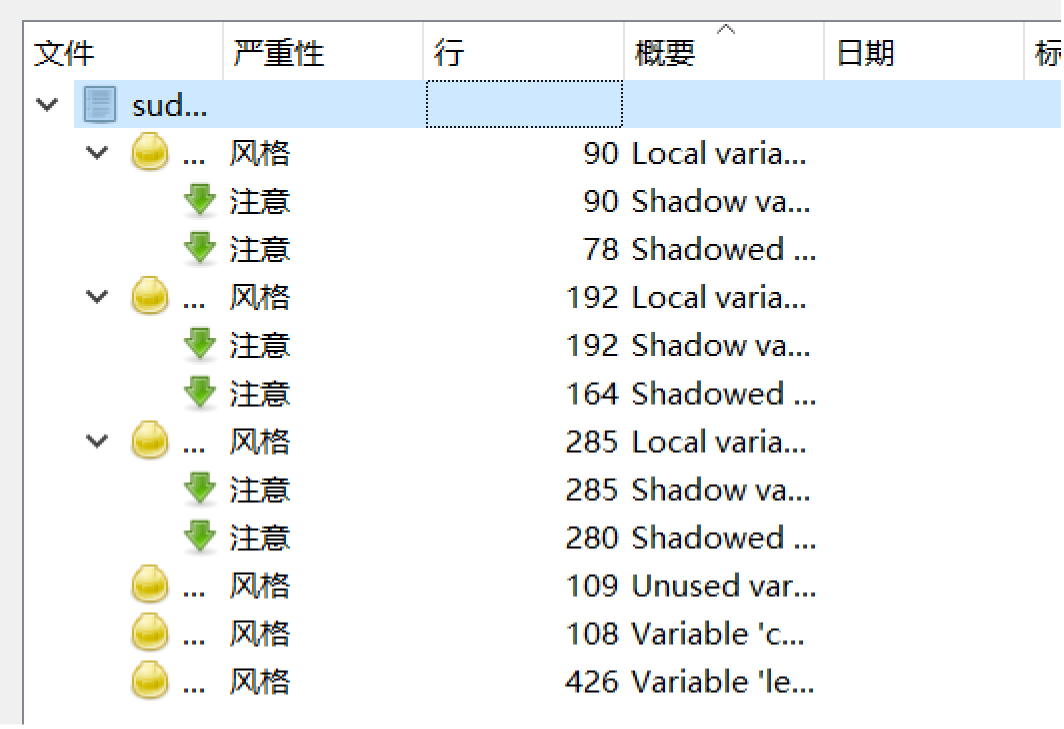
代码质量分析

静态分析

使用的工具

```
cppcheck 2.10
```

分析得到的结果如下：



文件	严重性	行	概要	日期	标
sud...					
...	风格	90	Local varia...		
	注意	90	Shadow va...		
	注意	78	Shadowed ...		
...	风格	192	Local varia...		
	注意	192	Shadow va...		
	注意	164	Shadowed ...		
...	风格	285	Local varia...		
	注意	285	Shadow va...		
	注意	280	Shadowed ...		
...	风格	109	Unused var...		
...	风格	108	Variable 'c...		
...	风格	426	Variable 'le...		

