CS 571

REPORT

FINAL PROJECT

BY

DHAVAL KAPGATE

RAKSHA SINDHU

ANUJ PATEL

SHARANYA SHIVAKUMAR

STUDENT CONTRIBUTION:

DHAVAL KAPGATE:

Worked on developing C and PROLOG. Also integrated it with JAVA driver. The code available on the internet for C was for 9*9 sudoku. So scaled it up to work for 16*16 sudoku. All the required changes in the logic were done so that it can be integrated with JAVA.

ANUJ PATEL:

Worked on developing JavaScript. Also integrated it with JAVA driver. The code available on the internet for JavaScript was for 9*9 sudoku. So scaled it up to work for 16*16 sudoku. All the required changes in the logic were done so that it can be integrated with JAVA.

RAKSHA SINDHU:

Worked on developing JAVA. Also integrated it with JAVA driver. The code available on the internet for JAVA was for 9*9 sudoku. So scaled it up to work for 16*16 sudoku. All the required changes in the logic were done so that it can be integrated with JAVA.

SHARANYA SHIVAKUMAR:

Worked on developing Python. Also integrated it with JAVA driver. The code available on the internet for Python was for 16*16 sudoku. All the required changes in the logic were done so that it can be integrated with JAVA. The grid output was converted to string.

IMPLEMENTAION:

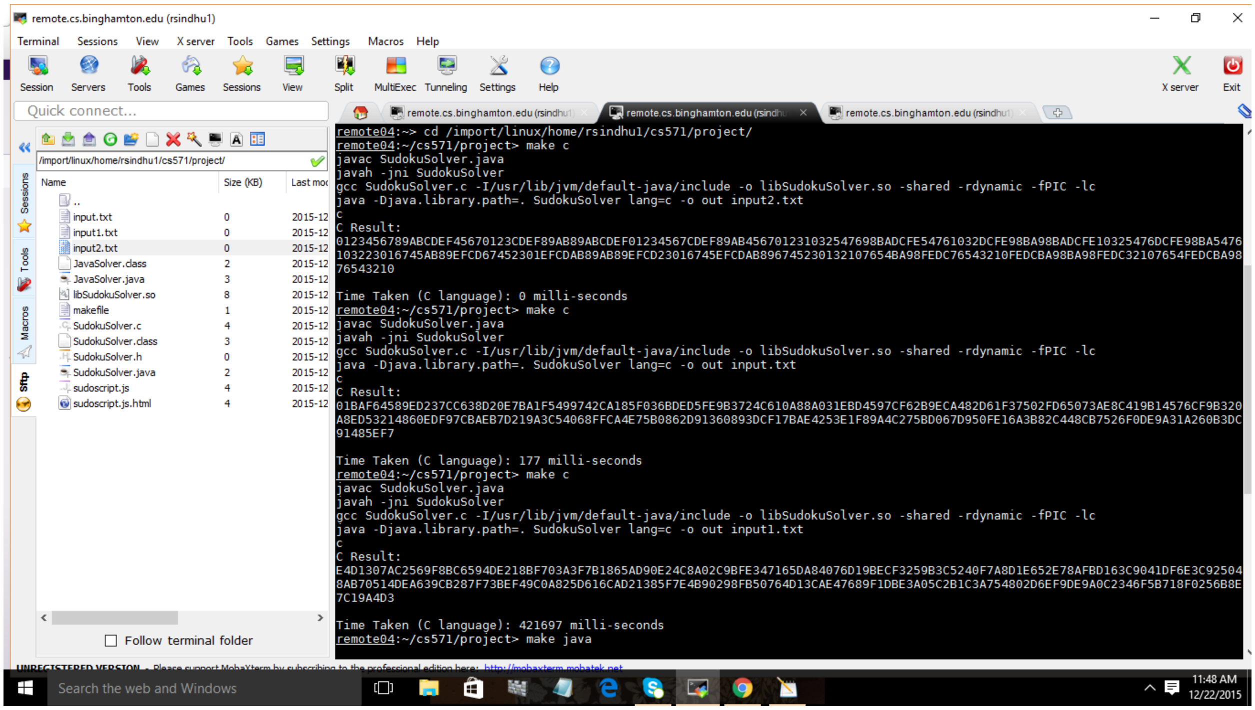
C: On executing the code from java driver it was analyzed that the integration is proper also we get the output as a string with the below execution time.

