PH-SFT Group

Application performance testing for CernVM

KARBON Application Profiler

Version 0.4 beta

Author:
Ioannis Charalampidis

Supervisor:
Predrag Buncic

July 3, 2011

Contents

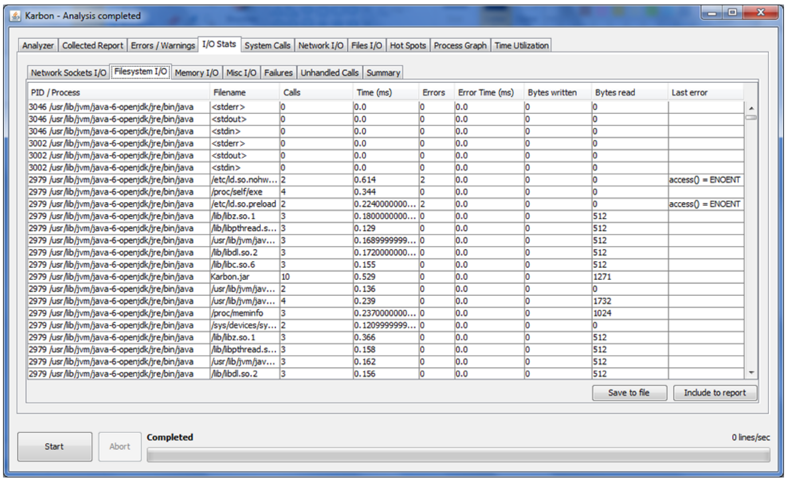
1 Overview

KARBON is a platform-independent visual application profiling tool written in Java, based on system call tracing. It processes trace files generated by strace, systemTAP or similar call tracing applications and displays the application performance results in a user-friendly graphical interface (Figure 1.1). It is capable of tracking network I/O, disk I/O and core OS operations, such as process management and signalling. Keep in mind that this application can not track memory access since it is not implemented through system calls. However memory access implemented using the mmap() is tracked.

KARBON has a modular design and a simple API that allows anyone to extend it's functionality. With the currently developed extensions, KARBON is capable of performing the following tasks:

- Collect statistics regarding system calls, such as: number of hits on each system call, mostly used system calls, failures, time penalties, "hot" spots etc.
- Track the evolution of file descriptors though time and group accordingly all the relevant system calls.
- Map the file descriptors to file names, network sockets or memory mappings and display the usage of the relevant system calls.
- Track the lifespan of child processes on multi-threaded applications and generate relation graphs.
- Display collective information that help detect bottlenecks and time-consuming operations.

After the trace file is analysed, it is possible to generate custom reports that can be exported for further processing or presentation.



PID / Process	Filename	Calls	Time (ms)	Errors	Error Time (ms)	Bytes written	Bytes read	Last error
3046 /usr/lib/jvm/java-6-openjdk/jre/bin/java	<stderr>	0	0.0	0	0.0	0	0	
3046 /usr/lib/jvm/java-6-openjdk/jre/bin/java	<stdout>	0	0.0	0	0.0	0	0	
3046 /usr/lib/jvm/java-6-openjdk/jre/bin/java	<stdin>	0	0.0	0	0.0	0	0	
3002 /usr/lib/jvm/java-6-openjdk/jre/bin/java	<stderr>	0	0.0	0	0.0	0	0	
3002 /usr/lib/jvm/java-6-openjdk/jre/bin/java	<stdout>	0	0.0	0	0.0	0	0	
3002 /usr/lib/jvm/java-6-openjdk/jre/bin/java	<stdin>	0	0.0	0	0.0	0	0	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/etc/ld.so.nohw...	2	0.614	2	0.0	0	0	access() = ENOENT
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/proc/self/exe	4	0.344	0	0.0	0	0	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/etc/ld.so.preload	2	0.2240000000...	2	0.0	0	0	access() = ENOENT
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/lib/libz.so.1	3	0.1800000000...	0	0.0	0	512	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/lib/libpthread.s...	3	0.129	0	0.0	0	512	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/usr/lib/jvm/jav...	3	0.1689999999...	0	0.0	0	512	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/lib/libdl.so.2	3	0.1720000000...	0	0.0	0	512	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/lib/libc.so.6	3	0.155	0	0.0	0	512	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	Karbon.jar	10	0.529	0	0.0	0	1271	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/usr/lib/jvm/jav...	2	0.136	0	0.0	0	0	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/usr/lib/jvm/jav...	4	0.239	0	0.0	0	1732	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/proc/meminfo	3	0.2370000000...	0	0.0	0	1024	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/sys/devices/sy...	2	0.1209999999...	0	0.0	0	0	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/lib/libz.so.1	3	0.366	0	0.0	0	512	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/lib/libpthread.s...	3	0.158	0	0.0	0	512	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/usr/lib/jvm/jav...	3	0.162	0	0.0	0	512	
2979 /usr/lib/jvm/java-6-openjdk/jre/bin/java	/lib/libdl.so.2	3	0.156	0	0.0	0	512	

