com/s?__biz=MzUxNzY0MDg2OQ==&mid=2247484195&idx=1&sn=5785c2e3bba2125b173c9ddf8a1a4f99&chksm=f99444e9cee3cdff

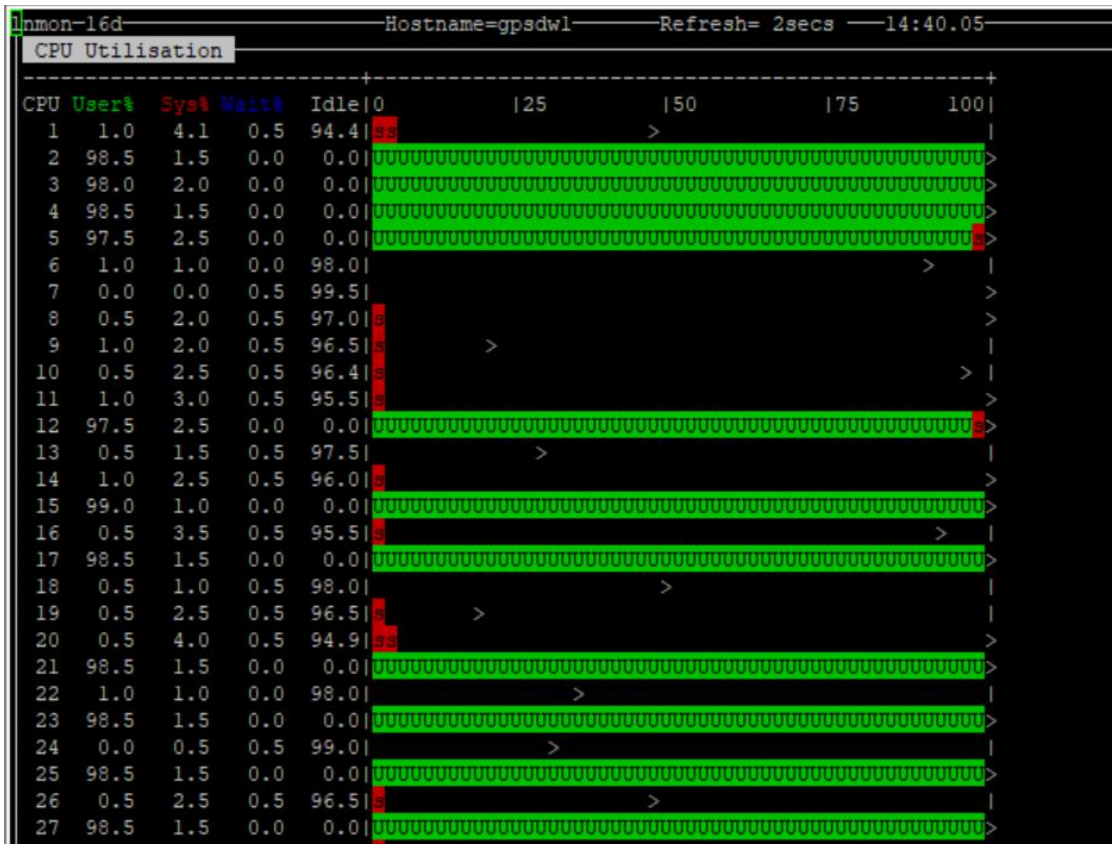
2.3 查看 cpu 与内存的使用情况

2.3.1 查看 Master CPU 与内存使用情况

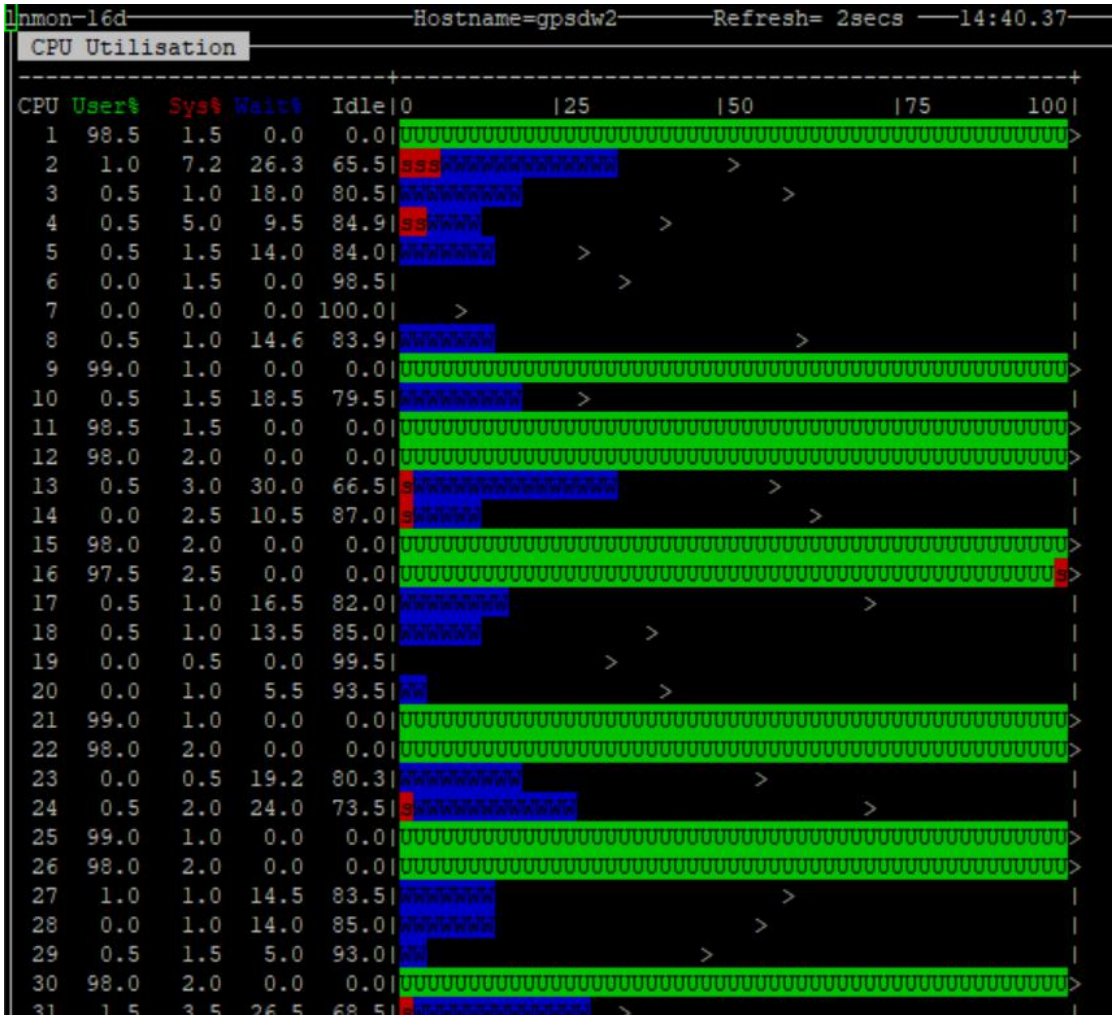


2.3.2 查看数据节点的 CPU 使用情况

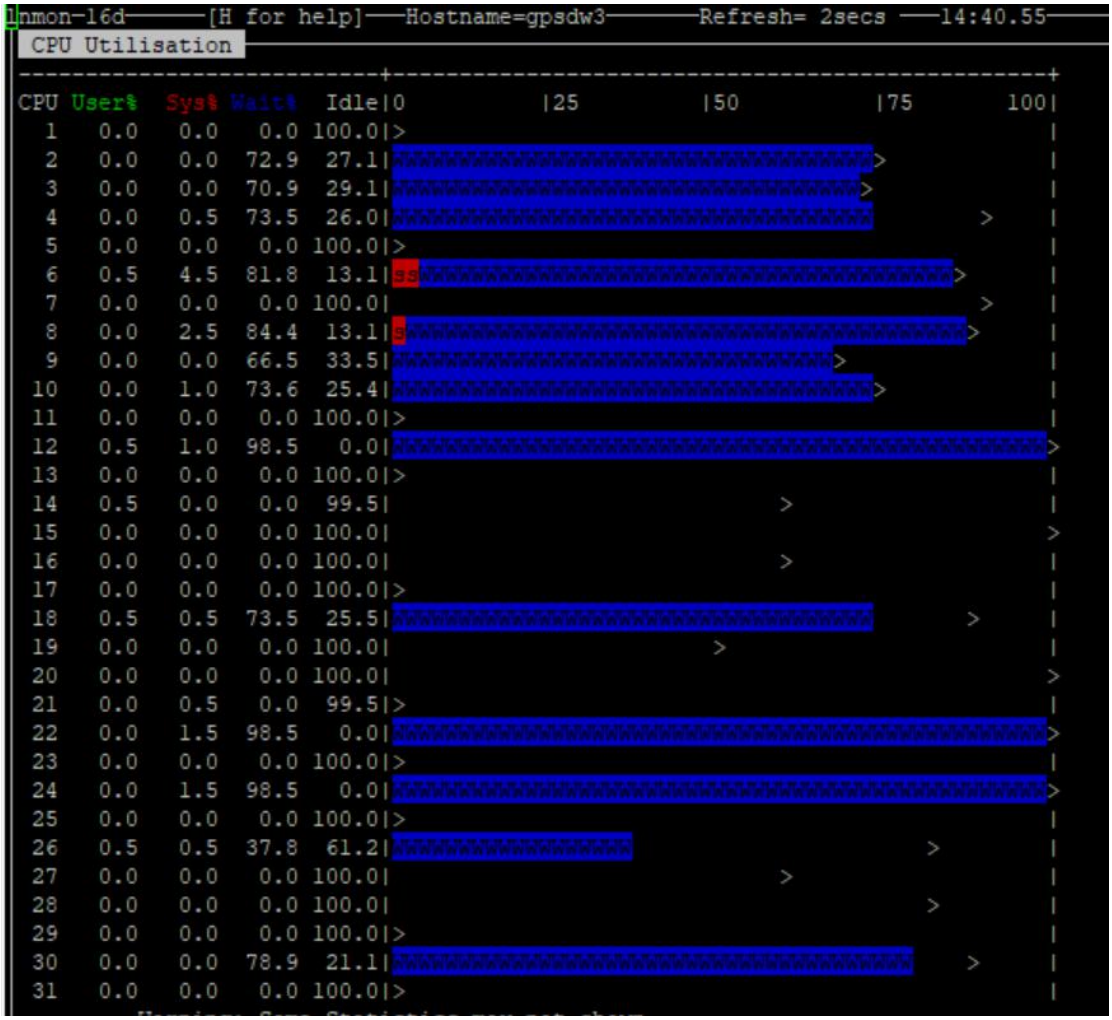
数据节点一使用情况


nmon-16d Hostname=gpsdw1 Refresh= 2secs 14:40.05
CPU Utilisation
CPU User% Sys% Wait% Idle|0 |25 |50 |75 100|
1 1.0 4.1 0.5 94.4|
2 98.5 1.5 0.0 0.0|
3 98.0 2.0 0.0 0.0|
4 98.5 1.5 0.0 0.0|
5 97.5 2.5 0.0 0.0|
6 1.0 1.0 0.0 98.0|
7 0.0 0.0 0.5 99.5|
8 0.5 2.0 0.5 97.0|
9 1.0 2.0 0.5 96.5|
10 0.5 2.5 0.5 96.4|
11 1.0 3.0 0.5 95.5|
12 97.5 2.5 0.0 0.0|
13 0.5 1.5 0.5 97.5|
14 1.0 2.5 0.5 96.0|
15 99.0 1.0 0.0 0.0|
16 0.5 3.5 0.5 95.5|
17 98.5 1.5 0.0 0.0|
18 0.5 1.0 0.5 98.0|
19 0.5 2.5 0.5 96.5|
20 0.5 4.0 0.5 94.9|
21 98.5 1.5 0.0 0.0|
22 1.0 1.0 0.0 98.0|