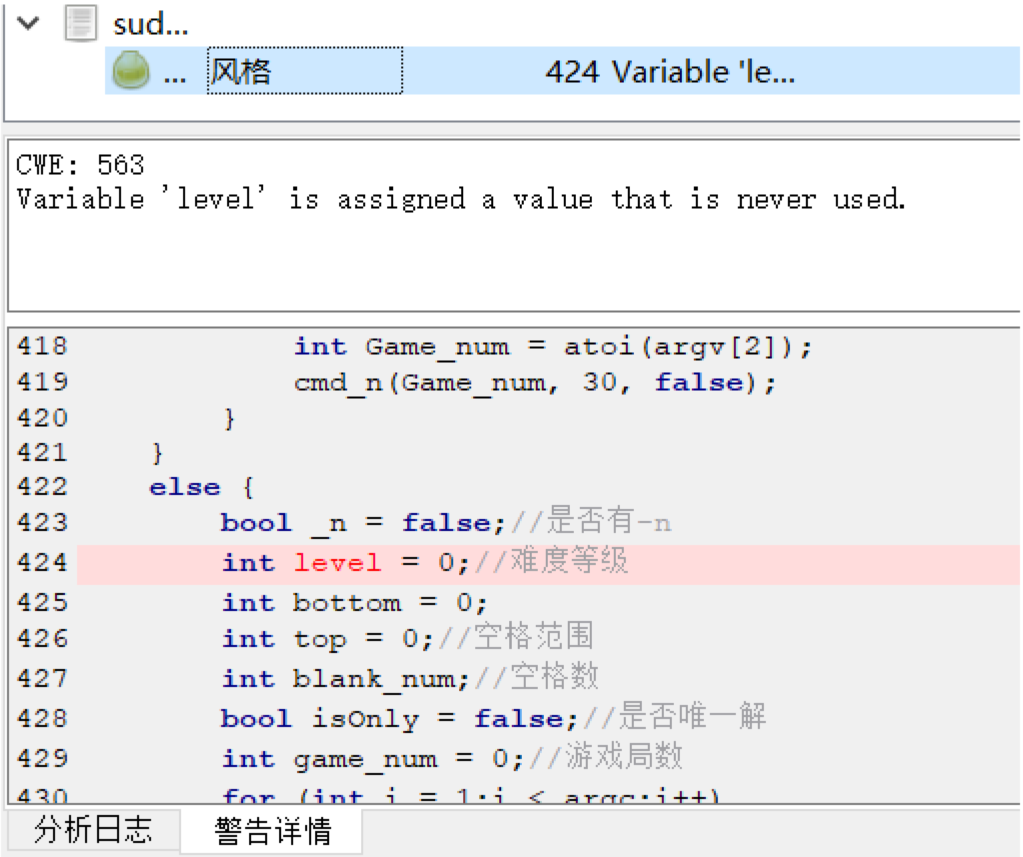
可以看到静态分析出的错误包括：局部变量与外部变量重名、未使用的变量、被赋值且未被使用的变量。

如：

```
int i = 0;
   while (i < suduku_final_num)
   {
       for (int i = 0; i < 9; i++) {
           for (int j = 0; j < 9; j++)
               outfile << Sudoku[i][j] << " ";
           outfile << endl;
       }
```

经过修改后静态分析只存在一个错误显示level变量未被使用过，但改参数实际在-m参数的功能中又被使用。由于静态分析存在一些局限性有时Cppcheck可能会产生一些误报或假阳性警告，即报告了并非真正存在问题的代码缺陷。

