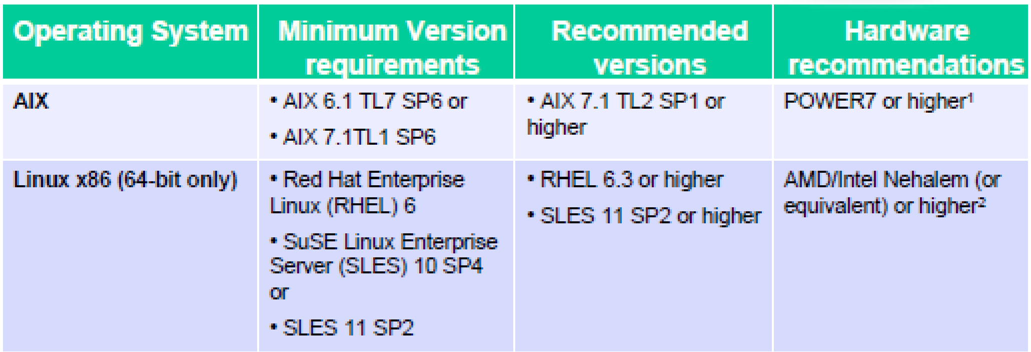
DB2 V10.5 BLU学习系列-01.支持的平台和部署

作者：bigdata lyn@126.com
主页:bigdatalyn.github.io
时间：2016/03

支持的系统平台目前主要是Unix为主



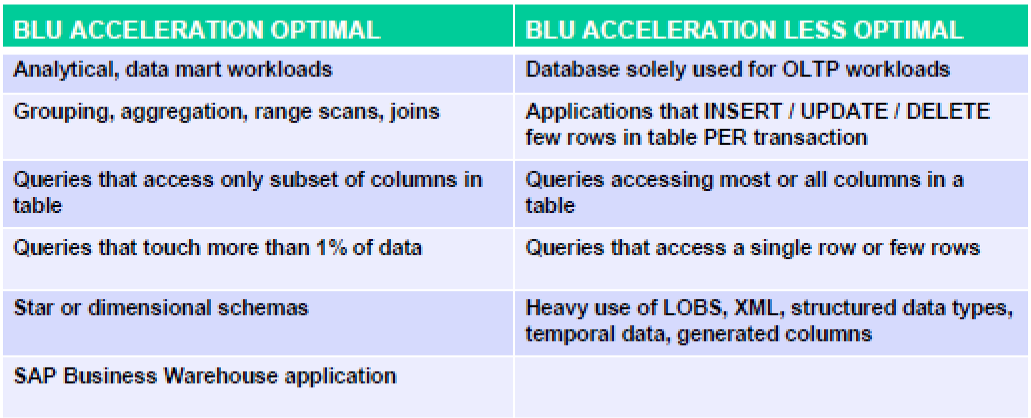
Operating System	Minimum Version requirements	Recommended versions	Hardware recommendations
AIX	• AIX 6.1 TL7 SP6 or • AIX 7.1TL1 SP6	• AIX 7.1 TL2 SP1 or higher	POWER7 or higher[1]
Linux x86 (64-bit only)	• Red Hat Enterprise Linux (RHEL) 6 • SuSE Linux Enterprise Server (SLES) 10 SP4 or • SLES 11 SP2	• RHEL 6.3 or higher • SLES 11 SP2 or higher	AMD/Intel Nehalem (or equivalent) or higher[2]

数据库版本
DB2 Advanced Enterprise Server Edition
DB2 Advanced Workgroup Server Edition
DB2 Developer Edition

另外也支持Cognos BI versions-Cognos 10.2 and 10.2.1

推荐配置
最低： 每3TB数据（没有压缩）配置 8 cores/64GB内存
高配： 每3TB数据（没有压缩）配置 16 cores/128GB内存

什么场景使用BLU



BLU ACCELERATION OPTIMAL	BLU ACCELERATION LESS OPTIMAL
Analytical, data mart workloads	Database solely used for OLTP workloads
Grouping, aggregation, range scans, joins	Applications that INSERT / UPDATE / DELETE few rows in table PER transaction
Queries that access only subset of columns in table	Queries accessing most or all columns in a table
Queries that touch more than 1% of data	Queries that access a single row or few rows
Star or dimensional schemas	Heavy use of LOBS, XML, structured data types, temporal data, generated columns
SAP Business Warehouse application	