23 98.5 1.5 0.0 0.0|
24 0.0 0.5 0.5 99.0|
25 98.5 1.5 0.0 0.0|
26 0.5 2.5 0.5 96.5|
27 98.5 1.5 0.0 0.0|

数据节点二使用情况

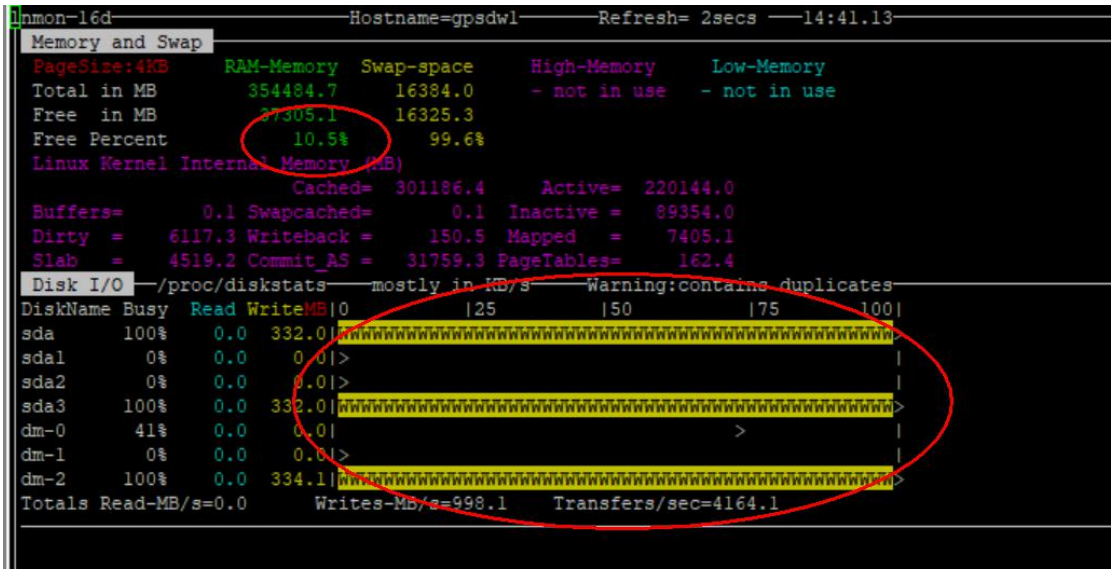

nmon-16d Hostname=gpsdw2 Refresh= 2secs 14:40.37
CPU Utilisation
CPU User% Sys% Wait% Idle|0 |25 |50 |75 100|
1 98.5 1.5 0.0 0.0|
2 1.0 7.2 26.3 65.5|
3 0.5 1.0 18.0 80.5|
4 0.5 5.0 9.5 84.9|
5 0.5 1.5 14.0 84.0|
6 0.0 1.5 0.0 98.5|
7 0.0 0.0 0.0 100.0|
8 0.5 1.0 14.6 83.9|
9 99.0 1.0 0.0 0.0|
10 0.5 1.5 18.5 79.5|
11 98.5 1.5 0.0 0.0|
12 98.0 2.0 0.0 0.0|
13 0.5 3.0 30.0 66.5|
14 0.0 2.5 10.5 87.0|
15 98.0 2.0 0.0 0.0|
16 97.5 2.5 0.0 0.0|
17 0.5 1.0 16.5 82.0|
18 0.5 1.0 13.5 85.0|
19 0.0 0.5 0.0 99.5|
20 0.0 1.0 5.5 93.5|
21 99.0 1.0 0.0 0.0|
22 98.0 2.0 0.0 0.0|
23 0.0 0.5 19.2 80.3|
24 0.5 2.0 24.0 73.5|
25 99.0 1.0 0.0 0.0|
26 98.0 2.0 0.0 0.0|
27 1.0 1.0 14.5 83.5|
28 0.0 1.0 14.0 85.0|
29 0.5 1.5 5.0 93.0|
30 98.0 2.0 0.0 0.0|