Completed

0 lines/sec

Figure 1.1: The KARBON User interface after the trace file analysis

2 Getting Started

2.1 Requirements

In order to run KARBON you need to have Java Runtime 6.0 or later installed on your machine. You can get the latest available version from the official website[1].

You also need a system call tracing application in order to generate a trace file. It's recommended to use the latest version of strace[2].

Additionally, if you want to perform a real-time analysis from a remote machine you need a socket client that reads from STDIN such as netcat.

2.2 Usage

There are mainly two ways to use KARBON :

1. Trace the application independantly, generate the trace file and then pass it to KARBON for analysis.
2. Run an application and perform real-time analysis with KARBON *(Experimental)*.

2.2.1 Example of separated tracing/analysis setup

In this case, you have to generate the trace log manually and then pass the output file to KARBON for analysis. To do so, you need to do the following:

1. Install strace on the machine you want to perform the trace onto.
2. Run your application under tracer using the following command:

   ```
   strace -Ttttf -s 128 -o <tracefile> -- <application> <arguments>
   ```

3. Copy the trace file to the machine that will run KARBON
4. Start KARBON by double-clicking the .jar file, using the start.sh or start.bat, or by invoking the following command:

   ```
   java -jar Karbon.jar
   ```

[1] You can find the latest Java releases here: http://www.oracle.com/technetwork/java/javase/downloads/index.html

[2] You can get the latest version of STrace from here: http://sourceforge.net/projects/strace/

5. On the Analyzer tab select the **From file** tab
6. Click **Browse** and select the trace file
7. Load the required plug-ins by checking the check-boxes in the list or click **All** to load them all. *(Keep in mind that there are dependencies between plugins and some of them will be loaded automatically even if you don't select them.)*
8. Click **Start**

2.2.2 Example of real-time analysis setup

KARBON supports TCP and UDP streaming input, allowing it to receive real-time logs from a remote machine without a need of an intermediate log file. To do so, you need to do the following:

1. Install `strace` and `netcat` on the machine you want to perform the trace onto.
2. Start KARBON by double-clicking the .jar file, using the start.sh or start.bat, or by invoking the following command:

   ```
   java -jar Karbon.jar
   ```
3. Select the **From TCP Stream** tab. You can change the port number but it is recommended to leave the port number to 1225.
4. Load the required plug-ins by checking the check-boxes in the list or click **All** to load them all. *(Keep in mind that there are dependencies between plugins and some of them will be loaded automatically even if you don't select them.)*
5. Click **Start**
6. Go to the testing machine and run your application under tracer using the following command:

   ```
   strace -Ttttf -s 128 -o <tracefile> -- <application> <arguments> | nc
   <karbon host> 1225
   ```
7. When the application finishes. Click the **Abort** button to finalize the trace.