怎么使用BLU

有三步：
1.设置实例的注册变量：
db2set DB2_WORKLOAD=ANALYTICS

2.创建新的数据库或者对现有数据库进行自动配置：
db2 autoconfigure apply db only

3.创建列式存储表或者对现有表进行DB2CONVERT转化

示例（创建新的数据库）

1.设置：

```
db2_workload=ANALYTICS
```

2.重启实例：

```
db2stop/db2start
```

3.确认存在的workload对象设置

```
db2 alter workload sysdefaultuserworkload MAXIMUM DEGREE DEFAULT
```

4.自动优化配置参数、

```
db2 AUTOCONFIGURE APPLY DB ONLY
```

5.自动存储

5.1 CREATE STOGROUP ibmcolstogrp ON ‘/data1’ SET AS DEFAULT
5.2 修改DMS表空间为自动存储：

```
ALTER TABLESPACE tbsp MANAGED BY AUTOMATIC STORAGE
ALTER TABLESPACE tbsp REBALANCE
```

5.3 创建新的表空间：（默认32kb 页 4 页扩展块的表空间）

```
CREATE TABLESPACE coltbsp
```

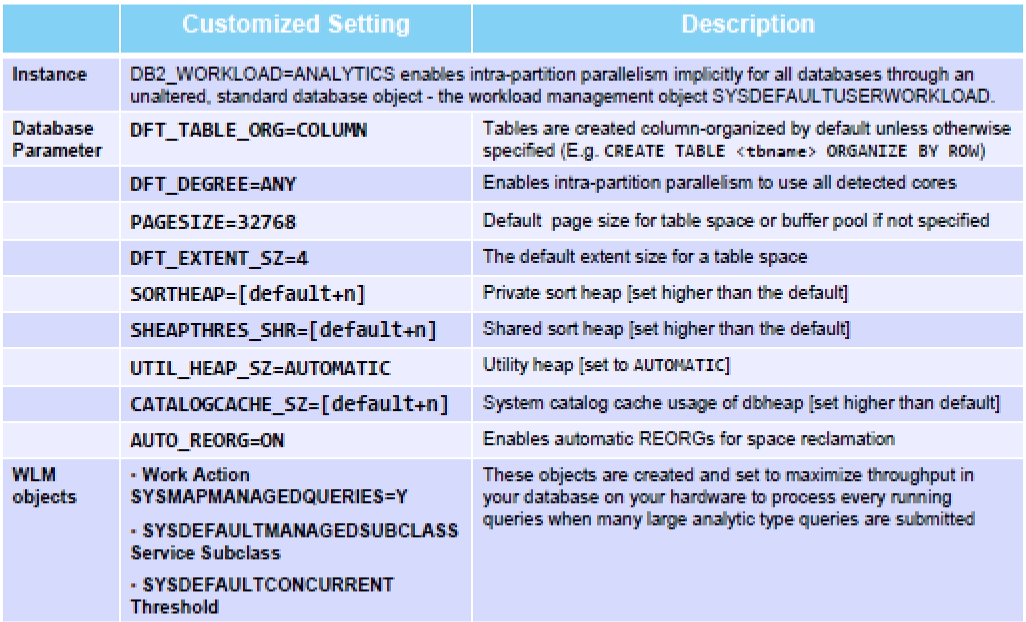
6.转换为列存储表：

6.1 create table之后load数据
6.2 通过db2convert转换

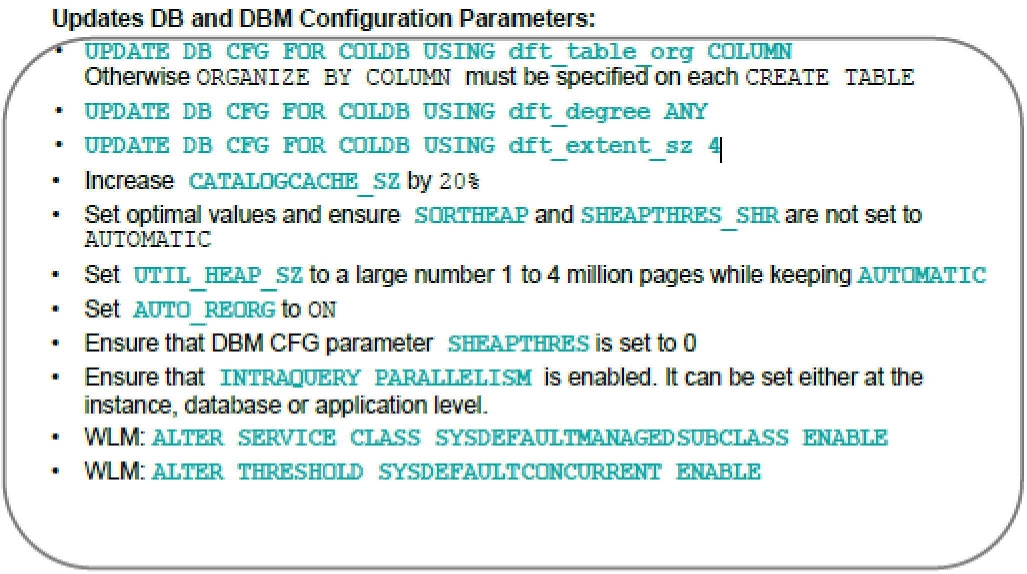
```
DB2CONVERT －d <DBNAME> [-z <SCHEMA> -t <TABLENAME>]
```

数据库要求是UNICODE字符集，校对IDENTITY or IDENTITY_16BIT

db2_workload=ANALYTICS做了哪些内容



	Customized Setting	Description
Instance	DB2_WORKLOAD=ANALYTICS enables intra-partition parallelism implicitly for all databases through an unaltered, standard database object - the workload management object SYSDEFAULTUSERWORKLOAD.	
Database Parameter	DFT_TABLE_ORG=COLUMN	Tables are created column-organized by default unless otherwise specified (E.g. CREATE TABLE <tbname> ORGANIZE BY ROW)
	DFT_DEGREE=ANY	Enables intra-partition parallelism to use all detected cores
	PAGESIZE=32768	Default page size for table space or buffer pool if not specified
