chosun@chosun-VirtualBox:~/hadoop$
/// ********** ///
>>> http://google.github.io/snappy/(스내피 공식 사이트)에서 snappy-1.1.3.tar.gz파일을 내려받았습니다.
>>> 스내피를 설치하기 위해 root계정으로 접속했습니다.
>>> 다운받은 snappy-1.1.3.tar.gz파일을 tar 명령어로 압축 해제했습니다.
/// ********** ///
chosun@chosun-VirtualBox:~/hadoop$ sudo -s
root@chosun-VirtualBox:~/hadoop-1.2.1#
root@chosun-VirtualBox:~/hadoop-1.2.1# tar xvfz ../다운로드/snappy-1.1.3.tar.gz
snappy-1.1.3/
snappy-1.1.3/snappy-sinksource.cc
snappy-1.1.3/configure
snappy-1.1.3/config.guess
snappy-1.1.3/snappy-c.cc
snappy-1.1.3/format_description.txt
snappy-1.1.3/snappy-stubs-internal.h
snappy-1.1.3/COPYING
snappy-1.1.3/configure.ac
snappy-1.1.3/snappy-sinksource.h
snappy-1.1.3/ltmain.sh
snappy-1.1.3/testdata/
snappy-1.1.3/testdata/urls.10K
snappy-1.1.3/testdata/baddata1.snappy
snappy-1.1.3/testdata/paper-100k.pdf
snappy-1.1.3/testdata/geo.protodata
snappy-1.1.3/testdata/html
snappy-1.1.3/testdata/baddata2.snappy
snappy-1.1.3/testdata/fireworks.jpeg
snappy-1.1.3/testdata/alice29.txt
snappy-1.1.3/testdata/plrabn12.txt
snappy-1.1.3/testdata/kppkn.gtb
snappy-1.1.3/testdata/baddata3.snappy
snappy-1.1.3/testdata/lcet10.txt
snappy-1.1.3/testdata/html_x_4
snappy-1.1.3/testdata/asyoulik.txt
snappy-1.1.3/INSTALL
snappy-1.1.3/snappy-stubs-internal.cc
snappy-1.1.3/config.sub
snappy-1.1.3/aclocal.m4
snappy-1.1.3/test-driver
snappy-1.1.3/m4/
snappy-1.1.3/m4/gtest.m4
snappy-1.1.3/snappy-internal.h
snappy-1.1.3/snappy-stubs-public.h
snappy-1.1.3/ChangeLog
snappy-1.1.3/depcomp
snappy-1.1.3/snappy.h
snappy-1.1.3/framing_format.txt
snappy-1.1.3/AUTHORS
snappy-1.1.3/snappy-stubs-public.h.in
snappy-1.1.3/Makefile.am
snappy-1.1.3/autogen.sh
snappy-1.1.3/README
snappy-1.1.3/snappy-test.cc
snappy-1.1.3/install-sh
snappy-1.1.3/snappy-c.h
snappy-1.1.3/snappy_unittest.cc
snappy-1.1.3/missing
snappy-1.1.3/compile
snappy-1.1.3/NEWS
snappy-1.1.3/snappy-test.h

```
snappy-1.1.3/snappy.cc
snappy-1.1.3/Makefile.in
snappy-1.1.3/config.h.in
root@chosun-VirtualBox:~/hadoop-1.2.1#
/// ********************************************************************************************** ///
>>> 압축을 해제하면 snappy-1.1.3 디렉터리가 생성됩니다.
>>> 이 디렉토리로 이동한 후, configure명령어를 사용해 make 파일을 생성했습니다.
