

How to use Valgrind with TimeStorm

Introduction

The Valgrind tool suite includes a number of debugging and profiling tools that help you make your programs faster and more correct. The most popular of these tools is called Memcheck. It can detect many memory-related errors that are common in C and C++ programs and that can lead to crashes and unpredictable behavior. More information on Valgrind is available here: <http://valgrind.org/>

TimeStorm includes Valgrind plugins and you can use them to profile applications running on your local host or remote target. This document explains how to use TimeStorm to profile an application running on the remote target using memcheck.

Requirements

Application: Build your application program using the Debug build configuration.

Local host: You have to install Valgrind on the localhost only if you want to profile an application running on your local host.

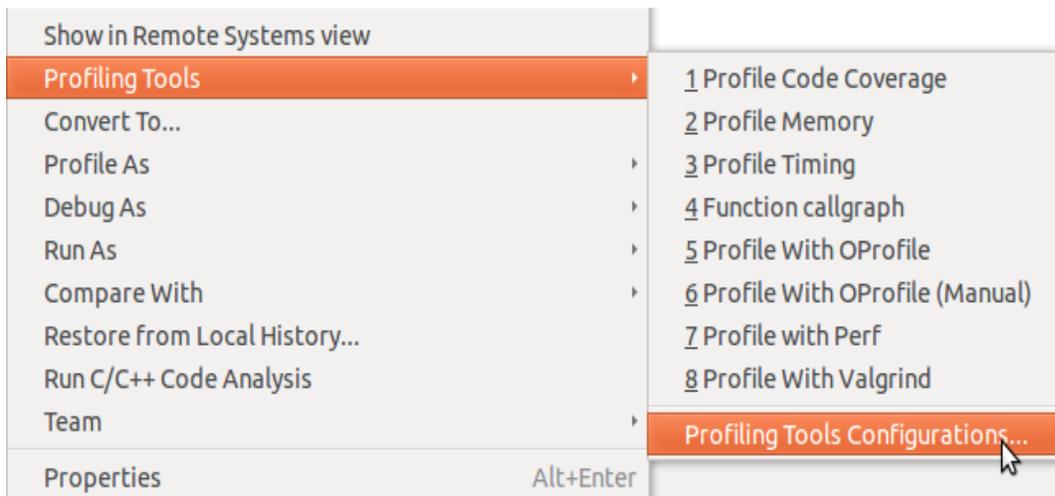
Remote Target: Include Valgrind and OpenSSH in your Target RFS. **Note:** Timesys supports Valgrind on ARMv7 (Cortex A-5, A-7, A-8, and A-9) and Intel X86/Atom processors.

Steps

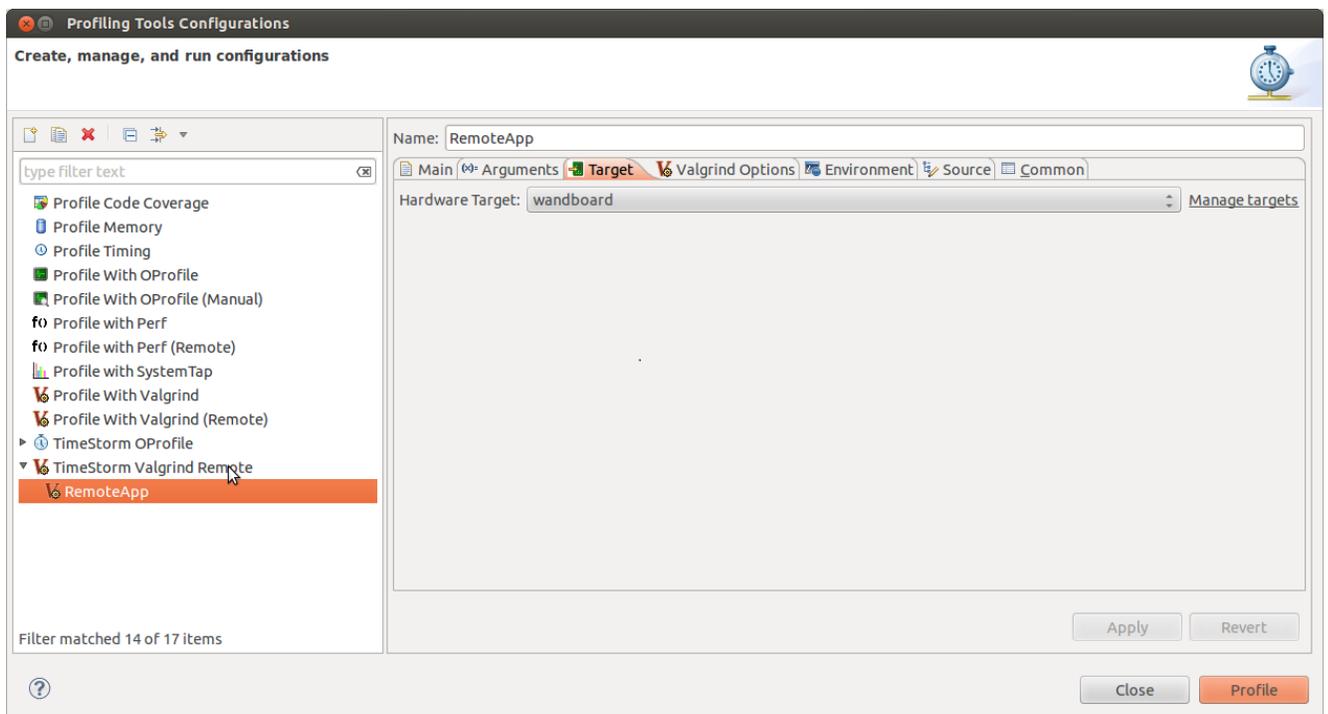
- Switch to C/C++ perspective. Compile your program with `-g` option to include debugging information. Building the program using the Debug build configuration will set the required compiler options for you. You may include the sample program in your C Project to try out Valgrind.

```
#include <stdlib.h>
void f(void)
{
    int* x = malloc(10 * sizeof(int));
    x[10] = 0;          // problem 1: heap block overrun
}                    // problem 2: memory leak -- x not freed
int main(void)
{
    f();
    return 0;
}
```