	DFT_EXTENT_SZ=4	The default extent size for a table space
	SORTHEAP=[default+n]	Private sort heap [set higher than the default]
	SHEAPTHRES_SHR=[default+n]	Shared sort heap [set higher than the default]
	UTIL_HEAP_SZ=AUTOMATIC	Utility heap [set to AUTOMATIC]
	CATALOGCACHE_SZ=[default+n]	System catalog cache usage of dbheap [set higher than default]
	AUTO_REORG=ON	Enables automatic REORGs for space reclamation
WLM objects	· Work Action SYSMAPMANAGEDQUERIES=Y · SYSDEFAULTMANAGEDSUBCLASS Service Subclass · SYSDEFAULTCONCURRENT Threshold	These objects are created and set to maximize throughput in your database on your hardware to process every running queries when many large analytic type queries are submitted



Updates DB and DBM Configuration Parameters:

- UPDATE DB CFG FOR COLDB USING dft_table_org COLUMN
  Otherwise ORGANIZE BY COLUMN must be specified on each CREATE TABLE
- UPDATE DB CFG FOR COLDB USING dft_degree ANY
- UPDATE DB CFG FOR COLDB USING dft_extent_sz 4
- Increase CATALOGCACHE_SZ by 20%
- Set optimal values and ensure SORTHEAP and SHEAPTHRES_SHR are not set to AUTOMATIC
- Set UTIL_HEAP_SZ to a large number 1 to 4 million pages while keeping AUTOMATIC
- Set AUTO_REORG to ON
- Ensure that DBM CFG parameter SHEAPTHRES is set to 0
- Ensure that INTRAQUERY PARALLELISM is enabled. It can be set either at the instance, database or application level.
- WLM: ALTER SERVICE CLASS SYSDEFAULTMANAGEDSUBCLASS ENABLE
- WLM: ALTER THRESHOLD SYSDEFAULTCONCURRENT ENABLE