/// ********************************************************************************************** ///
root@chosun-VirtualBox:~/hadoop-1.2.1# cd snappy-1.1.3/
root@chosun-VirtualBox:~/hadoop-1.2.1/snappy-1.1.3# ./configure --enable-shared
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking how to print strings... printf
checking for style of include used by make... GNU
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking dependency style of gcc... gcc3
checking for a sed that does not truncate output... /bin/sed
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for fgrep... /bin/grep -F
checking for ld used by gcc... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for BSD- or MS-compatible name lister (nm)... /usr/bin/nm -B
checking the name lister (/usr/bin/nm -B) interface... BSD nm
checking whether ln -s works... yes
checking the maximum length of command line arguments... 1572864
checking whether the shell understands some XSI constructs... yes
checking whether the shell understands "+="... yes
checking how to convert x86_64-unknown-linux-gnu file names to x86_64-unknown-linux-gnu format... func_convert_file_noop
checking how to convert x86_64-unknown-linux-gnu file names to toolchain format... func_convert_file_noop
checking for /usr/bin/ld option to reload object files... -r
checking for objdump... objdump
checking how to recognize dependent libraries... pass_all
checking for dlltool... no
checking how to associate runtime and link libraries... printf %s₩n
checking for ar... ar
checking for archiver @FILE support... @
checking for strip... strip
checking for ranlib... ranlib
checking command to parse /usr/bin/nm -B output from gcc object... ok
checking for sysroot... no
checking for mt... mt
checking if mt is a manifest tool... no
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
```

checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking for dlfcn.h... yes
checking for objdir... .libs
checking if gcc supports -fno-rtti -fno-exceptions... no
checking for gcc option to produce PIC... -fPIC -DPIC
checking if gcc PIC flag -fPIC -DPIC works... yes
checking if gcc static flag -static works... yes
checking if gcc supports -c -o file.o... yes
checking if gcc supports -c -o file.o... (cached) yes
checking whether the gcc linker (/usr/bin/ld -m elf_x86_64) supports shared libraries... yes
checking whether -lc should be explicitly linked in... no
checking dynamic linker characteristics... GNU/Linux ld.so
checking how to hardcode library paths into programs... immediate
checking whether stripping libraries is possible... yes
checking if libtool supports shared libraries... yes
checking whether to build shared libraries... yes
checking whether to build static libraries... yes
checking for g++... g++
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking dependency style of g++... gcc3
checking how to run the C++ preprocessor... g++ -E
checking for ld used by g++... /usr/bin/ld -m elf_x86_64
checking if the linker (/usr/bin/ld -m elf_x86_64) is GNU ld... yes
checking whether the g++ linker (/usr/bin/ld -m elf_x86_64) supports shared libraries... yes
checking for g++ option to produce PIC... -fPIC -DPIC
checking if g++ PIC flag -fPIC -DPIC works... yes
checking if g++ static flag -static works... yes
checking if g++ supports -c -o file.o... yes
checking if g++ supports -c -o file.o... (cached) yes
checking whether the g++ linker (/usr/bin/ld -m elf_x86_64) supports shared libraries... yes
checking dynamic linker characteristics... (cached) GNU/Linux ld.so
checking how to hardcode library paths into programs... immediate
checking whether byte ordering is bigendian... no
checking for size_t... yes
checking for ssize_t... yes
checking for stdint.h... (cached) yes
checking stddef.h usability... yes
checking stddef.h presence... yes
checking for stddef.h... yes
checking sys/mman.h usability... yes
checking sys/mman.h presence... yes
checking for sys/mman.h... yes
checking sys/resource.h usability... yes
checking sys/resource.h presence... yes
checking for sys/resource.h... yes
checking windows.h usability... no
checking windows.h presence... no
checking for windows.h... no
checking byteswap.h usability... yes
checking byteswap.h presence... yes
checking for byteswap.h... yes
checking sys/byteswap.h usability... no

checking sys/byteswap.h presence... no
checking for sys/byteswap.h... no
checking sys/endian.h usability... no
checking sys/endian.h presence... no
checking for sys/endian.h... no
checking sys/time.h usability... yes
checking sys/time.h presence... yes
checking for sys/time.h... yes
checking for mmap... yes
checking for 'gtest-config'... checking for gtest-config... no
no
checking for pkg-config... no
checking for gflags... no
checking if the compiler supports __builtin_expect... yes
checking if the compiler supports __builtin_ctzll... yes
checking for zlibVersion in -lz... no
checking for lzo1x_1_15_compress in -llzo2... no
checking for lzf_compress in -llzf... no
checking for fastlz_compress in -lfastlz... no
checking for qlz_compress in -lquicklz... no
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating snappy-stubs-public.h
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
root@chosun-VirtualBox:~/hadoop-1.2.1/snappy-1.1.3#
/// ** ///
>>> make파일이 생성되었으므로, 소스코드를 컴파일 해서 바이너리 파일을 생성했습니다.