数据节点三使用情况



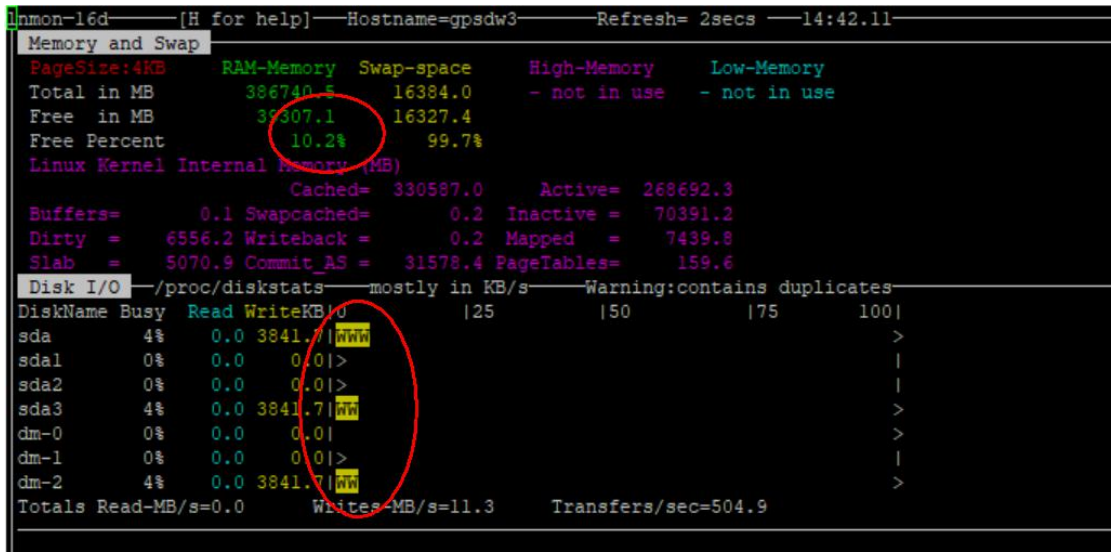
2.3.3 查看数据节点的内存与磁盘使用情况

数据节点一使用情况



数据节点二使用情况

数据节点三使用情况



2.4 查看耗时与表的大小

查看耗时

```
create table ods.test1 with (appendonly = true, compresstype = zlib, compresslevel = 5
,orientation=column, checksum = false,blocksize = 2097152)  as
select zspid,encode_v1,encode_v2 from dim.test1
Distributed by (zspid);
```

时间: 108.063s

受影响的行: 182683056

查看表的大小
select pg_size_pretty(pg_relation_size('ods.test1'));
-- 27 GB

在以上可以看出耗时 108.063s 同步了 27G 的数据，大约 256M / S

3 相同集群不同数据库之间同步数据

3.1 使用 pg_dum 方式同步数据

3.1.1 查看原始表的信息

原始数据的大小
select pg_size_pretty(pg_relation_size('dim.test1'));
-- 43 GB

原始数据的行数
select count(*) from dim.test1;
-- 182,683,056

原始数据的表结构
stagging=# \d dim.test1;

```
                    Table "dim.test1"
   Column      |                Type                | Modifiers
-----------+------------------------+-----------
  zspid          | character varying(500) |
  encode_v1 | character varying(500) |
  encode_v2 | character varying(500) |
Distributed by: (zspid)
```

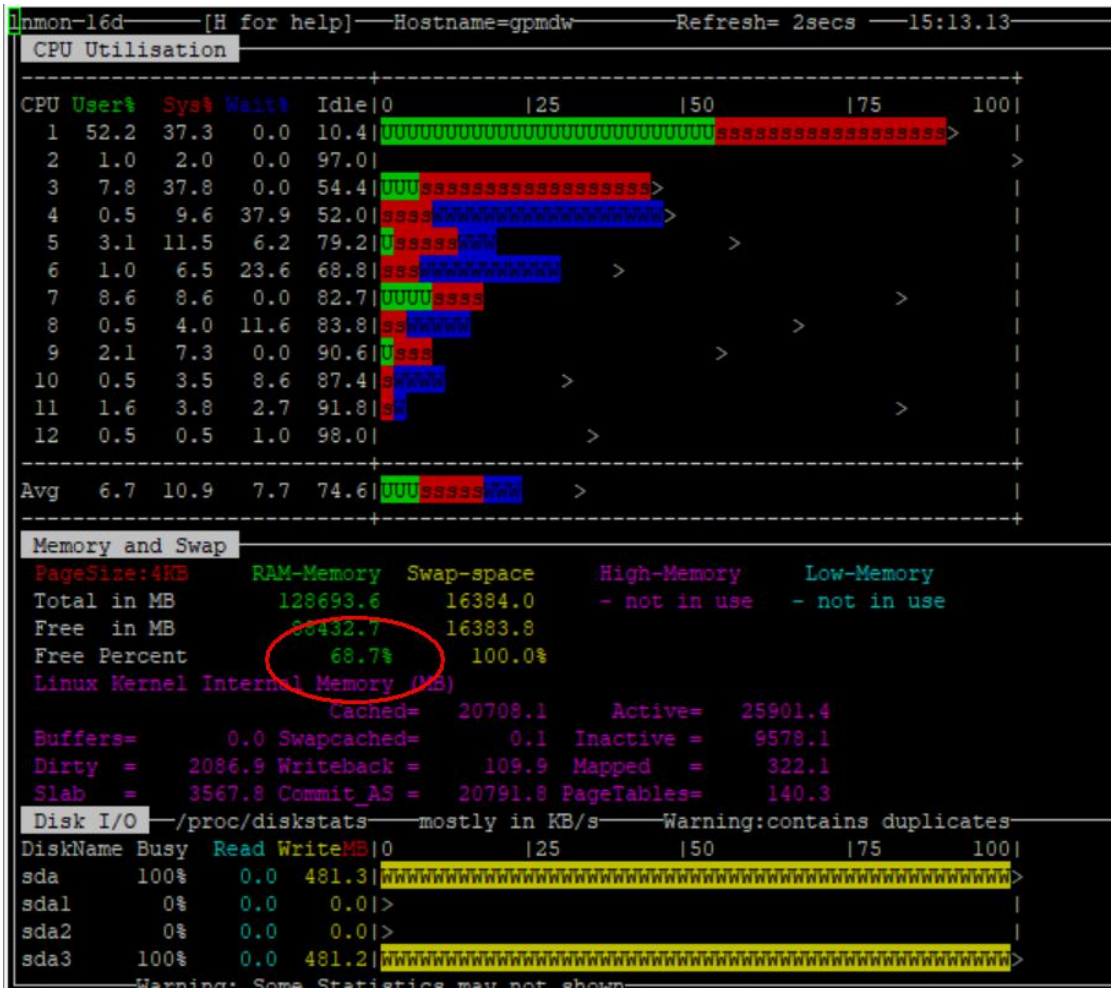
3.1.2 把数据下载到磁盘

3.1.2.1 下载数据语句

$ time pg_dump -h 192.168.***.** -p 5432 -t dim.test1 -U gpadmin stagging -f test1.sql
real 1m58.278s
user 0m10.242s
sys 1m5.665s