练习：

```
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> db2set -all
[e] DB2CODEPAGE=1208
[e] DB2PATH=/opt/ibm/db2/V10.5
[i] DB2RSHCMD=/usr/bin/ssh
[i] DB2COMM=TCPIP
[i] DB2AUTOSTART=YES
[g] DB2SYSTEM=db2v105
[g] DB2INSTDEF=db2inst1
[g] DB2ADMINSERVER=dasusr1
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> db2set DB2_WORKLOAD=ANALYTICS
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> db2set -all
[e] DB2CODEPAGE=1208
[e] DB2PATH=/opt/ibm/db2/V10.5
[i] DB2_WORKLOAD=ANALYTICS
[i] DB2RSHCMD=/usr/bin/ssh
[i] DB2COMM=TCPIP
[i] DB2AUTOSTART=YES
[g] DB2SYSTEM=db2v105
[g] DB2INSTDEF=db2inst1
[g] DB2ADMINSERVER=dasusr1
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> db2start
03/12/2016 11:26:43     0   0   SQL1026N  The database manager is already active.
SQL1026N  The database manager is already active.
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> db2_ps
strings: /opt/ibm/cognos/c10_64/bin64/libz.so.1: no version information available (required by
/usr/lib64/libbfd-2.23.1.so)
Node 0
     UID        PID       PPID    C     STIME     TTY      TIME CMD
db2inst1       4247       4245    0     11:14        ? 00:00:01 db2sysc 0
    root       4248       4247    0     11:14        ? 00:00:00 db2ckpwd 0
    root       4249       4247    0     11:14        ? 00:00:00 db2ckpwd 0
    root       4250       4247    0     11:14        ? 00:00:00 db2ckpwd 0
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals>
```

小Tip：

```
show_analytics_params.sh
#!/bin/bash
db2 get db cfg for coltest | grep -iE '(dft_table_org)|(dft_degree)|(dft_extent_sz)|(catalogcache_sz)|
(sortheap)|(sheapthres_shr)|(util_heap_sz)|(auto_reorg)'
```

创建瞬间前后比较：

```
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> ./show_analytics_params.sh
 Degree of parallelism                      (DFT_DEGREE) = 1
 Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 5000
 Sort list heap (4KB)                         (SORTHEAP) = 256
 Catalog cache size (4KB)              (CATALOGCACHE_SZ) = (MAXAPPLS*5)
 Utilities heap size (4KB)                (UTIL_HEAP_SZ) = 5000
 Default tablespace extentsize (pages)   (DFT_EXTENT_SZ) = 32
     Automatic reorganization               (AUTO_REORG) = OFF
 Default table organization              (DFT_TABLE_ORG) = ROW
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> ./show_analytics_params.sh
 Degree of parallelism                      (DFT_DEGREE) = ANY
 Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 93376
 Sort list heap (4KB)                         (SORTHEAP) = 32768
 Catalog cache size (4KB)              (CATALOGCACHE_SZ) = 360
 Utilities heap size (4KB)                (UTIL_HEAP_SZ) = AUTOMATIC(50709)
 Default tablespace extentsize (pages)   (DFT_EXTENT_SZ) = 32
     Automatic reorganization               (AUTO_REORG) = ON
 Default table organization              (DFT_TABLE_ORG) = COLUMN
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals>
```