Easy= 0ms

Medium= 177ms

Hard=421697ms

JAVA: Proper integration was done with Java. It was analyzed that the string input gives the solved output. Java takes the following execution time.


remote.cs.binghamton.edu (rsindhu1)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Settings Help X server Exit
Quick connect...
remote.cs.binghamton.edu (rsindhu1)
remote.cs.binghamton.edu (rsindhu1)
remote.cs.binghamton.edu (rsindhu1)
Sessions Tools Macros Sftp
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net
Search the web and Windows
11:48 AM
12/22/2015

/import/linux/home/rsindhu1/cs571/project/

Name	Size (KB)	Last moc
..		
input.txt	0	2015-12
input1.txt	0	2015-12
input2.txt	0	2015-12
JavaSolver.class	2	2015-12
JavaSolver.java	3	2015-12
libSudokuSolver.so	8	2015-12
makefile	1	2015-12
SudokuSolver.c	4	2015-12
SudokuSolver.class	3	2015-12
SudokuSolver.h	0	2015-12
SudokuSolver.java	2	2015-12
sudoscript.js	4	2015-12
sudoscript.js.html	4	2015-12

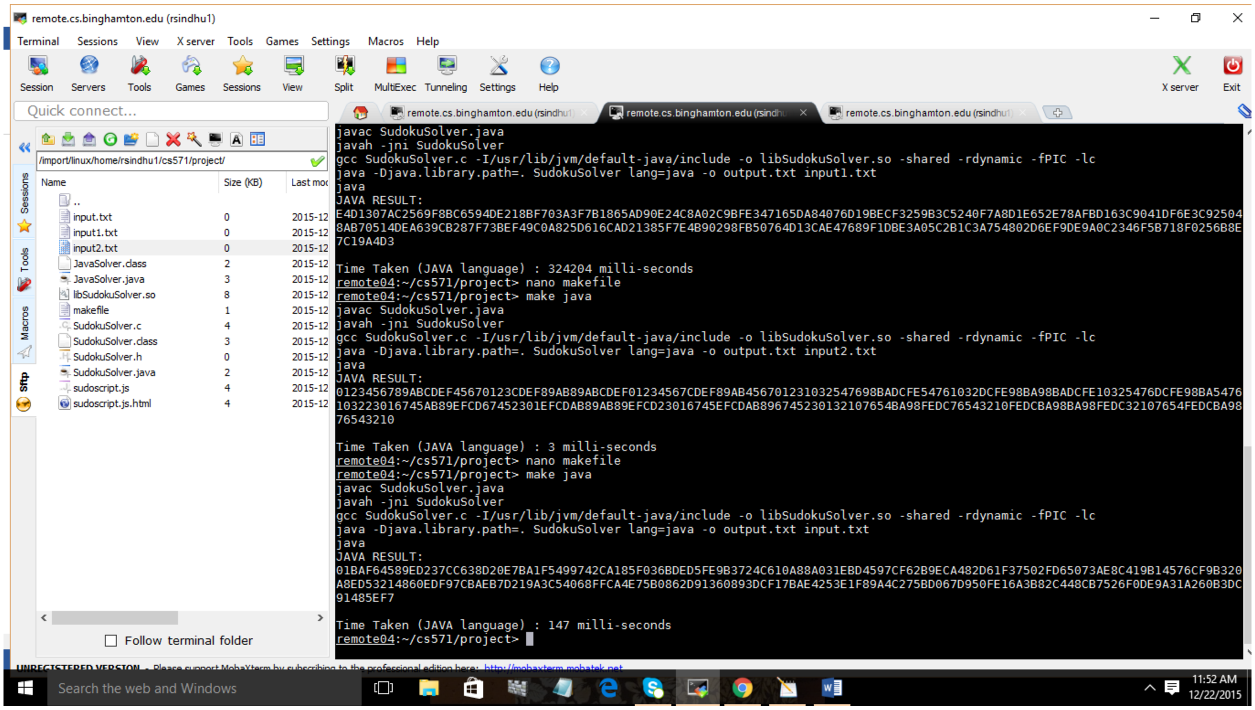
```
remote04:~> cd /import/linux/home/rsindhu1/cs571/project/
remote04:~/cs571/project> make c
javac SudokuSolver.java
javah -jni SudokuSolver
gcc SudokuSolver.c -I/usr/lib/jvm/default-java/include -o libSudokuSolver.so -shared -rdynamic -fPIC -lc
java -Djava.library.path=. SudokuSolver lang=c -o out input2.txt
c
C Result:
0123456789ABCDEF45670123CDEF89AB89ABCDEF01234567CDEF89AB4567012310325476 98BADCFE54761032DCFE98BA98BADCFE10325476DCFE98BA5476
103223016745AB89EFCD67452301EFCDAB89AB89EFCD23016745EFCDAB896745230132107654BA98FEDC76543210FEDCBA98BA98FEDC32107654FEDCBA98
76543210

Time Taken (C language): 0 milli-seconds
remote04:~/cs571/project> make c
