

BASEMARK CL USER GUIDE

Rauli Laatikainen

Rightware 2011

Table of Contents

Running executable	2
Available versions	2
Requirements	2
Run rules	3
Configuring with settings files	3
Command line parameters.....	3
Using binary kernels.....	3
Running the Android build	4
Report data	4
Main score	4
Test scores	4
Building the benchmark from source	5
Windows.....	5
Linux.....	6
Android.....	7

Running executable

Available versions

We provide the Basemark CL for the following platforms and configurations:

- Windows 32 bit, OpenGL 2.1
- Linux 64 bit, OpenGL 2.1
- Android, OpenGL ES2.0

Requirements

In order to execute the benchmark the platform must have drivers for OpenCL 1.1 and the OpenGL version required by the platform (shown in list above).

The embedded profile requires around 80 megabytes of memory. Full version (aimed for desktop use) uses 275 megabytes. The memory allocation can be controlled from the configuration file.

Run rules

The run rules are specified in the Basemark CL Specification. The run rules must be followed to receive valid benchmark score.

Configuring with settings files

Basemark CL settings can be configured from the “application.cfg” settings file. The settings file includes comments which describe the purpose of each setting. The file can be modified with any regular text editor.

Command line parameters

Basemark CL can be customized to launch other executable before and after each frame. This feature is implemented to help benchmark users to include their own logging and timing code into the benchmark loop.

This happens with the following two command line parameters.

```
--on-measure-start="c:\start.bat"
```

```
--on-measure-stop="c:\stop.bat"
```

The tests which are launched automatically can be provided as parameter in a comma separated list. This option overrides settings in “autotest” section of the “application.cfg” file.

```
--tests="TestSoftBodySimulation, TestFluidSimulation"
```

The following option enables repeating the given tests infinitely.

```
--repeat=1
```

Test names are listed below:

TestSoftBodySimulation, TestFluidSimulation, TestSphFluid, TestWaveSimulation, TestMandelbulb, TestJuliaFractal, TestOnlineCompiler, TestImageSmoothing, TestImageColorCorrection, TestImageNoiseReduction, TestImageSurfaceAwareSmoothing, TestImageSharpening, TestVideoSmoothing, TestVideoColorCorrection, TestVideoNoiseReduction, TestVideoSurfaceSmoothing, TestVideoSharpening

Using binary kernels

The benchmark can be run using binary kernels. To use binary kernels instead of regular kernels place the binary kernel in the data folder next to .cl file with extension .bin. The benchmark will then automatically use the binary kernel file placed in the data directory.

For example to use binary kernel for “julia.cl” place the binary kernel file in the data folder with file name “julia.cl.bin”.

Running the Android build

Android build seeks the data files from path “/sdcard/rightware/kanzi/basemarkcl”. Install the files to that folder before running.

The phone should also have folder and write permissions to “/sdcard/rightware/kanzi/basemarkcl/reports”-folder.

Report data

Basemark CL outputs XML-report after a benchmark run. On Windows and Linux builds the report is output to reports folder under executable.

The XML-report document generated by benchmark can be opened in web browser or it can be read directly in text editor.

Main score

The main score of the benchmark is shown at the top of the document. The benchmark score describes the time spent on operations. So the lower the score the better it is. Notice the FULL_PROFILE or EMBEDDED_PROFILE text next to score. The scores between these profiles are not comparable.



Score

Benchmark score: 71.326248 for FULL_PROFILE

Below the main score scores the scores of each test category are listed. Also there is general run information which shows the information about version and run parameters.

Test scores

The table which describes benchmark scores contains the following values

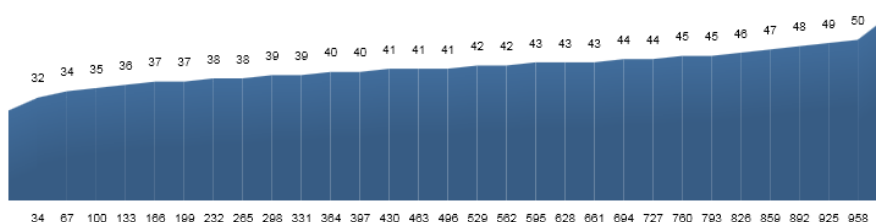
Slowest	Slowest frame time of all frames (milliseconds)
Fastest	Fastest frame time of all frames (milliseconds)
Average	Average duration of frames (milliseconds)
Median	Median of frame times
Geometric mean	Geometric mean of frame times
Frame times	Each frames time in comma separated list

The graph below shows a set of frame times as a graph presentation. The graph is useful for getting overview how the frame times behave on different parts of the test.