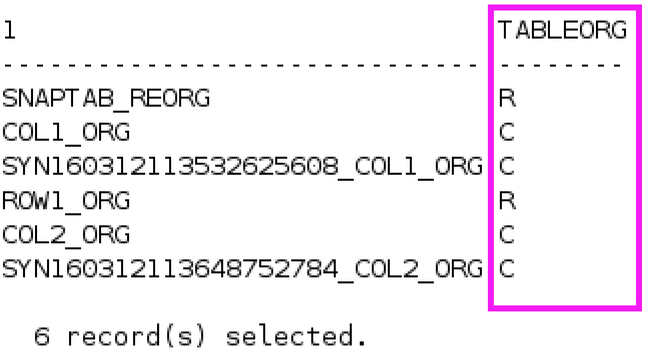
The *dft_degree* (default degree of parallelism) database configuration parameter is set to ANY.
The values of the *sortheap* (sort heap) and *sheapthres_shr* (sort heap threshold for shared sorts) database configuration parameters are calculated specifically for an analytics workload. These settings take into account the additional memory requirements for processing column-organized data.
The *catalogcache_sz* (catalog cache) database configuration parameter is set to a value that is higher than the value for a non-analytics workload.
The *util_heap_sz* (utility heap size) database configuration parameter is set to a value that takes into account the additional memory that is required to load the data into column-organized tables.
The *dft_extent_sz* (default extent size) database configuration parameter is set to 4.
The *auto_reorg* (automatic reorganization) database configuration parameter is set to ON.
The *dft_table_org* (default table organization for user tables) database configuration parameter is set to COLUMN.

DFT_TABLE_ORG参数为Col，所以创建表默认是列存储表。

```
db2 connect to coltest
db2 "create table COL1_ORG (a int)"
db2 "create table ROW1_ORG (a int) organize by row"
db2 "create table COL2_ORG (a int) organize by column"
```

查看下表类型是row还是col：

```
db2 "select substr(tabname,1,30), tableorg  from syscat.tables where tabname like '%_ORG'"
```



```
1                                TABLEORG
-------------------------------- --------
SNAPTAB_REORG                    R
COL1_ORG                         C
SYN160312113532625608_COL1_ORG   C
ROW1_ORG                         R
COL2_ORG                         C
SYN160312113648752784_COL2_ORG   C

  6 record(s) selected.
```

```
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> cat /home/db2inst1/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts/data/rowcoltest.csv
1
2
3
4
5
6
7
8
9
10
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals>
```

Load数据到列存储表跟load到行表命令没有什么差别

```
db2 "load from /home/db2inst1/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts/data/rowcoltest.csv of del replace into COL1_ORG(a)"
db2 "load from /home/db2inst1/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts/data/rowcoltest.csv of del replace into ROW1_ORG(a)"
```

只是load过程有不一样：

```
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> db2 "load from /home/db2inst1/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts/data/rowcoltest.csv of del replace into COL1_ORG(a)"
SQL3501W  The table space(s) in which the table resides will not be placed in
backup pending state since forward recovery is disabled for the database.

SQL3109N  The utility is beginning to load data from file
"/home/db2inst1/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts/data".

SQL3500W  The utility is beginning the "ANALYZE" phase at time "03/12/2016
11:43:34.687319".

SQL3519W  Begin Load Consistency Point. Input record count = "0".

SQL3520W  Load Consistency Point was successful.

SQL3515W  The utility has finished the "ANALYZE" phase at time "03/12/2016
11:43:35.298741".

SQL3500W  The utility is beginning the "LOAD" phase at time "03/12/2016
11:43:35.300002".

SQL3110N  The utility has completed processing.  "10" rows were read from the
input file.

SQL3519W  Begin Load Consistency Point. Input record count = "10".

SQL3520W  Load Consistency Point was successful.

SQL3515W  The utility has finished the "LOAD" phase at time "03/12/2016
11:43:36.566061".

SQL3500W  The utility is beginning the "BUILD" phase at time "03/12/2016
11:43:36.568514".

SQL3213I  The indexing mode is "REBUILD".
```

```
SQL3515W  The utility has finished the "BUILD" phase at time "03/12/2016
11:43:37.255599".


Number of rows read         = 10
Number of rows skipped      = 0
Number of rows loaded       = 10
Number of rows rejected     = 0
Number of rows deleted      = 0
Number of rows committed    = 10

db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals>
```