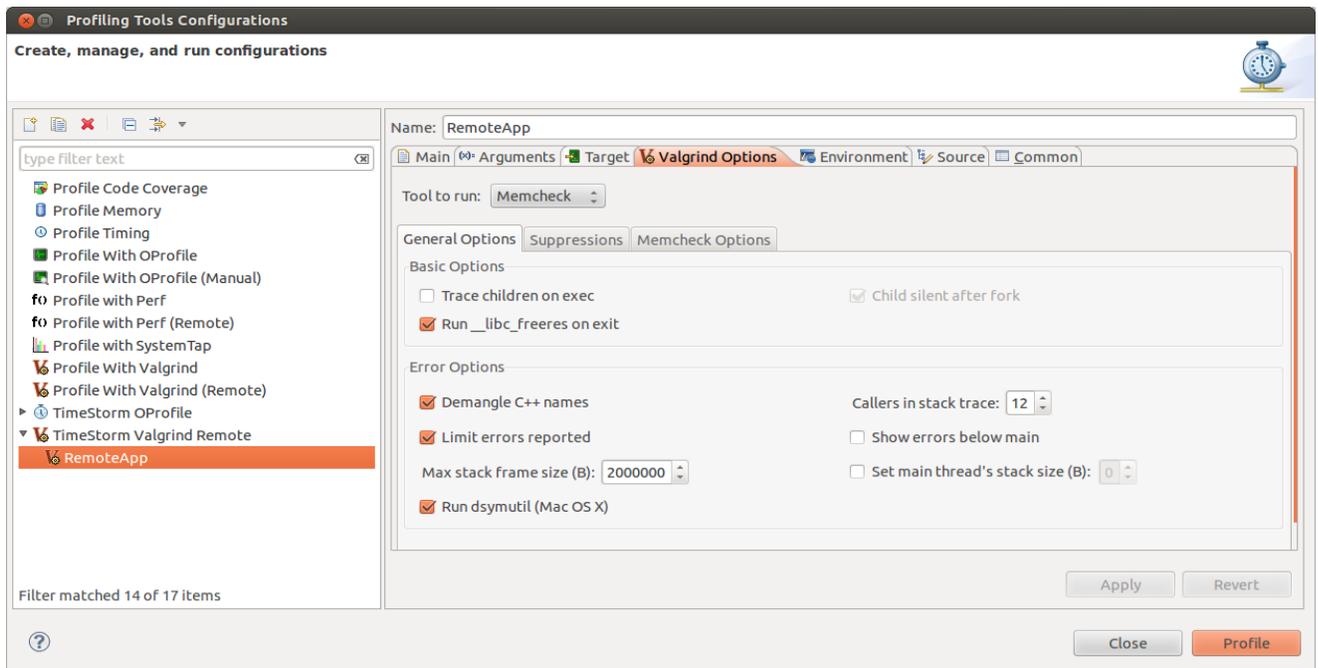
- Right click on the project and select Profiling Tools > Profiling Tools Configuration... to open the launch configuration dialog.