javac SudokuSolver.java
javah -jni SudokuSolver
gcc SudokuSolver.c -I/usr/lib/jvm/default-java/include -o libSudokuSolver.so -shared -rdynamic -fPIC -lc
java -Djava.library.path=. SudokuSolver lang=c -o out input.txt
c
C Result:
01BAF64589ED237CC638D20E7BA1F5499742CA185F036BDED5FE9B3724C610A88A031EBD4597CF62B9ECA482D61F37502FD65073AE8C419B14576CF9B320
A8ED53214860EDF97CBAEB7D219A3C54068FFCA4E75B0862D91360893DCF17BAE4253E1F89A4C275BD067D950FE16A3B82C448CB7526F0DE9A31A260B3DC
91485EF7

Time Taken (C language): 177 milli-seconds
remote04:~/cs571/project> make c
javac SudokuSolver.java
javah -jni SudokuSolver
gcc SudokuSolver.c -I/usr/lib/jvm/default-java/include -o libSudokuSolver.so -shared -rdynamic -fPIC -lc
java -Djava.library.path=. SudokuSolver lang=c -o out input1.txt
c
C Result:
E4D1307AC2569F8BC6594DE218BF703A3F7B1865AD90E24C8A02C9BFE347165DA84076D19BECF3259B3C5240F7A8D1E652E78AFBD163C9041DF6E3C92504
8AB70514DEA639CB287F73BEF49C0A825D616CAD21385F7E4B90298FB50764D13CAE47689F1DBE3A05C2B1C3A754802D6EF9DE9A0C2346F5B718F0256B8E
7C19A4D3

Time Taken (C language): 421697 milli-seconds
remote04:~/cs571/project> make java
```

Easy = 3ms

Medium = 147ms

Hard = 324204ms

JavaScript: The integration is proper with Java driver. It takes the input from java driver and gives out the output in string format. It has the following execution time.



remote.cs.binghamton.edu (rsindhu1)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Settings Help X server Exit
Quick connect...
remote.cs.binghamton.edu (rsindhu1)
Sessions Tools Macros Sftp
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net
Search the web and Windows
11:52 AM
12/22/2015