col开始多了分析阶段：用于采样数据并构建压缩字典
row有index copy阶段，并且需要set integrity一致性检查

测试方法和统计可以借鉴学习：

创建测试数据库：

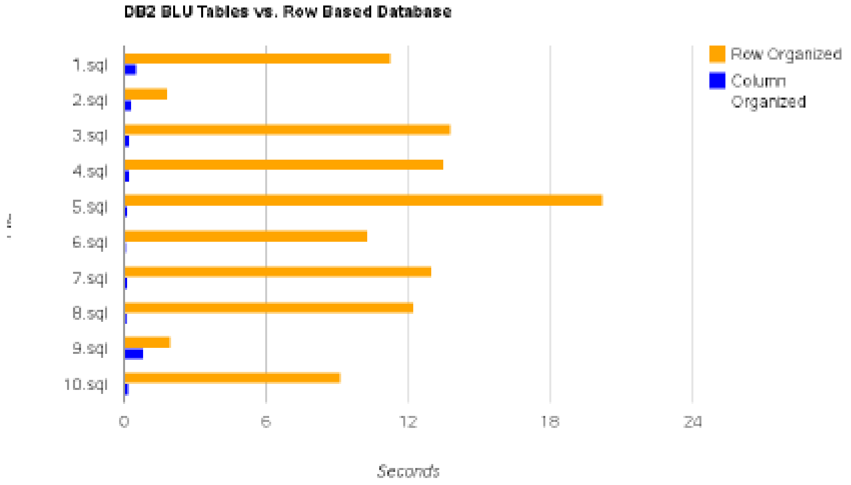
```
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> cat Build_Model_RowColumnTables.sh
#!/bin/bash
cd TestScripts/
./buildmart.sh
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals> cd TestScripts/
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts> cat buildmart.sh
#!/bin/sh
echo "nocomprow" > logs/state.txt
#erasing everything
#db2 connect reset
db2stop force
db2start
db2 drop database coltest
db2 drop database rowtest
db2 drop database rowctest

# row things
db2set DB2_WORKLOAD=
db2 create database rowtest
db2 connect to rowtest
db2 -tvf tables/dss.ddl.row > logs/bm/row/tables
db2 commit
before=$(date +%s.%N)
db2 -tvf data/dss.load > logs/bm/row/data
after=$(date +%s.%N)
difference=$(echo "$after - $before"|bc)
echo $difference | tee -a logs/loadspeed/rowloadspeed.txt
#db2 -tvf constraints/dss.ri > logs/bm/row/constraints
db2 connect reset



#column things
db2set DB2_WORKLOAD=ANALYTICS
db2 create database coltest
db2 connect to coltest
db2 -tvf tables/dss.ddl.col > logs/bm/col/tables
db2 commit
before=$(date +%s.%N)
db2 -tvf data/dss.load > logs/bm/col/data
```

```
after=$(date +%s.%N)
difference=$(echo "$after - $before"|bc)
echo $difference | tee -a logs/loadspeed/colloadspeed.txt
#db2 -tvf constraints/dss.ri > logs/bm/col/constraints
db2 connect reset
```