下载到磁盘大约用时 1m58.278s

3.1.2.2 查看 Master 节点详细信息



3.1.3 把数据导入到数据库中

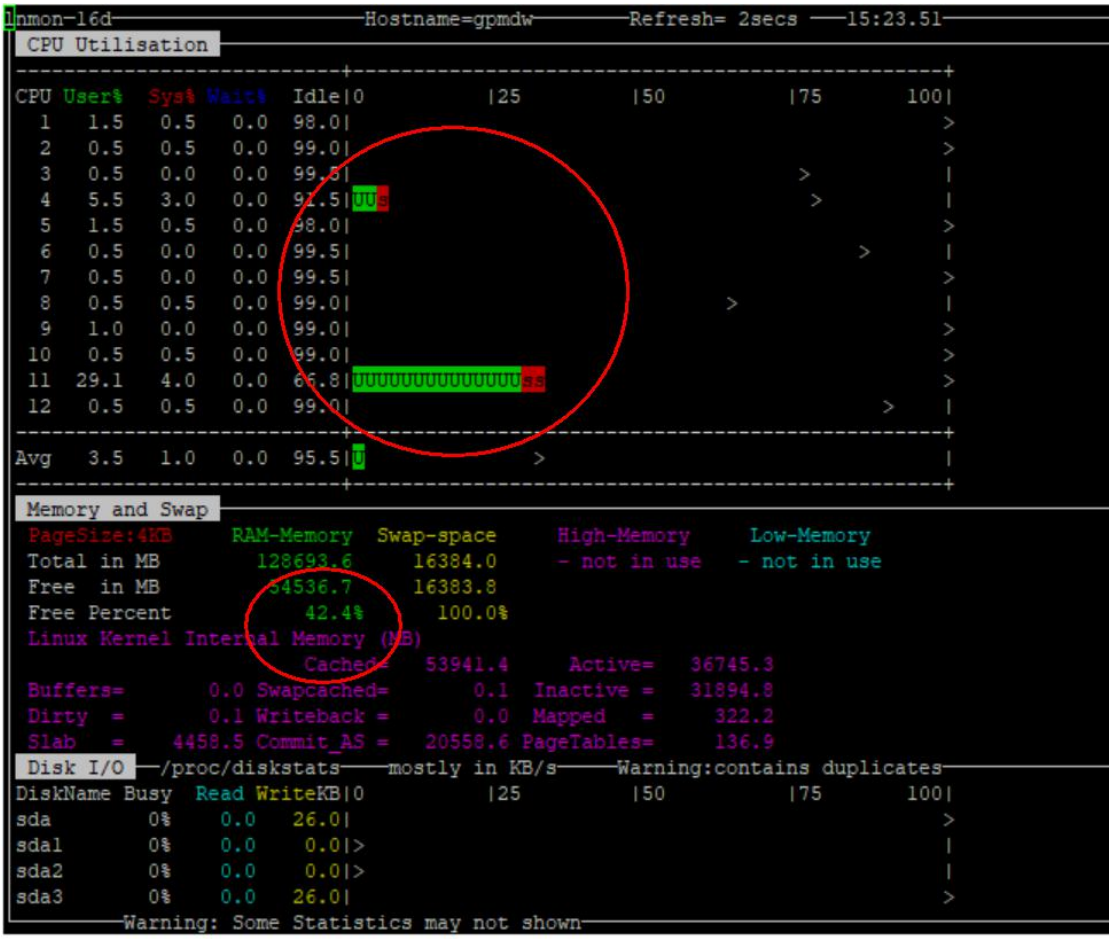
3.1.3.1 导入数据库语句

```
$ time psql -h 192.168.***.**    -p 5432 -Ugpadmin stagging -f test1.sql
SET
SET
SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE

real  20m33.675s
user 0m58.438s
sys   0m33.612s
```