/// ** ///
root@chosun-VirtualBox:~/hadoop-1.2.1/snappy-1.1.3# make
make all-am
make[1]: 디렉터리 '/home/chosun/hadoop-1.2.1/snappy-1.1.3' 들어감
/bin/bash ./libtool --tag=CXX --mode=compile g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy.lo -MD -MP -MF .deps/snappy.Tpo -c -o snappy.lo snappy.cc
libtool: compile: g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy.lo -MD -MP -MF .deps/snappy.Tpo -c snappy.cc -fPIC -DPIC -o .libs/snappy.o
libtool: compile: g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy.lo -MD -MP -MF .deps/snappy.Tpo -c snappy.cc -o snappy.o >/dev/null 2>&1
mv -f .deps/snappy.Tpo .deps/snappy.Plo
/bin/bash ./libtool --tag=CXX --mode=compile g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-sinksource.lo -MD -MP -MF .deps/snappy-sinksource.Tpo -c -o snappy-sinksource.lo snappy-sinksource.cc
libtool: compile: g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-sinksource.lo -MD -MP -MF .deps/snappy-sinksource.Tpo -c snappy-sinksource.cc -fPIC -DPIC -o .libs/snappy-sinksource.o
libtool: compile: g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-sinksource.lo -MD -MP -MF .deps/snappy-sinksource.Tpo -c snappy-sinksource.cc -o snappy-sinksource.o >/dev/null 2>&1
mv -f .deps/snappy-sinksource.Tpo .deps/snappy-sinksource.Plo
/bin/bash ./libtool --tag=CXX --mode=compile g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-stubs-internal.lo -MD -MP -MF .deps/snappy-stubs-internal.Tpo -c -o snappy-stubs-internal.lo snappy-stubs-internal.cc
libtool: compile: g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-stubs-internal.lo -MD -MP -MF .deps/snappy-stubs-internal.Tpo -c snappy-stubs-internal.cc -fPIC -DPIC -o .libs/snappy-stubs-internal.o
libtool: compile: g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-stubs-internal.lo -MD -MP -MF .deps/snappy-stubs-internal.Tpo -c snappy-stubs-internal.cc -o snappy-stubs-internal.o >/dev/null 2>&1
mv -f .deps/snappy-stubs-internal.Tpo .deps/snappy-stubs-internal.Plo
/bin/bash ./libtool --tag=CXX --mode=compile g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-c.lo -MD -MP -MF .deps/snappy-c.Tpo -c -o snappy-c.lo snappy-c.cc
libtool: compile: g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-c.lo -MD -MP -MF .deps/snappy-c.Tpo -c snappy-c.cc -fPIC -DPIC -o .libs/snappy-c.o
libtool: compile: g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy-c.lo -MD -MP -MF .deps/snappy-c.Tpo -c snappy-c.cc -o snappy-c.o >/dev/null 2>&1

```
mv -f .deps/snappy-c.Tpo .deps/snappy-c.Plo
/bin/bash ./libtool --tag=CXX --mode=link g++ -g -O2 -version-info 4:0:3 -o libsnappy.la -rpath /usr/local/lib snappy.lo snappy-sinksource.lo snappy-stubs-internal.lo snappy-c.lo
libtool: link: g++ -fPIC -DPIC -shared -nostdlib /usr/lib/gcc/x86_64-linux-gnu/7/../../../x86_64-linux-gnu/crti.o /usr/lib/gcc/x86_64-linux-gnu/7/crtbeginS.o .libs/snappy.o .libs/snappy-sinksource.o .libs/snappy-stubs-internal.o .libs/snappy-c.o -L/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7/../../../x86_64-linux-gnu -L/usr/lib/gcc/x86_64-linux-gnu/7/../../../../lib -L/lib/x86_64-linux-gnu -L/lib/../lib -L/usr/lib/x86_64-linux-gnu -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-linux-gnu/7/../../.. -lstdc++ -lm -lc -lgcc_s /usr/lib/gcc/x86_64-linux-gnu/7/crtendS.o /usr/lib/gcc/x86_64-linux-gnu/7/../../../x86_64-linux-gnu/crtn.o -O2 -Wl,-soname -Wl,libsnappy.so.1 -o .libs/libsnappy.so.1.3.0
libtool: link: (cd ".libs" && rm -f "libsnappy.so.1" && ln -s "libsnappy.so.1.3.0" "libsnappy.so.1")
libtool: link: (cd ".libs" && rm -f "libsnappy.so" && ln -s "libsnappy.so.1.3.0" "libsnappy.so")
libtool: link: ar cru .libs/libsnappy.a snappy.o snappy-sinksource.o snappy-stubs-internal.o snappy-c.o