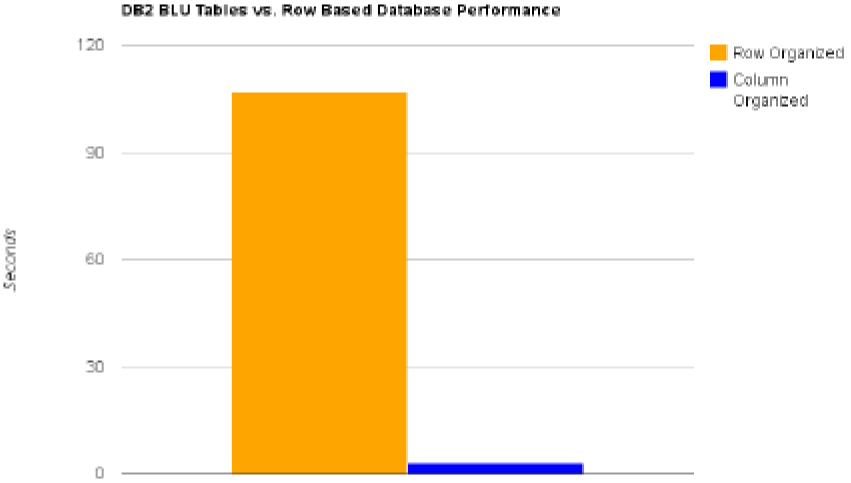
```
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts>
```

```
CREATE TABLE LINEITEM ( L_ORDERKEY    INTEGER NOT NULL,
                             L_PARTKEY     INTEGER NOT NULL,
                             L_SUPPKEY     INTEGER NOT NULL,
                             L_LINENUMBER  INTEGER NOT NULL,
                             L_QUANTITY    DECIMAL(15,2) NOT NULL,
                             L_EXTENDEDPRICE  DECIMAL(15,2) NOT NULL,
                             L_DISCOUNT    DECIMAL(15,2) NOT NULL,
                             L_TAX         DECIMAL(15,2) NOT NULL,
                             L_RETURNFLAG  CHAR(1) NOT NULL,
                             L_LINESTATUS  CHAR(1) NOT NULL,
                             L_SHIPDATE    DATE NOT NULL,
                             L_COMMITDATE  DATE NOT NULL,
                             L_RECEIPTDATE DATE NOT NULL,
                             L_SHIPINSTRUCT CHAR(25) NOT NULL,
                             L_SHIPMODE     CHAR(10) NOT NULL,
                              L_COMMENT      VARCHAR(44) NOT NULL) ORGANIZE BY COLUMN ;

CREATE TABLE LINEITEM ( L_ORDERKEY    INTEGER NOT NULL,
                             L_PARTKEY     INTEGER NOT NULL,
                             L_SUPPKEY     INTEGER NOT NULL,
                             L_LINENUMBER  INTEGER NOT NULL,
                             L_QUANTITY    DECIMAL(15,2) NOT NULL,
                             L_EXTENDEDPRICE  DECIMAL(15,2) NOT NULL,
                             L_DISCOUNT    DECIMAL(15,2) NOT NULL,
                             L_TAX         DECIMAL(15,2) NOT NULL,
                             L_RETURNFLAG  CHAR(1) NOT NULL,
                             L_LINESTATUS  CHAR(1) NOT NULL,
                             L_SHIPDATE    DATE NOT NULL,
                             L_COMMITDATE  DATE NOT NULL,
                             L_RECEIPTDATE DATE NOT NULL,
                             L_SHIPINSTRUCT CHAR(25) NOT NULL,
                             L_SHIPMODE     CHAR(10) NOT NULL,
                              L_COMMENT      VARCHAR(44) NOT NULL) ORGANIZE BY COLUMN COMPRESS YES ;


CREATE TABLE LINEITEM ( L_ORDERKEY    INTEGER NOT NULL,
                             L_PARTKEY     INTEGER NOT NULL,
                             L_SUPPKEY     INTEGER NOT NULL,
                             L_LINENUMBER  INTEGER NOT NULL,
                             L_QUANTITY    DECIMAL(15,2) NOT NULL,
                             L_EXTENDEDPRICE  DECIMAL(15,2) NOT NULL,
                             L_DISCOUNT    DECIMAL(15,2) NOT NULL,
                             L_TAX         DECIMAL(15,2) NOT NULL,
                             L_RETURNFLAG  CHAR(1) NOT NULL,
                             L_LINESTATUS  CHAR(1) NOT NULL,
                             L_SHIPDATE    DATE NOT NULL,
                             L_COMMITDATE  DATE NOT NULL,
                             L_RECEIPTDATE DATE NOT NULL,
                             L_SHIPINSTRUCT CHAR(25) NOT NULL,
                             L_SHIPMODE     CHAR(10) NOT NULL,
                              L_COMMENT      VARCHAR(44) NOT NULL) ORGANIZE BY ROW COMPRESS YES ;


CREATE TABLE LINEITEM ( L_ORDERKEY    INTEGER NOT NULL,
                             L_PARTKEY     INTEGER NOT NULL,
                             L_SUPPKEY     INTEGER NOT NULL,