/import/linux/home/rsindhu1/cs571/project/

Name	Size (KB)	Last mod
..		
input.txt	0	2015-12
input1.txt	0	2015-12
input2.txt	0	2015-12
JavaSolver.class	2	2015-12
JavaSolver.java	3	2015-12
libSudokuSolver.so	8	2015-12
makefile	1	2015-12
SudokuSolver.c	4	2015-12
SudokuSolver.class	3	2015-12
SudokuSolver.h	0	2015-12
SudokuSolver.java	2	2015-12
sudoscript.js	4	2015-12
sudoscript.js.html	4	2015-12

```
javac SudokuSolver.java
javah -jni SudokuSolver
gcc SudokuSolver.c -I/usr/lib/jvm/default-java/include -o libSudokuSolver.so -shared -rdynamic -fPIC -lc
java -Djava.library.path=. SudokuSolver lang=java -o output.txt input1.txt
java
JAVA RESULT:
E4D1307AC2569F8BC6594DE218BF703A3F7B1865AD90E24C8A02C9BFE347165DA84076D19BECF3259B3C5240F7A8D1E652E78AFBD163C9041DF6E3C92504
8AB70514DEA639CB287F73BEF49C0A825D616CAD21385F7E4B90298FB50764D13CAE47689F1DBE3A05C2B1C3A754802D6EF9DE9A0C2346F5B718F0256B8E
7C19A4D3

Time Taken (JAVA language) : 324204 milli-seconds
remote04:~/cs571/project> nano makefile
remote04:~/cs571/project> make java
javac SudokuSolver.java
javah -jni SudokuSolver
gcc SudokuSolver.c -I/usr/lib/jvm/default-java/include -o libSudokuSolver.so -shared -rdynamic -fPIC -lc
java -Djava.library.path=. SudokuSolver lang=java -o output.txt input2.txt
java
JAVA RESULT:
0123456789ABCDEF45670123CDEF89AB89ABCDEF01234567CDEF89AB4567012310325476988ADCFE54761032DCFE98BA98BADCFE10325476DCFE98BA5476
103223016745AB89EFCD67452301EFCDAB89AB89EFCD23016745EFCDAB89674523013210765 4BA98FEDC76543210FEDCBA98BA98FEDC32107654FEDCBA98
76543210

Time Taken (JAVA language) : 3 milli-seconds
remote04:~/cs571/project> nano makefile
remote04:~/cs571/project> make java
javac SudokuSolver.java
javah -jni SudokuSolver
gcc SudokuSolver.c -I/usr/lib/jvm/default-java/include -o libSudokuSolver.so -shared -rdynamic -fPIC -lc
java -Djava.library.path=. SudokuSolver lang=java -o output.txt input.txt
java
JAVA RESULT:
01BAF64589ED237CC638D20E7BA1F5499742CA185F036BDED5FE9B3724C610A88A031EBD4597CF62B9ECA482D61F37502FD65073AE8C419B14576CF9B320
A8ED53214860EDF97CBAEB7D219A3C54068FFCA4E75B0862D91360893DCF17BAE4253E1F89A4C275BD067D950FE16A3B82C448CB7526F0DE9A31A260B3DC
91485EF7

Time Taken (JAVA language) : 147 milli-seconds
remote04:~/cs571/project> 
```

Easy: 1855ms

Python: The integration with Java driver is possible. The grid generated at the Python end is converted into a string and returned back to the Java driver.

Easy: 374ms

Medium: 178ms

Hard: 179ms

Prolog: The integration of Prolog with Java was not possible. But the prolog program of 16*16 Sudoku works perfectly on swish. With the following execution time.

Hard: 470ms

More the number of blanks the less it has to check for the combinations of Sudoku. The complexity is less. The less number of blanks it will have more number of inputs from the game side hence only fixed number of solutions can be generated. Hence it takes the longer time.

Python is giving the fastest output for the hardest input file. This is possible because Python is an interpreted language. Java and C are slow compared to Python because it uses lot of recursion and backtracking. Python on other hand does not use backtracking hence it takes less time.

Java can solve the medium and hard Sudoku faster since it will give optimize the c code faster. As the same code is used over and over again Java optimizes that same code and runs faster.

References:

python: https://github.com/diiode/hexadoku/blob/master/src/Hexadoku.py
c and java http://codereview.stackexchange.com/questions/37430/sudoku-solver-in-c
prolog: http://swish.swi-prolog.org/example/clpfd_sudoku.pl
javascript: http://www.kernel-panic.it/software/sudokiller/sudokiller.js.html