Test scores

1. Smoothed particle hydrodynamics test

Name	Value
slowest	104
fastest	28
average	41.563000
median	42.000000
geometricMean	41.205757
frameTimes	28, 28, 28, 29, 29, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 31, 31, 31, 31, 31, 31, 31, 31, more .



Sample of the test scores

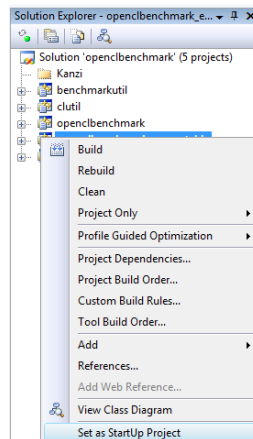
Building the benchmark from source

Here are brief instructions on building the benchmark from source code. Please note that if you license only the benchmark binaries you will not receive the source code.

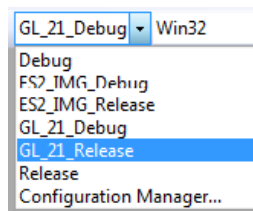
Windows

Use Visual Studio 2008 or newer to build.

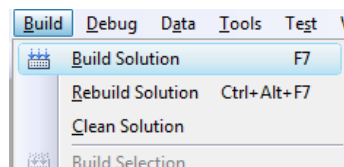
- Unzip the source code to your hard drive
- Open the Visual Studio solution file from the path
"Benchmark\BasemarkCL\application\configs\platforms\win32\basemarkcl.sln" in Visual Studio
- Set "basemarkcl_executable" as startup project by right clicking the item in Solution Explorer



- Set used configuration to “GL_21_Release” to compile for OpenGL 2.1



- Select build, rebuild solution



Run the executable with the data files provided. The executable expects to find “binaries.cfg” and “settings.cfg” -files from the same folder and data and report folders from the same level with the executable. Easiest way to make sure you have correct file and folder structure is to unzip the windows binary package and overwrite the old executable.

The NO_GL profiles build command line version which can be used to benchmark the OpenCL performance without any OpenGL interoperability. When using that version remember to configure the autorun feature to run test automatically from application.cfg file.

Linux

In order to build for Linux you need to have SCons installed.

To build the benchmark, first unpack the sources and then run the command:

```
scons BasemarkCL <debug | release> <GL21 | noES>
```

In the directory: Benchmarks/BasemarkCL/application/configs/platforms/linux_x64

Use either option from within the brackets, e.g. release gl21: “scons BasemarkCL release GL21”.

The benchmark uses these Open Source libraries:

- freetype

- libjpeg
- libpng
- zlib
- avcodec, swscale, avutil and avformat from ffmpeg

If your distribution does not have all of the libraries the sources can be found

Kanzi/Engine/libraries/common for all the required libraries except FFmpeg. FFmpeg can be found on Benchmark/ BasemarkCL /application/libraries/common/ffmpeg/

Android

Note: *Currently shipped version will compile for Android with empty OpenCL calls instead of the normal ones. That is because there are no OpenCL libraries for Android currently. Later when building for OpenCL supported Android SDK set "clu_opencl_base.h" -files define as follows* `#define CLU_OPENCL_VERSION CLU_OPENCL_1_1`

Note: *We will port the benchmark to run properly Android whenever we receive a platform where we can run it.*

To build the current version for Android you need to have:

- Android SDK
- Android NDK (rev 5)
- SCons

Before running scons:

- You should set NDK_PATH variable to point to your NDK- root-directory
- Run android SDK update in the folder "source/Benchmark/ BasemarkCL /application/configs/platforms/android". The command for updating is "android update project -p . -t android-8".

Building the APK file can be done by executing command: "scons BasemarkCL release es2" in folder "Benchmark/ BasemarkCL /application/configs/platforms/android".

To run Android version check running instructions for Android from above.

Note: *As we can't currently test the Android build we are prepared to do fixes and testing on it when we get Android device with OpenCL support.*