                             L_LINENUMBER  INTEGER NOT NULL,
                             L_QUANTITY    DECIMAL(15,2) NOT NULL,
                             L_EXTENDEDPRICE  DECIMAL(15,2) NOT NULL,
                             L_DISCOUNT    DECIMAL(15,2) NOT NULL,
                             L_TAX         DECIMAL(15,2) NOT NULL,
                             L_RETURNFLAG  CHAR(1) NOT NULL,
                             L_LINESTATUS  CHAR(1) NOT NULL,
                             L_SHIPDATE    DATE NOT NULL,
                             L_COMMITDATE  DATE NOT NULL,
                             L_RECEIPTDATE DATE NOT NULL,
                             L_SHIPINSTRUCT CHAR(25) NOT NULL,
                             L_SHIPMODE     CHAR(10) NOT NULL,
                              L_COMMENT      VARCHAR(44) NOT NULL) ORGANIZE BY ROW ;
```

```
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts> cat benchmarkmart.sh
#!/bin/sh
#db2 connect reset
db2stop force
db2start
db2 connect to rowtest
#db2 update db cfg using SHEAPTHRES_SHR 164000 SORTHEAP 64000

echo "Running row queries"
rm logs/querytimes/rowspeed.txt
touch logs/querytimes/rowspeed.txt
cd queries
for f in `ls -v *.sql`
do
        echo "$f" | tee -a ../logs/querytimes/rowspeed.txt
        before=$(date +%s.%N)
        db2 -tvf $f > "../logs/rowq/Row $f.txt"
        after=$(date +%s.%N)
        difference=$(echo "$after - $before"|bc)
        echo $difference | tee -a ../logs/querytimes/rowspeed.txt

done

echo "getting database size"
db2 "CALL GET_DBSIZE_INFO(?, ?, ?, 0)" > ../logs/dbsize/rowsize.txt




db2 connect reset

if grep -q compressedrow ../logs/state.txt; then
        db2 connect to rowctest
        echo "Running compressed row queries"
        rm ../logs/querytimes/compressedrowspeed.txt
        touch ../logs/querytimes/compressedrowspeed.txt
        for f in `ls -v *.sql`
        do
                echo "$f" | tee -a ../logs/querytimes/compressedrowspeed.txt
                before=$(date +%s.%N)
                db2 -tvf $f > "../logs/rowcq/Compressed Row $f.txt"
                after=$(date +%s.%N)
                difference=$(echo "$after - $before"|bc)
                echo $difference | tee -a ../logs/querytimes/compressedrowspeed.txt
        done

echo "getting database size"
db2 "CALL GET_DBSIZE_INFO(?, ?, ?, 0)" > ../logs/dbsize/compressedrowsize.txt
fi

db2 connect reset

db2 connect to coltest
db2 update db cfg using SHEAPTHRES_SHR 164000 SORTHEAP 64000

echo "Running columnar queries"
rm ../logs/querytimes/colspeed.txt
touch ../logs/querytimes/colspeed.txt

for f in `ls -v *.sql`
do
        echo "$f" | tee -a ../logs/querytimes/colspeed.txt
        before=$(date +%s.%N)
        db2 -tvf $f > "../logs/colq/Column $f.txt"
        after=$(date +%s.%N)
        difference=$(echo "$after - $before"|bc)
        echo $difference | tee -a ../logs/querytimes/colspeed.txt
done

db2 "CALL GET_DBSIZE_INFO(?, ?, ?, 0)" > ../logs/dbsize/colsize.txt
cd ../results
python pagegen.py
python sizegraph.py

if wget -o /dev/null http://www.google.com  ; then
  nohup firefox benchmarks.html benchmarksTallied.html sizes.html &
else
  echo
  echo "Internet not available. Please check the following files."
  ls -1 logs/querytimes/*
fi
db2inst1@db2v105:~/Desktop/LabScripts/Lab-BLUFundamentals/TestScripts>
```



Query Speed Comparison



Database Performance