ar: `u' modifier ignored since `D' is the default (see `U')
libtool: link: ranlib .libs/libsnappy.a
libtool: link: ( cd ".libs" && rm -f "libsnappy.la" && ln -s "../libsnappy.la" "libsnappy.la" )
g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy_unittest-snappy_unittest.o -MD -MP -MF .deps/snappy_unittest-snappy_unittest.Tpo -c -o snappy_unittest-snappy_unittest.o `test -f 'snappy_unittest.cc' || echo './'`snappy_unittest.cc
mv -f .deps/snappy_unittest-snappy_unittest.Tpo .deps/snappy_unittest-snappy_unittest.Po
g++ -DHAVE_CONFIG_H -I. -g -O2 -MT snappy_unittest-snappy-test.o -MD -MP -MF .deps/snappy_unittest-snappy-test.Tpo -c -o snappy_unittest-snappy-test.o `test -f 'snappy-test.cc' || echo './'`snappy-test.cc
mv -f .deps/snappy_unittest-snappy-test.Tpo .deps/snappy_unittest-snappy-test.Po
/bin/bash ./libtool --tag=CXX --mode=link g++ -g -O2 -o snappy_unittest snappy_unittest-snappy_unittest.o snappy_unittest-snappy-test.o libsnappy.la
libtool: link: g++ -g -O2 -o .libs/snappy_unittest snappy_unittest-snappy_unittest.o snappy_unittest-snappy-test.o ./.libs/libsnappy.so
make[1]: 디렉터리 '/home/chosun/hadoop-1.2.1/snappy-1.1.3' 나감
root@chosun-VirtualBox:~/hadoop-1.2.1/snappy-1.1.3#
/// ********************************************************************************************* ///
>>> 생성된 바이너리 파일을 Snappy에서 지정한 디렉토리로 옮겼습니다.
/// ********************************************************************************************* ///
root@chosun-VirtualBox:~/hadoop-1.2.1/snappy-1.1.3# make install
make[1]: 디렉터리 '/home/chosun/hadoop-1.2.1/snappy-1.1.3' 들어감
 /bin/mkdir -p '/usr/local/lib'
 /bin/bash ./libtool --mode=install /usr/bin/install -c libsnappy.la '/usr/local/lib'
libtool: install: /usr/bin/install -c .libs/libsnappy.so.1.3.0 /usr/local/lib/libsnappy.so.1.3.0
libtool: install: (cd /usr/local/lib && { ln -s -f libsnappy.so.1.3.0 libsnappy.so.1 || { rm -f libsnappy.so.1 && ln -s libsnappy.so.1.3.0 libsnappy.so.1; }; })
libtool: install: (cd /usr/local/lib && { ln -s -f libsnappy.so.1.3.0 libsnappy.so || { rm -f libsnappy.so && ln -s libsnappy.so.1.3.0 libsnappy.so; }; })
libtool: install: /usr/bin/install -c .libs/libsnappy.lai /usr/local/lib/libsnappy.la
libtool: install: /usr/bin/install -c .libs/libsnappy.a /usr/local/lib/libsnappy.a
libtool: install: chmod 644 /usr/local/lib/libsnappy.a
libtool: install: ranlib /usr/local/lib/libsnappy.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/home/chosun/hadoop/bin/:.:/sbin" ldconfig -n /usr/local/lib
----------------------------------------------------------------------
Libraries have been installed in:
   /usr/local/lib
If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the `-LLIBDIR'
flag during linking and do at least one of the following:
   - add LIBDIR to the `LD_LIBRARY_PATH' environment variable
     during execution
   - add LIBDIR to the `LD_RUN_PATH' environment variable
     during linking
   - use the `-Wl,-rpath -Wl,LIBDIR' linker flag
   - have your system administrator add LIBDIR to `/etc/ld.so.conf'
See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
----------------------------------------------------------------------
 /bin/mkdir -p '/usr/local/share/doc/snappy'
```

```
 /usr/bin/install -c -m 644 ChangeLog COPYING INSTALL NEWS README format_description.txt framing_format.txt
'/usr/local/share/doc/snappy'
 /bin/mkdir -p '/usr/local/include'
 /usr/bin/install -c -m 644 snappy.h snappy-sinksource.h snappy-stubs-public.h snappy-c.h '/usr/local/include'
make[1]: 디렉터리 '/home/chosun/hadoop-1.2.1/snappy-1.1.3' 나감
root@chosun-VirtualBox:~/hadoop-1.2.1#
root@chosun-VirtualBox:~/hadoop-1.2.1/snappy-1.1.3#  cd ..