修改后的分析结果：

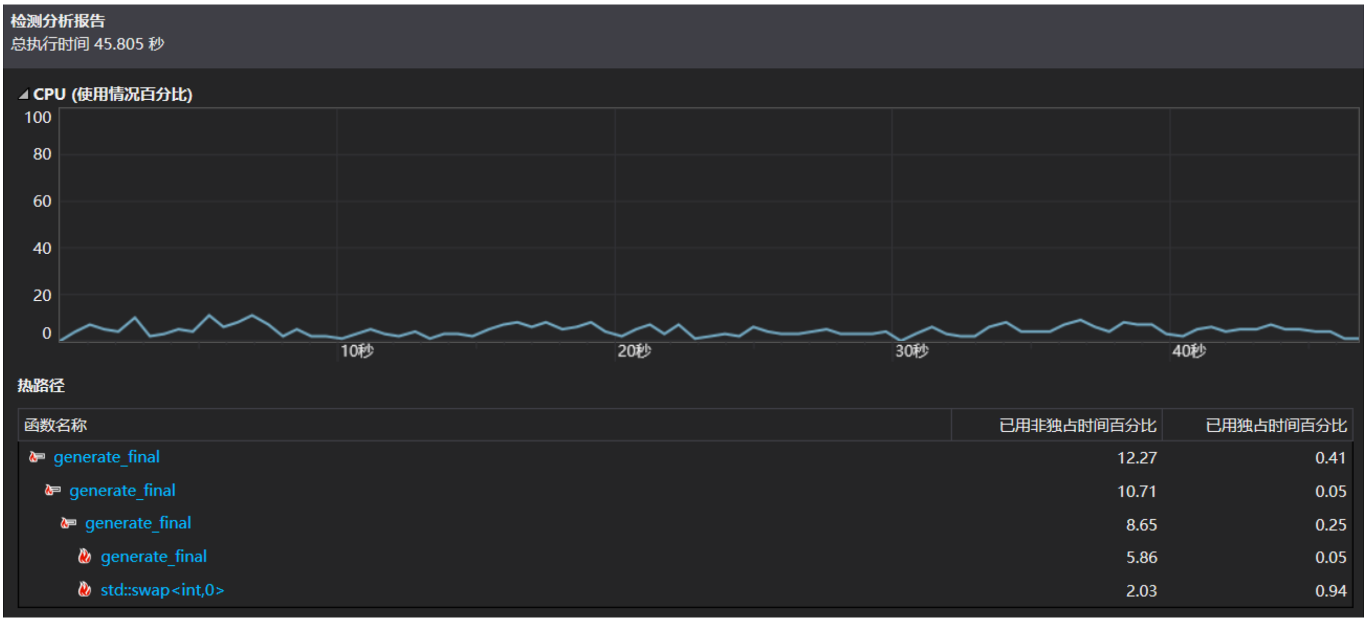


```
CWE: 563
Variable 'level' is assigned a value that is never used.
```

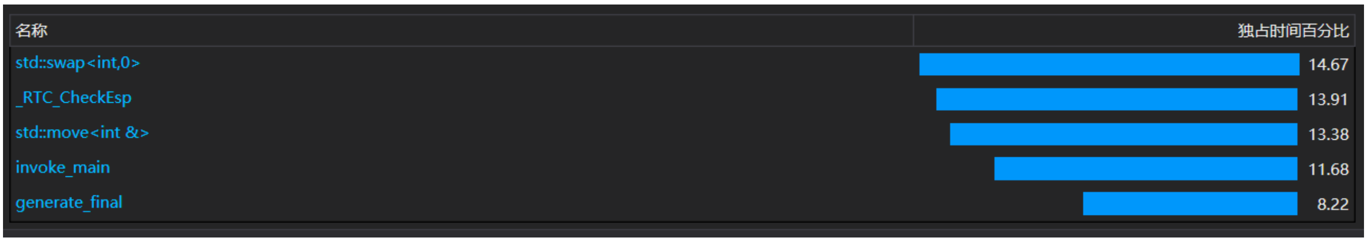
```
418             int Game_num = atoi(argv[2]);
419             cmd_n(Game_num, 30, false);
420         }
421     }
422     else {
423         bool _n = false;//是否有-n
424         int level = 0;//难度等级
425         int bottom = 0;
426         int top = 0;//空格范围
427         int blank_num;//空格数
428         bool isOnly = false;//是否唯一解
429         int game_num = 0;//游戏局数
430         for (int i = 1;i < argc;i++)
```