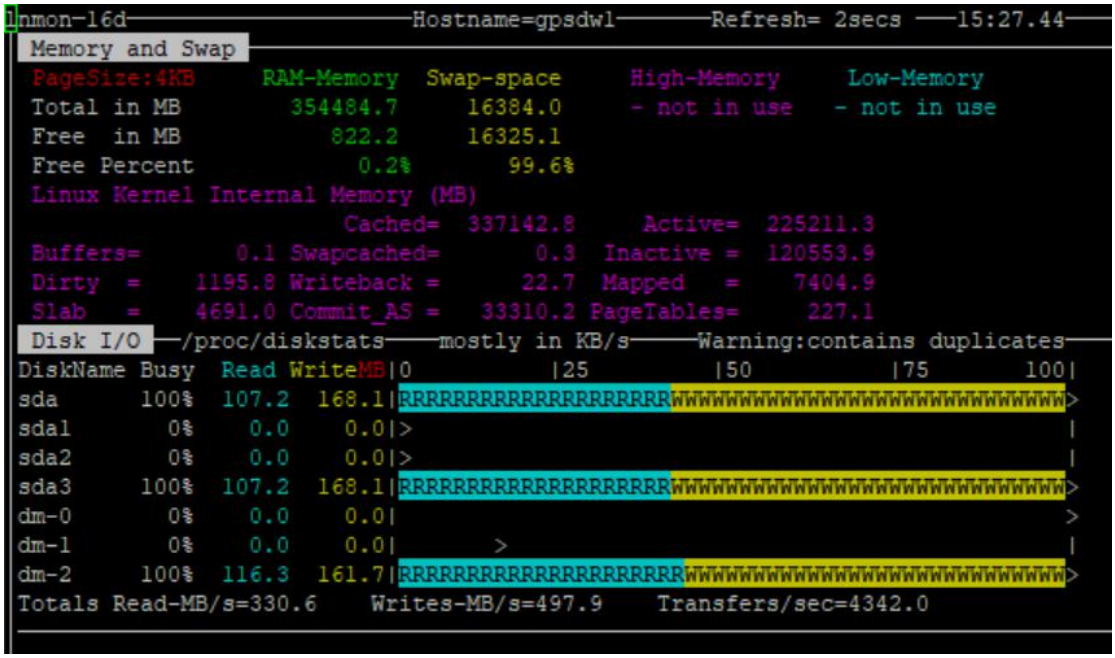
在以上可以看出导入数据耗时 20m33.675s

3.1.3.2 查看 Master 节点的详细信息

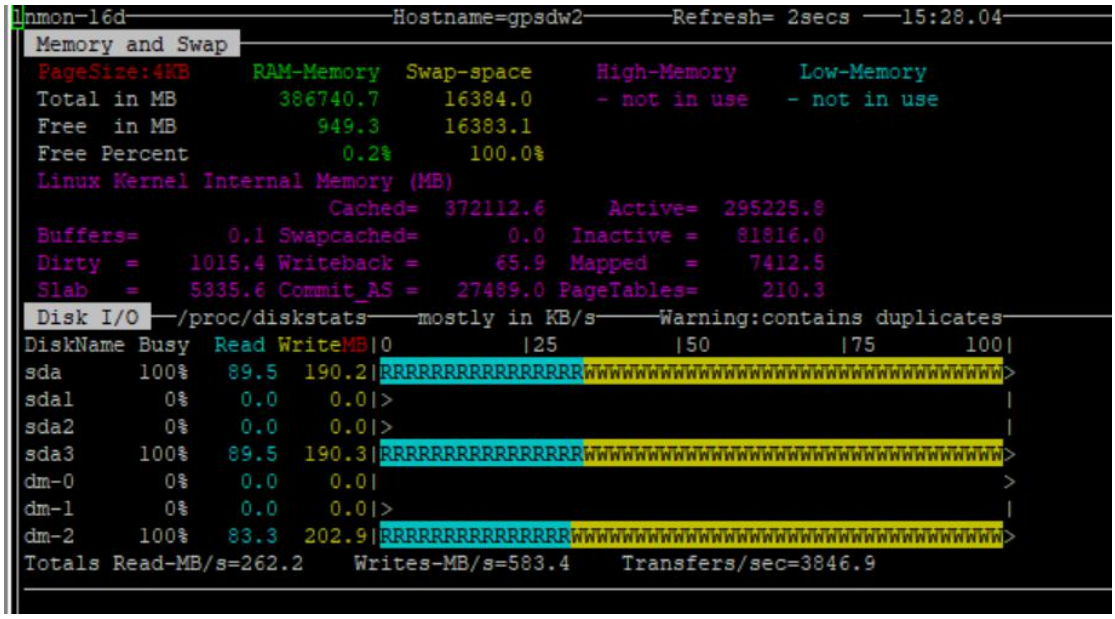


3.1.3.3 查看数据节点的详细信息

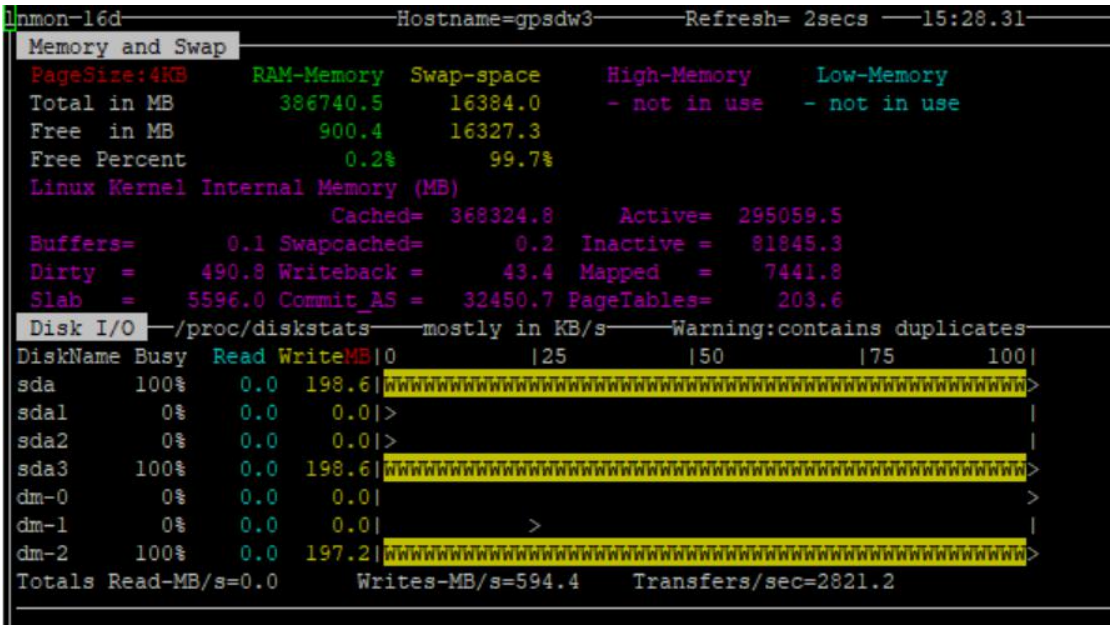
数据节点一的详细信息



数据节点二的详细信息



数据节点三的详细信息



3.1.4 验证数据的准确性

select pg_size_pretty(pg_relation_size('dim.test1'));
-- 43 GB

select count(*) from dim.test1;
-- 182683056

在以上可以看出数据完好

3.1.5 pg_dump 同步数据总结

在以上可以看出 pg_dump 命令在导出时速度还是比较快的，在导入时比较慢，此工具适合于备份用。

3.2 使用 dblink 同步数据

3.2.1 dblink 同步相同集群不同数据库的数据

3.2.1.1 同步语句

drop table if exists dim.test1;
create table dim.test1 with (appendonly = true, compresstype = zlib, compresslevel = 5

,orientation=column, checksum = false,blocksize = 2097152) as
select * from dblink('hostaddr=192.168.***.** port=5432 dbname=chin*** password=*****',
'SELECT zspid,encode_v1,encode_v2 FROM dim.dim_cif_indmap_xiaoxu')
AS dbltab(zspid character varying(500),encode_v1 character varying(500),encode_v2 character varying(500))
DISTRIBUTED BY (zspid);