root@chosun-VirtualBox:~/hadoop-1.2.1#
/// ********************************************************************************************** ///
>>> 스내피 설치가 완료되었습니다.
>>> 해당 라이브러리를 하둡의 네이티브 라이브러리 디렉토리로 복사해야합니다.
>>> 64bit 운영체제를 사용하고 있으므로 Linux-amd64-64 디렉토리로 라이브러리를 복사했습니다.
>>> root계정을 나와서 chosun계정으로 돌아왔습니다.
/// ********************************************************************************************** ///
root@chosun-VirtualBox:~/hadoop-1.2.1#
root@chosun-VirtualBox:~/hadoop-1.2.1# cp /usr/local/lib/libsnappy.* ./lib/native/Linux-amd64-64
root@chosun-VirtualBox:~/hadoop-1.2.1#
root@chosun-VirtualBox:~/hadoop-1.2.1# exit
exit
chosun@chosun-VirtualBox:~/hadoop$
/// ********************************************************************************************** ///
>>> 복사가 완료되면 맵리듀스 클러스터를 재구동합니다.
>>> stop-mapred.sh명령어를 사용한 후 jps를 실행시켜보면 정상적으로 TaskTracker JobTracker이 종료된 것을 확인합니다.
>>> start-mapred.sh명령어를 사용한 후 jps를 실행시켜보면 정상적으로 TaskTracker JobTracker이 실행된 것을 확인합니다.
/// ********************************************************************************************** ///
chosun@chosun-VirtualBox:~/hadoop$
chosun@chosun-VirtualBox:~/hadoop$ stop-mapred.sh
stopping jobtracker
localhost: stopping tasktracker
chosun@chosun-VirtualBox:~/hadoop$
chosun@chosun-VirtualBox:~/hadoop$ jps
12772 DataNode
12985 SecondaryNameNode
14556 Jps
12559 NameNode
chosun@chosun-VirtualBox:~/hadoop$
chosun@chosun-VirtualBox:~/hadoop$ start-mapred.sh
starting jobtracker, logging to /home/chosun/hadoop-1.2.1/libexec/../logs/hadoop-chosun-jobtracker-chosun-VirtualBox.out
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/home/chosun/hadoop-1.2.1/
hadoop-core-1.2.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
localhost: starting tasktracker, logging to /home/chosun/hadoop-1.2.1/libexec/../logs/hadoop-chosun-tasktracker-chosun-VirtualBox.out
localhost: WARNING: An illegal reflective access operation has occurred
localhost: WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/home/chosun/hadoop-1.2.1/
hadoop-core-1.2.1.jar) to method sun.security.krb5.Config.getInstance()
localhost: WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
localhost: WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
localhost: WARNING: All illegal access operations will be denied in a future release
chosun@chosun-VirtualBox:~/hadoop$
chosun@chosun-VirtualBox:~/hadoop$ jps
14851 TaskTracker
12772 DataNode
12985 SecondaryNameNode
14938 Jps
14618 JobTracker
12559 NameNode
chosun@chosun-VirtualBox:~/hadoop$
chosun@chosun-VirtualBox:~/hadoop$
```

/// ********** ///
>>> jar파일을 실행합니다.
>>> HW_0512 디렉토리의 파일(2008.csv)을 jar파일로 처리한 것을 HW_0515_delay_snappy디렉토리에 저장합니다.
>>> ArrivalDelayCount는 클래스 이름입니다.