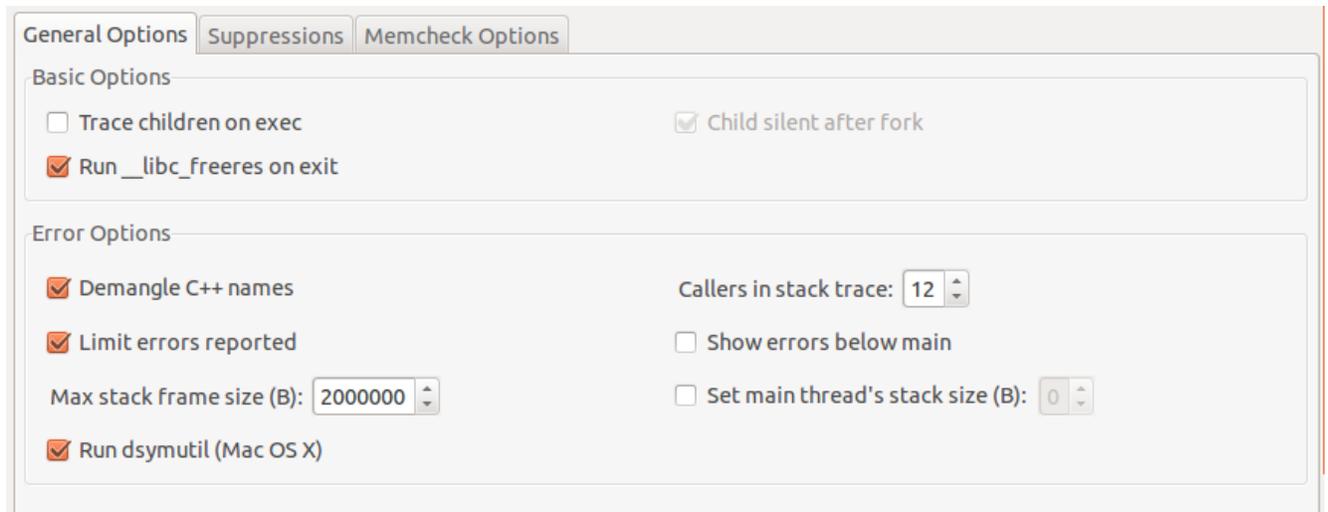
- Select 'TimeStorm Valgrind Remote', click on the 'New Launch Configuration' icon in the top left corner and enter a name for your profile launch configuration.