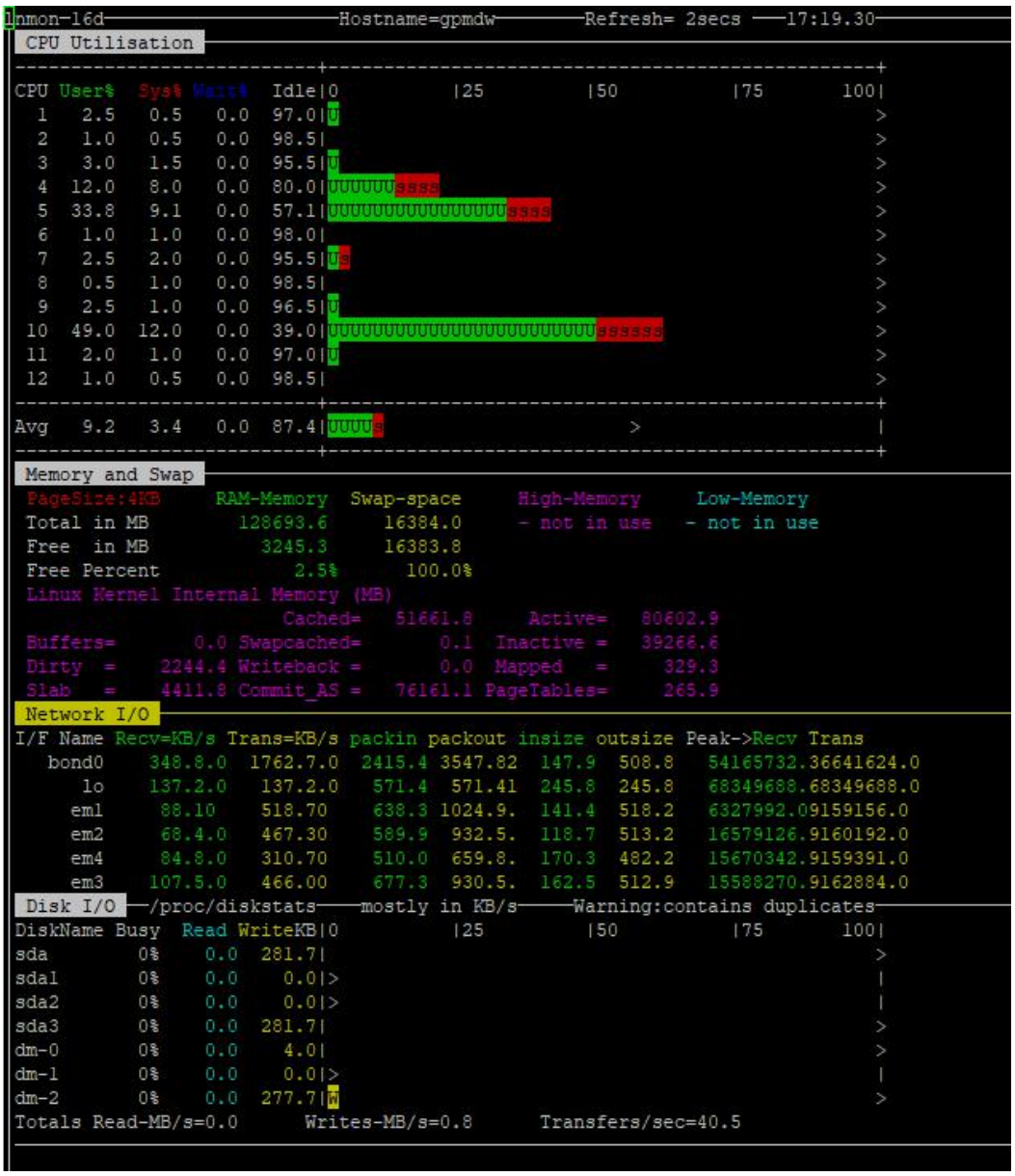
时间: 1631.477s

受影响的行: 0

受影响的行: 182683056

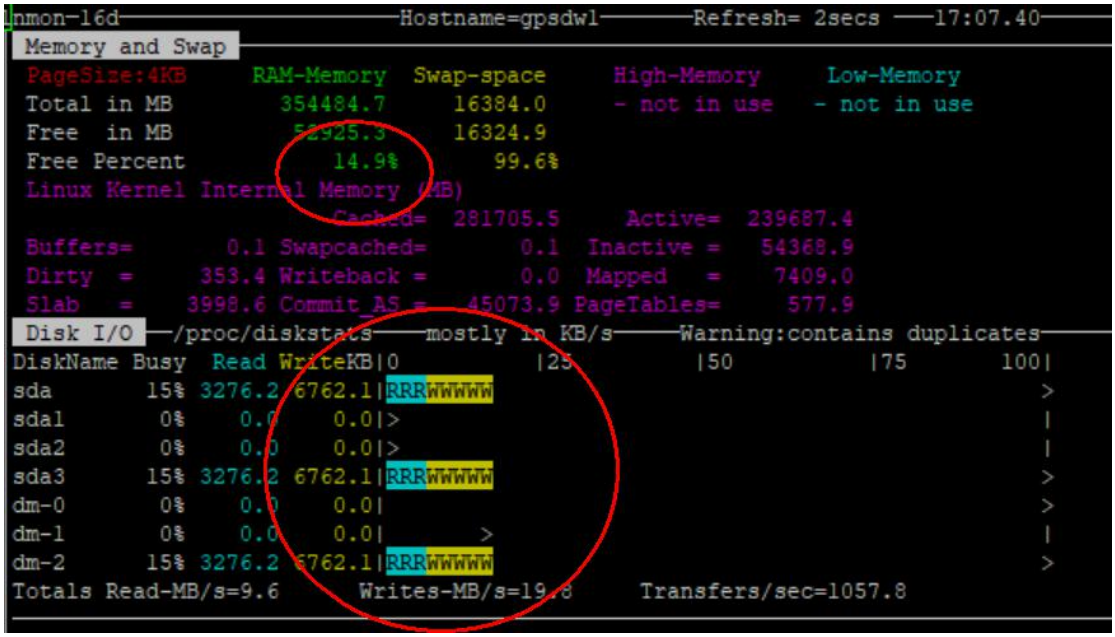
在以上可以看出用时 1631.477s，大约需要 27m，简直太慢,不可行

3.2.1.2 查看 Master 节点的详细信息

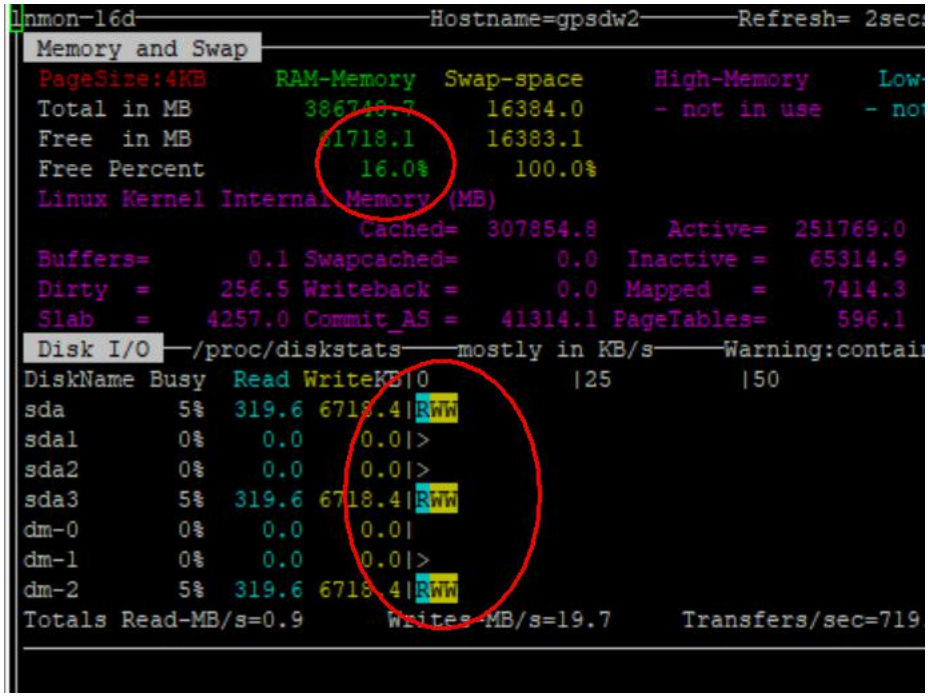


3.2.1.3 查看数据节点的详细信息

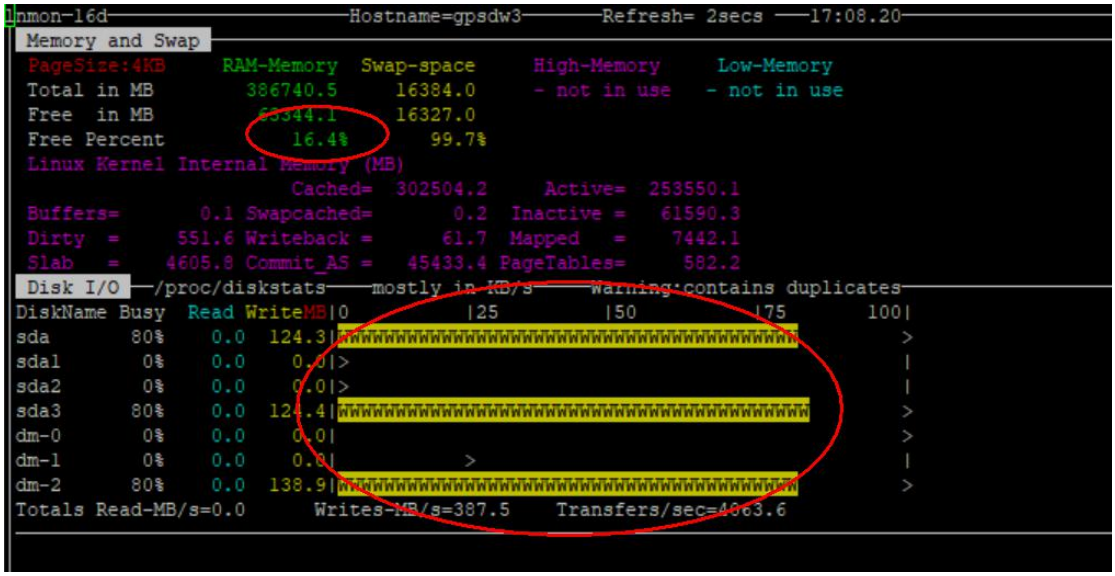
数据节点一的详细信息



数据节点二的详细信息



数据节点三的详细信息



3.2.2 验证数据的准确性

select count(*) from dim.test1;
-- 182683056

select pg_size_pretty(pg_relation_size('dim.test1'));
-- 27 GB

数据的准确性没问题

3.2.3 dblink 使用总结

dblink 因为中间需要映射一张临时表 dbltab，在以上的截图中可以看出， master 主节点的 cpu 比较高，数据节点还可以，可以用于比较小的表同步。

3.3 使用 gptransfer 同步数据

3.3.1 gptransfer 介绍

详细请查看官网介绍:
https://gpdb.docs.pivotal.io/510/utility_guide/admin_utilities/gptransfer.html

或查看中文文档:
https://gp-docs-cn.github.io/docs/admin_guide/managing/gptransfer.htm

1

3.3.2 gptransfer 命令参数介绍

gptransfer 参数详细介绍请查看:
https://blog.csdn.net/xfg0218/article/details/90233815

```
$ gptransfer --help
gptransfer
   { --full |
   { [-d <database1> [ -d <database2> ... ]] |
   [-t <db.schema.table> [ -t <db1.schema1.table1> ... ]] |
   [-f <table-file> [--partition-transfer |
--partition-transfer-non-partition-target]]
   [-T <db.schema.table> [ -T <db1.schema1.table1> ... ]]
   [-F <table-file> ] } }
   [--skip-existing | --truncate | --drop]
   [--analyze] [--validate=<type> ] [-x] [--dry-run]
   [--schema-only ]
   [--no-final-count]

   [--source-host=<source_host> [--source-port=<source_port>]
   [--source-user=<source_user>] ]
   [--base-port=<base_gpfdist_port>]
   [--dest-host=<dest_host> --source-map-file=<host_map_file>
   [--dest-port=<dest_port>] [--dest-user=<dest_user>] ]
   [--dest-database=<dest_database_name>]

   [--batch-size=<batch_size>] [--sub-batch-size=<sub_batch_size>]
   [--timeout <seconds>]
   [--max-line-length=<length>]
   [--work-base-dir=<work_dir>] [-l <log_dir>]
   [--delimiter=<delim> ]
   [--format=[CSV|TEXT] ]
   [--quote=<character> ]

   [-v | --verbose]
   [-q | --quiet]
   [--gpfdist-verbose]
   [--gpfdist-very-verbose]
   [-a]
```