>>> 스내피를 적용한 상태에서 HWEx_5.5.jar 파일을 실행시켰습니다.
>>> 스내피를 적용하지 않은 HWEx5_5, 스내피 적용한 HWEx5_5, Gzip적용한 HWEx8_3의 실행결과를 비교합니다.
>>> 우선 잡 실행 로그가 Snappy native library not loaded였던 것이 Snappy native library is available로 출력되는 것을 확인했습니다.
/// ********** ///
chosun@chosun-VirtualBox:~/hadoop$ hadoop jar HWEx5_5.jar ArrivalDelayCount HW_0512 HW_0515delay_snappy
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/home/chosun/hadoop-1.2.1/hadoop-core-1.2.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
19/05/16 19:27:41 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
19/05/16 19:27:41 INFO input.FileInputFormat: Total input paths to process : 1
19/05/16 19:27:41 WARN snappy.LoadSnappy: Snappy native library is available // snappy 적용됨
19/05/16 19:27:41 INFO util.NativeCodeLoader: Loaded the native-hadoop library
19/05/16 19:27:41 INFO snappy.LoadSnappy: Snappy native library loaded
19/05/16 19:27:41 INFO mapred.JobClient: Running job: job_201905161858_0002 // job 시작
19/05/16 19:27:42 INFO mapred.JobClient: map 0% reduce 0% // 맵리듀스 시작
19/05/16 19:27:47 INFO mapred.JobClient: map 18% reduce 0% // ~~ ing
19/05/16 19:27:51 INFO mapred.JobClient: map 36% reduce 0%
19/05/16 19:27:55 INFO mapred.JobClient: map 54% reduce 12%
19/05/16 19:27:58 INFO mapred.JobClient: map 72% reduce 12%
19/05/16 19:28:01 INFO mapred.JobClient: map 81% reduce 12%
19/05/16 19:28:02 INFO mapred.JobClient: map 90% reduce 12%
19/05/16 19:28:03 INFO mapred.JobClient: map 100% reduce 12%
19/05/16 19:28:04 INFO mapred.JobClient: map 100% reduce 24% // ~~ ing
19/05/16 19:28:07 INFO mapred.JobClient: map 100% reduce 100% // 맵 끝
19/05/16 19:28:07 INFO mapred.JobClient: Job complete: job_201905161858_0002 // job 끝
19/05/16 19:28:07 INFO mapred.JobClient: Counters: 29
19/05/16 19:28:07 INFO mapred.JobClient: Map-Reduce Framework
19/05/16 19:28:07 INFO mapred.JobClient: Spilled Records=8401465 // 작업 중에 디스크에 기록 된 총 레코드 수
19/05/16 19:28:07 INFO mapred.JobClient: Map output materialized bytes=39379199 // 실제로 디스크에 기록된 맵 출력 바이트 수
19/05/16 19:28:07 INFO mapred.JobClient: Reduce input records=2979504 // 리듀스에 입력된 데이터의 수
19/05/16 19:28:07 INFO mapred.JobClient: Virtual memory (bytes) snapshot=25076244480 // 모든 작업에 사용된 총 가상 메모리
19/05/16 19:28:07 INFO mapred.JobClient: Map input records=7009728 // 맵에 입력되는 데이터의 수
19/05/16 19:28:07 INFO mapred.JobClient: SPLIT_RAW_BYTES=1265 // 맵에서 가져온 입력 분할 개체의 바이트 수
19/05/16 19:28:07 INFO mapred.JobClient: Map output bytes=33420125 // 모든 맵에 의해 생성된 압축되지 않은 출력 바이트수
19/05/16 19:28:07 INFO mapred.JobClient: Reduce shuffle bytes=39379199 // 리듀스 단계에 셔플된 맵 출력의 바이트 수
19/05/16 19:28:07 INFO mapred.JobClient: Physical memory (bytes) snapshot=2892034048 // 모든 작업에 사용되는 총 물리적 메모리
19/05/16 19:28:07 INFO mapred.JobClient: Reduce input groups=12 // 리듀스 단계에서 처리한 개별 키 그룹의 수
19/05/16 19:28:07 INFO mapred.JobClient: Combine output records=0 // 출력 레코드 수
19/05/16 19:28:07 INFO mapred.JobClient: Reduce output records=12 // 리듀스의 출력으로 생성된 데이터의 수