动态分析

使用visual studio 2019的性能嗅探器进行性能分析得到结果如下：



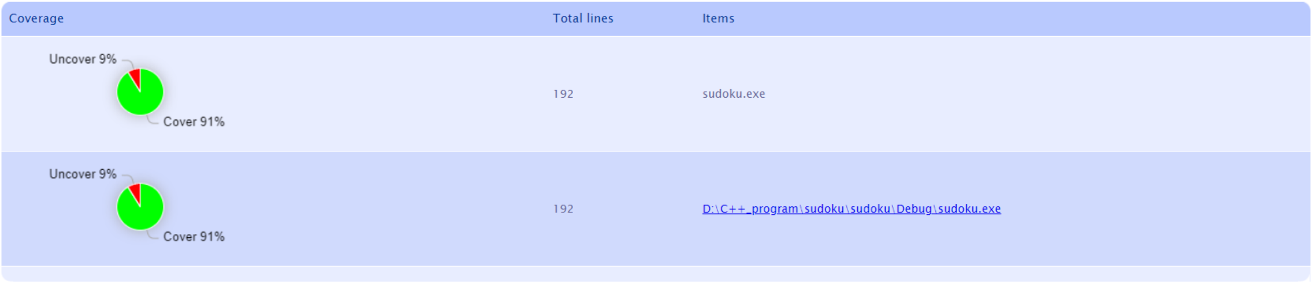
函数名称	已用非独占时间百分比	已用独占时间百分比
generate_final	12.27	0.41
generate_final	10.71	0.05
generate_final	8.65	0.25
generate_final	5.86	0.05
std::swap<int,0>	2.03	0.94



名称	独占时间百分比
std::swap<int,0>	14.67
_RTC_CheckEsp	13.91
std::move<int &>	13.38
invoke_main	11.68
generate_final	8.22

代码覆盖率

sudoku.exe



Coverage	Total lines	Items
Uncover 9% Cover 91%	192	sudoku.exe
Uncover 9% Cover 91%	192	D:\C++_program\sudoku\sudoku\Debug\sudoku.exe

测试用例

```
sudoku -c 100
sudoku -c 10000
sudoku -s Game.txt
sudoku -n 100
sudoku -n 10 -u
sudoku -n 10 -m 2
sudoku -n 10 -r 20~55
```

相关bat文件

```
@echo off

set SUDOKU_EXECUTABLE=sudoku.exe
set COVERAGE_REPORT=coverage_report.xml
set MERGED_COVERAGE_REPORT=merged_coverage_report.xml

rem Clean up previous coverage reports
del %COVERAGE_REPORT% 2>nul
del %MERGED_COVERAGE_REPORT% 2>nul

rem Test Case 1: sudoku -c 100
echo Test Case 1: Generating Sudoku Final Set (100 puzzles)...
OpenCppCoverage.exe -- %SUDOKU_EXECUTABLE% -c 100
echo.

rem Test Case 2: sudoku -c 10000
echo Test Case 2: Generating Sudoku Final Set (100 puzzles)...
OpenCppCoverage.exe -- %SUDOKU_EXECUTABLE% -c 100
echo.

rem Test Case 3: sudoku -s Game.txt
echo Test Case 3: Solving Sudoku Games (from Game.txt)...
OpenCppCoverage.exe -- %SUDOKU_EXECUTABLE% -s Game.txt
echo.
```

```
rem Test Case 4: sudoku -n 100
echo Test Case 4: Generating Sudoku Games (100 puzzles)...
OpenCppCoverage.exe -- %SUDOKU_EXECUTABLE% -n 100
echo.

rem Test Case 5: sudoku -n 10 -u
echo Test Case 5: Generating Sudoku Games (100 puzzles)...
OpenCppCoverage.exe -- %SUDOKU_EXECUTABLE% -n 10 -u
echo.

rem Test Case 6: sudoku -n 10 -m 2
echo Test Case 6: Generating Sudoku Games (10 puzzles, 2 given cells)...
OpenCppCoverage.exe -- %SUDOKU_EXECUTABLE% -n 10 -m 2
echo.

rem Test Case 7: sudoku -n 10 -r 20~55
echo Test Case 7: Generating Sudoku Games (10 puzzles, 2 given cells)...
OpenCppCoverage.exe -- %SUDOKU_EXECUTABLE% -n 10 -r 20~55
echo.

rem Merge coverage reports
echo Merging coverage reports...
OpenCppCoverage.exe --sources . --output=%MERGED_COVERAGE_REPORT%
%COVERAGE_REPORT%

echo Coverage reports merged.

pause

exit /b
```