3.3.3 查看集群及硬件信息

硬件信息请查看:
https://blog.csdn.net/xfg0218/article/details/82785196

3.3.4 同集群之间同步数据

3.3.4.1 查看表的详细信息

查看表的大小
```
select pg_size_pretty(pg_relation_size('dim.xiaoxu_test1'));
-- 27 GB
```

查看表的行数
```
select count(*) from dim.xiaoxu_test1;
-- 182683056
```

3.3.4.2 进行表数据同步

查看 source_host_map_file 文件的配置
$ cat source_host_map_file

gpdev152,192.168.***.**
gpdev153,192.168.***.**
gpdev154,192.168.***.**
gpdev155,192.168.***.**

查看 source_tb_list 文件的配置,如果是多个表请一行一行的追加
$ cat source_tb_list
test.dim.test1
test : 数据库的名字
dim:schema 的名字
test1:表的名字

$ gptransfer --source-host=192.168.***.** --source-port=5432 --source-user=gpadmin -f source_tb_list --source-map-file=source_host_map_file -a --dest-host=192.168.***.** --dest-port=5432 --dest-database=stagging --drop

20190515:15:21:04:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Starting gptransfer with args:

--source-host=192.168.***.** --source-port=5432 --source-user=gpadmin -t test.dim.test1 --source-map-file=source_host_map_file -a --dest-host=192.168.***.** --dest-port=5432 --dest-database=stagging --drop
20190515:15:21:04:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Validating options...
20190515:15:21:04:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Retrieving configuration of source Greenplum Database...
20190515:15:21:05:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Retrieving configuration of destination Greenplum Database...
20190515:15:21:06:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Retrieving source tables...
20190515:15:21:06:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Checking for gptransfer schemas...
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Retrieving list of destination tables...
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Reading source host map file...
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Building list of source tables to transfer...
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Number of tables to transfer: 1
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-gptransfer will use "fast" mode for transfer.
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Validating source host map...
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Validating transfer table set...
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Using batch size of 2
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Using sub-batch size of 24
20190515:15:21:07:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Creating work directory '/home/gpadmin/gptransfer_143245'
20190515:15:21:08:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Creating schema dim in database stagging...
20190515:15:21:09:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Starting transfer of test.dim.test1 to stagging.dim.test1...
20190515:15:21:09:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Creating target table stagging.dim.test1...
20190515:15:21:09:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Retrieving schema for table test.dim.test1...
20190515:15:21:12:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Transfering data test.dim.test1 -> stagging.dim.test1...
20190515:15:30:00:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Finished transferring table test.dim.test1, remaining 0 of 1 tables
20190515:15:30:00:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Running final table row count validation on destination tables...
20190515:15:30:09:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Validation of stagging.dim.test1 successful
20190515:15:30:09:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Removing work directories...

20190515:15:30:10:143245 gptransfer:gpdev152:gpadmin-[INFO]:-Finished.

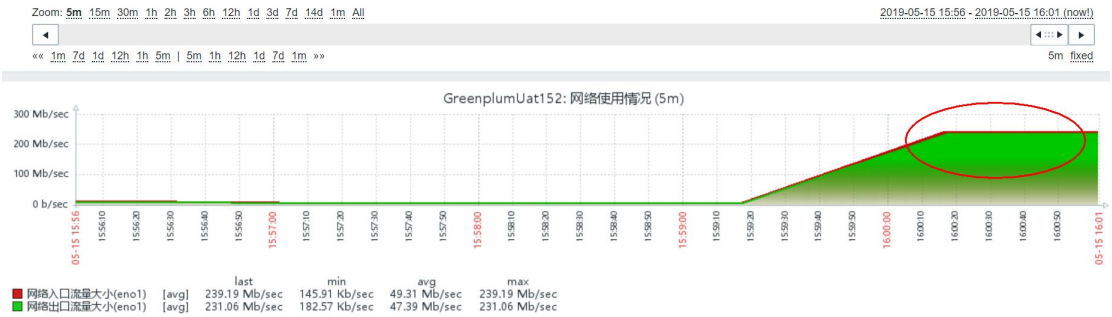
在以上的日志可以看出执行的顺序是
1、先校验参数的正确性
2、检验数据源与目标源的数据库配置
3、校验源数据库中的表
4、检查 gptransfer 是否存在，如果存在回报 gptransfer schema already exists on the source system 警告，在源数据库的 schema 手动删除掉即可
5、检验源数据库集群映射文件
6、把原始文件加载到转换器
7、统计加载的数据源文件
8、在本地磁盘创建转换目录
9、在目标数据库中创建 schema
10、开启数据库员表到目标表的任务
11、创建目标表
12、开始到数据到目标表中
13、统计导到目标表的详细信息

任务是从 20190515:15:21:04 到 20190515:15:30:10 用时大概 9 分钟，大概 27G / 9m ≈ 3G/m 大约 51.2m/s，速度还可以