19/05/16 19:28:07 INFO mapred.JobClient: Map output records=2979504 // 맵의 출력으로 생성된 데이터의 수
19/05/16 19:28:07 INFO mapred.JobClient: Combine input records=0 // 입력 레코드 수
19/05/16 19:28:07 INFO mapred.JobClient: CPU time spent (ms)=38400 // 소요된 cpu시간
19/05/16 19:28:07 INFO mapred.JobClient: Total committed heap usage (bytes)=2022703104 // JVM에 사용할 수 있는 총 메모리 양
19/05/16 19:28:07 INFO mapred.JobClient: File Input Format Counters
19/05/16 19:28:07 INFO mapred.JobClient: Bytes Read=689454004 // 모든 파일 시스템에 대해 모든 테스크에서 읽은 데이터 양
19/05/16 19:28:07 INFO mapred.JobClient: FileSystemCounters
19/05/16 19:28:07 INFO mapred.JobClient: HDFS_BYTES_READ=689455269 // HDFS에서 읽은 데이터 양
19/05/16 19:28:07 INFO mapred.JobClient: FILE_BYTES_WRITTEN=111502989 // 로컬 파일 시스템에 기록된 데이터 양
19/05/16 19:28:07 INFO mapred.JobClient: FILE_BYTES_READ=71433572 // 로컬 파일 시스템에서 읽은 데이터 양
19/05/16 19:28:07 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=171 // HDFS에 기록된 데이터 양
19/05/16 19:28:07 INFO mapred.JobClient: Job Counters
19/05/16 19:28:07 INFO mapred.JobClient: Launched map tasks=11 // 실행된 총 맵 작업 수
19/05/16 19:28:07 INFO mapred.JobClient: Launched reduce tasks=1 // 실행된 총 리듀스 작업 수

```
19/05/16 19:28:07 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=20243              // 모든 리듀스 작업에 의해 소요된 총 시간(ms)
19/05/16 19:28:07 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
19/05/16 19:28:07 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=37238                 // 모든 맵 작업에 의해 사용한 총 시간(ms)
19/05/16 19:28:07 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
19/05/16 19:28:07 INFO mapred.JobClient:     Data-local map tasks=11          // 필요한 데이터가 포함된 노드에서 시작된 맵 테스크 수
19/05/16 19:28:07 INFO mapred.JobClient:   File Output Format Counters
19/05/16 19:28:07 INFO mapred.JobClient:     Bytes Written=171               // 모든 파일 시스템에 대해 모든 태스크에서 작성된 데이터 양
chosun@chosun-VirtualBox:~/hadoop$
/// ********************************************************************************************** ///
```

>>> 스내피가 적용되기 전과 후의 실행 결과를 비교합니다.

카운터	Gzip, 스내피 x	Gzip 적용 o	스내피 적용 o
01. FILE_BYTES_WRITTEN	**111502929**	**906833**	**111502989**
02. HDFS_BYTES_READ	689455269	689455269	689455269
03. FILE_BYTES_READ	**71433572**	**139601**	**71433572**
04. HDFS_BYTES_WRITTEN	171	324	171
05. Reduce shuffle bytes	**39379199**	**76990**	**39379199**
06. Spilled Records	8401465	8401465	8401465
07. CPU time spent (ms)	40260	56980	38400
08. Total committed heap usage	1993342976	1995440128	2022703104
09. Reduce input records	2979504	2979504	2979504
10. Physical memory (bytes) snapshot	2803625984	2808045568	2835308544
11. SLOTS_MILLIS_MAPS	42821	51746	37238
12. SLOTS_MILLIS_REDUCES	25784	35839	20243

Gzip을 적용했을 때 대부분의 수치가 눈에 띄게 개선했음을 알 수 있습니다.
이렇게 파일입출력이 감소한 폭만 봤을 때는 Gzip의 성능이 좋아보이나, (01, 03, 05)
맵리듀스 잡 수행 시간과 CPU 점유율은 스내피가 더 좋은 성능을 보여줍니다. (07, 11, 12)

>>> 압축률이 중요할 때는 Gzip을, 속도가 중요할 때는 스내피를 사용하는 것이 좋습니다.

/// ** ///