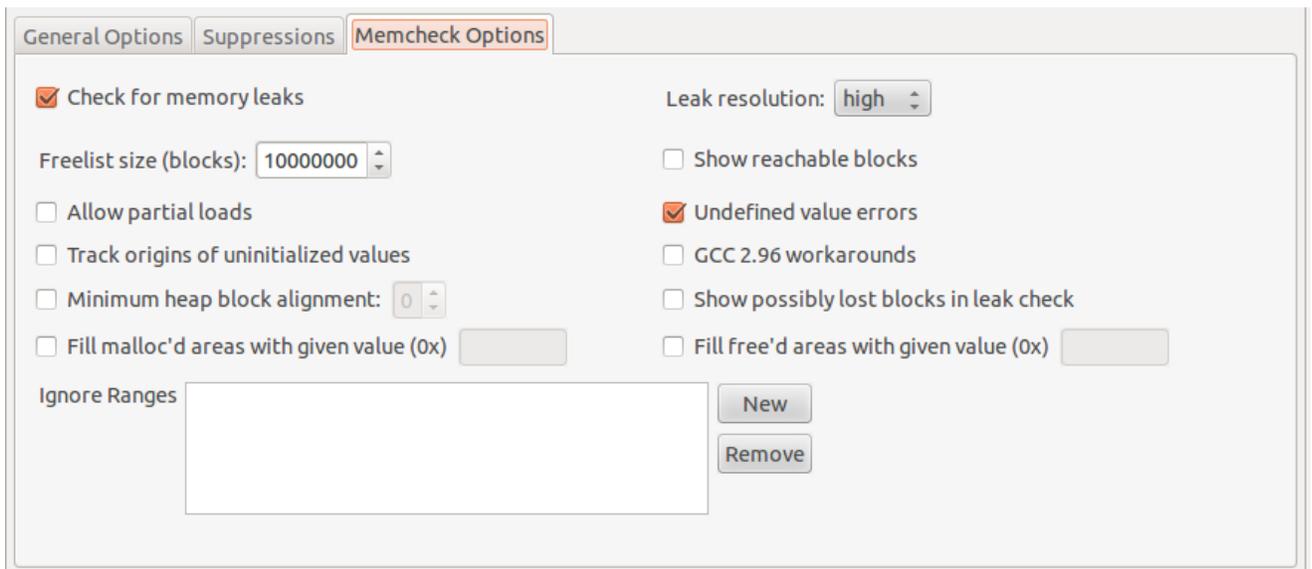
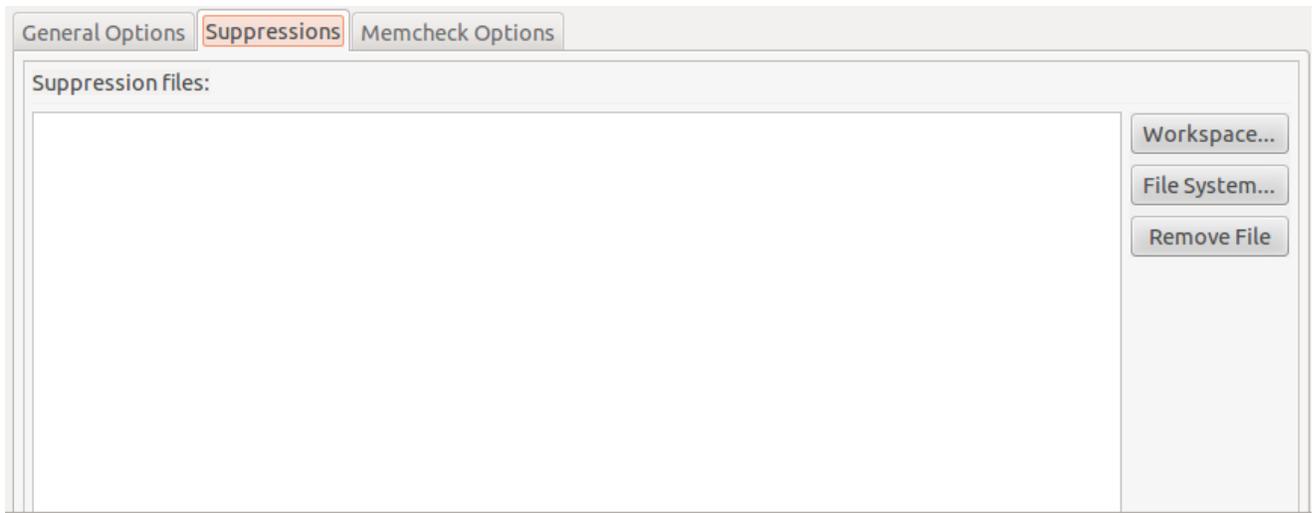


- Select a target on the Target tab.
- Select memcheck on the Valgrind options tab.