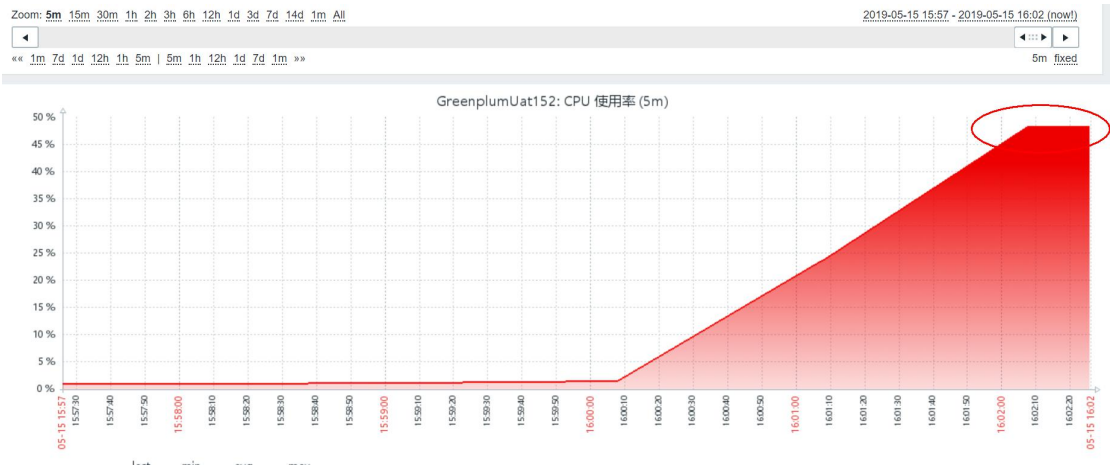
3.3.4.3 查看硬件详细信息

3.3.4.3.1 查看 master 节点的详细信息

查看 master 节点的网卡信息

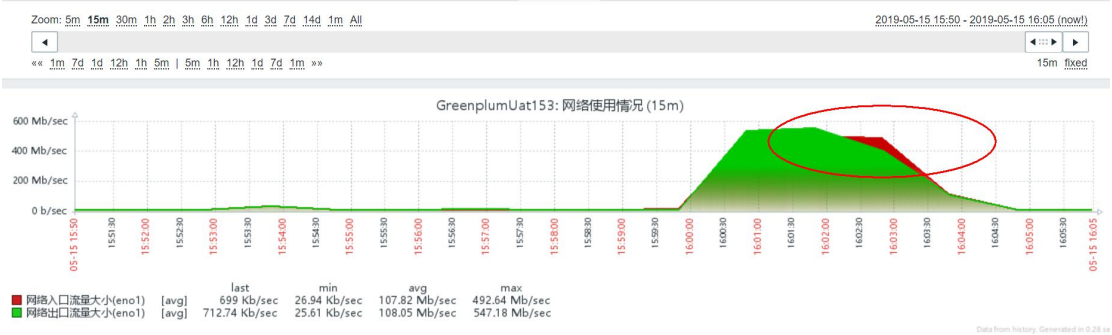


查看 master 节点的 cpu 使用率

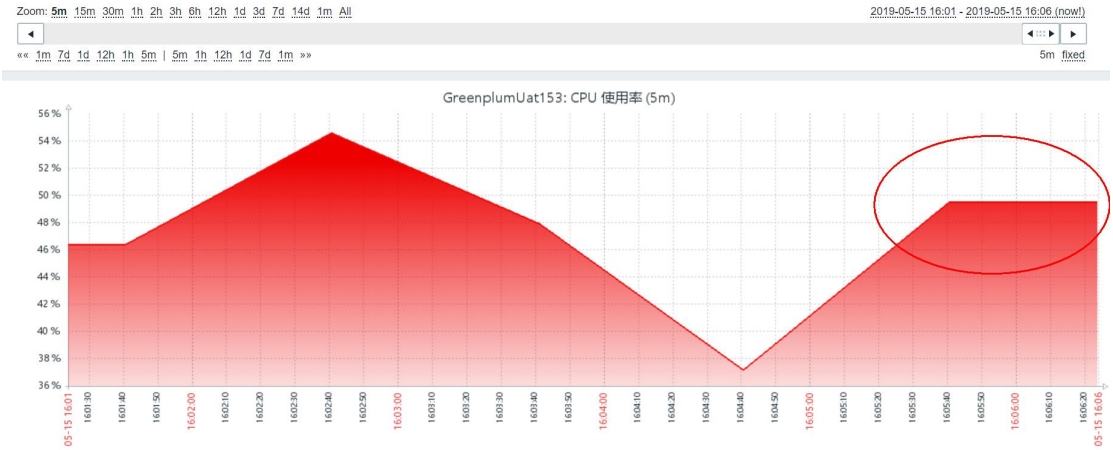


3.3.4.3.2 查看数据节点的详细信息

查看数据节点一的网卡信息



查看数据节点一的 cpu 使用率



3.3.4.4 查看数据的准确性

查看表的大小

```
select pg_size_pretty(pg_relation_size('dim.xiaoxu_test1'));
-- 27 GB
```

查看表的行数

```
select count(*) from dim.xiaoxu_test1;
-- 182683056
```

在以上可以看出数据都准确无误

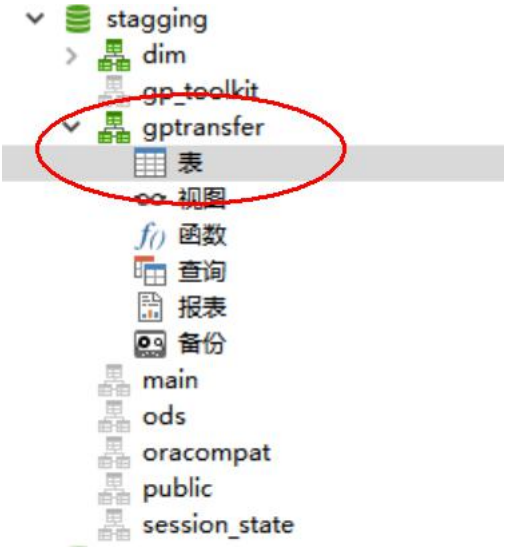
3.3.4.5 使用 gptransfer 总结

在执行命令的服务器上执行 ps -ef|grep gptransfer 会看到以下日志信息，表示在机器上启动 gpfdist 服务，供外表查询数据提供服务

```
$ ps -ef|grep gptransfer
gpadmin   143245   98695   2 15:21 pts/1    00:00:00 python
/usr/local/greenplum-db/./bin/gptransfer --source-host=192.168.***.** --source-port=5432
--source-user=gpadmin -t test.dim.test1 --source-map-file=source_host_map_file -a
--dest-host=192.168.***.** --dest-port=5432 --dest-database=stagging --drop
gpadmin   144007        1 13 15:21 ?        00:00:01 gpfdist -d
/home/gpadmin/gptransfer_143245/test.dim.test1 -p 8000 -P 9000 -m 10485760 -t 300
gpadmin   144039        1 12 15:21 ?        00:00:01 gpfdist -d
/home/gpadmin/gptransfer_143245/test.dim.test1 -p 8000 -P 9000 -m 10485760 -t 300
gpadmin   144048        1 13 15:21 ?        00:00:01 gpfdist -d
/home/gpadmin/gptransfer_143245/test.dim.test1 -p 8000 -P 9000 -m 10485760 -t 300
gpadmin   144077        1 13 15:21 ?        00:00:01 gpfdist -d
/home/gpadmin/gptransfer_143245/test.dim.test1 -p 8000 -P 9000 -m 10485760 -t 300
gpadmin   144079        1 13 15:21 ?        00:00:01 gpfdist -d
/home/gpadmin/gptransfer_143245/test.dim.test1 -p 8000 -P 9000 -m 10485760 -t 300
gpadmin   144097        1 12 15:21 ?        00:00:01 gpfdist -d
/home/gpadmin/gptransfer_143245/test.dim.test1 -p 8000 -P 9000 -m 10485760 -t 300
gpadmin   144118 143610   0 15:21 ?        00:00:00 sh -c
GP_MASTER_HOST='192.168.***.**' && export GP_MASTER_HOST &&
GP_MASTER_PORT='5432' && export GP_MASTER_PORT &&
GP_SEG_PG_CONF='/data/gpsegment/p6/gpseg5/postgresql.conf' && export GP_SEG_PG_CONF
&& GP_SEG_DATADIR='/data/gpsegment/p6/gpseg5' && export GP_SEG_DATADIR &&
GP_DATABASE='test' && export GP_DATABASE && GP_USER='gpadmin' && export GP_USER &&
GP_DATE='20190515' && export GP_DATE && GP_TIME='152112' && export GP_TIME &&
GP_XID='1556272948-0002059298' && export GP_XID && GP_CID='2' && export GP_CID &&
GP_SN='0' && export GP_SN && GP_SEGMENT_ID='5' && export GP_SEGMENT_ID &&
GP_SEG_PORT='40005' && export GP_SEG_PORT && GP_SESSION_ID='30883' && export
```

GP_SESSION_ID && GP_SEGMENT_COUNT='24' && export GP_SEGMENT_COUNT && GP_HADOOP_CONN_JARDIR='lib//hadoop' && export GP_HADOOP_CONN_JARDIR && GP_HADOOP_CONN_VERSION='' && export GP_HADOOP_CONN_VERSION && cat > /home/gpadmin/gptransfer_143245/test.dim.test1/test.dim.test1.pipe.$GP_SEGMENT_ID
****

在同步的 schema 的下会创建一个 gptransfer 的 schema，但不会存放数据



如果在执行的过程中 kill 掉进程请先在目标的 schema 下删除 gptransfer 否则回报以下的错误
****

20190515:10:42:22:106113 gptransfer:gpdev152:gpadmin-[WARNING]:-The gptransfer schema already exists on the source system.
20190515:10:42:22:106113 gptransfer:gpdev152:gpadmin-[WARNING]:-This is likely due to a previous run on gptransfer
20190515:10:42:22:106113 gptransfer:gpdev152:gpadmin-[WARNING]:-being forcefully terminated and not properly cleaned up.
20190515:10:42:22:106113 gptransfer:gpdev152:gpadmin-[WARNING]:-Removing existing gptransfer schema on source system

3.3.5 不同集群之间同步数据

gptransfer --source-host=192.168.***.** --source-port=5432 --source-user=gpadmin -f source_tb_list --source-map-file=source_host_map_file -a --dest-host=192.168.***.** --dest-port=5432 --dest-database=stagging --truncate

只需要修改: --source-host 与--source-map-file 即可
在提示上输入目标 master 服务器的密码即可