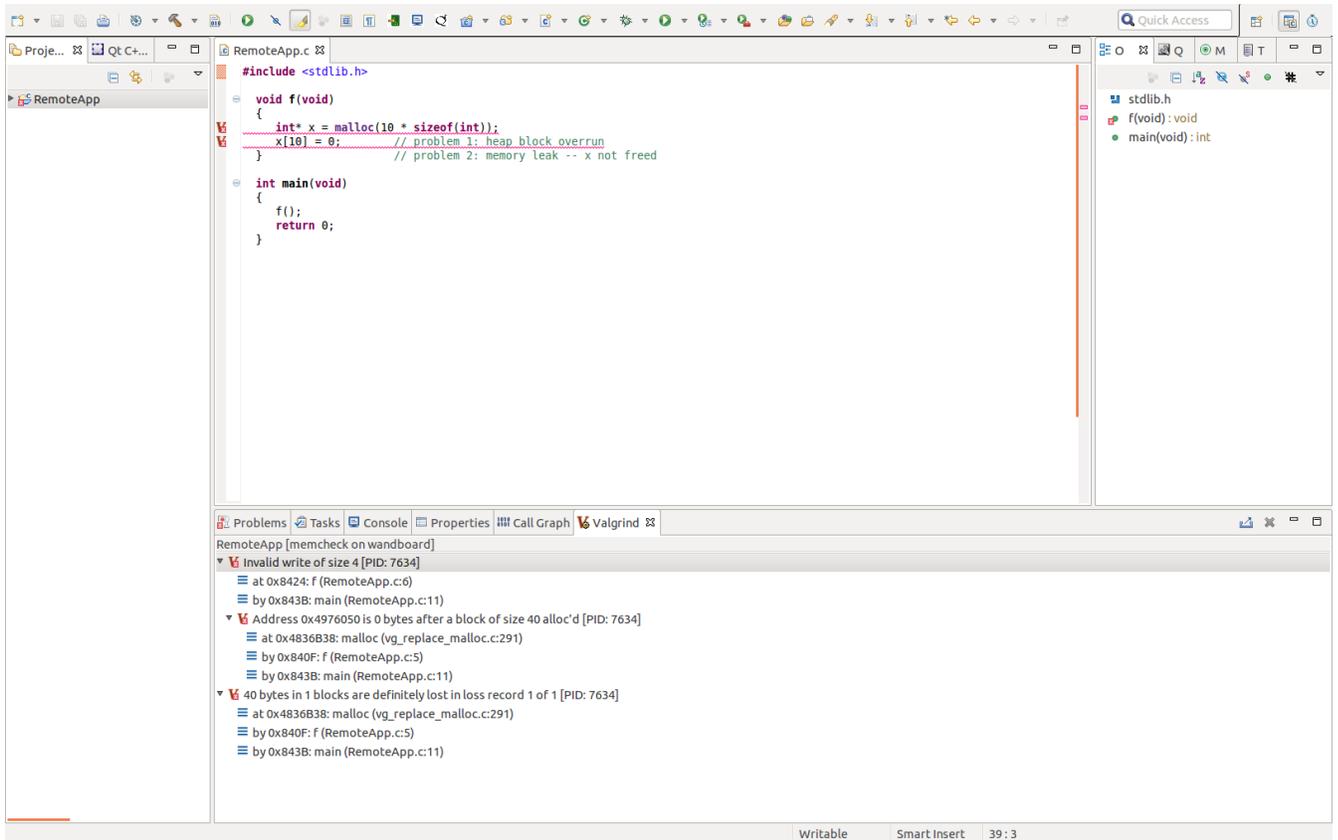


- You can tweak Valgrind by changing the settings on the General Options, Suppressions and Memcheck Options tabs. The default settings should work in most cases. You may refer to <http://valgrind.org/docs/manual/mc-manual.html> for information on memcheck.

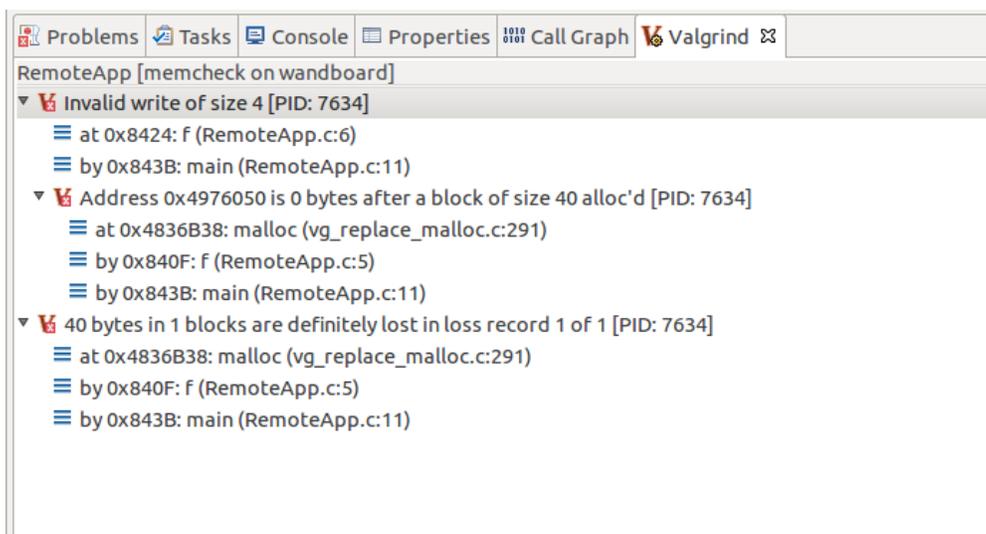




- Click the Profile button to start profiling.
- TimeStorm will run memcheck on the target, upload memcheck's output to the host and display the results in the Valgrind View in the bottom of the C/C++ perspective and display the Valgrind error markers in the source code.



- Valgrind view displays the memcheck results in a tree view. The tree can be expanded to view the stack trace. The results include the stack trace, source file with the line number and the memory address. Double clicking on a line will open the corresponding source file in the source editor with error markers on the lines that have the memory leak.



```
RemoteApp.c 23
#include <stdlib.h>

void f(void)
{
    int* x = malloc(10 * sizeof(int));
    x[10] = 0; // problem 1: heap block overrun
              // problem 2: memory leak -- x not freed
}

int main(void)
{
    f();
    return 0;
}
```