Rockchip Linux Benchmark KPI

ID: RK-CS-YF-375

Release Version: V1.0.0

Release Date: 2020-08-05

Security Level: □Top-Secret □Secret □Internal ■Public

DISCLAIMER

THIS DOCUMENT IS PROVIDED “AS IS”. ROCKCHIP ELECTRONICS CO., LTD.(“ROCKCHIP”)DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS,MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip’s registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document provides an overview of some data references for benchmark. Benchmark is a method of testing code performance and can also be used to identify CPU or memory
efficiency problems of a piece of code. aiming to help engineers use benchmarks to test
different concurrency patterns or use benchmarks to assist in configuring the number of work pools to maximize system throughput.

Intended Audience

This document is mainly intended for:

Technical support engineers

Software development engineers

Support benchmarks

name	test items	summary
glmark2	gpu	OpenGL 2.0 and ES 2.0 benchmark
unixBench	cpu/mem/io/system	oveall performance test
lmbench	cpu/mem/io/bandwidth/latency	test cpu/mem/io bandwidth and latency

Revision History

Date	Version	Author	Revision History
2020-08-05	V1.0.0	Caesar Wang	Initial version

Contents

1. Glmark2

glmark2 is an OpenGL 2.0 and ES 2.0 benchmark.

The scores of Glmark2's fullscreen and offscreen mode of each chip are shown in the table below:

Item	RK3399/RK3399Pro	RK3288	RK3326/PX30
Full Screen	56	37	30
Off Screen	823	544	301
GPU Type	Mali-T864	Mali-T764	Mali-G31MP2

The resolution of test screen on different chip platforms is as follows:

Item	RK3399/RK3399Pro	RK3288	RK3326/PX30
Screen resolution	EDP:1536x2048p60	EDP:1536x2048p60	HDMI:1920x1080p60

The performance of fullscreen mode is related to the screen, so the actual performance can refer to the score of off screen mode.

2. UnixBench

The purpose of UnixBench is to provide a basic indicator of the performance of a Unix-like system; hence, multiple tests are used to test various aspects of the system's performance.
This is designed to allow you to assess:

- the performance of your system when running a single task
- the performance of your system when running multiple tasks
- the gain from your system's implementation of parallel processing

UnixBench consists of a number of individual tests that are targeted at specific areas. Here is a summary of what each test does:

- Dhrystone

Developed by Reinhold Weicker in 1984. This benchmark is used to measure and compare the performance of computers. The test focuses on string handling, as there are no floating point operations. It is heavily influenced by hardware and software design, compiler and linker options, code optimization, cache memory, wait states, and integer data types.

- Whetstone

This test measures the speed and efficiency of floating-point operations. This test contains several modules that are meant to represent a mix of operations typically performed in scientific applications. A wide variety of C functions including sin, cos, sqrt, exp, and log are used as well as integer and floating-point math operations, array accesses, conditional branches, and procedure calls. This test measure both integer and floating-point arithmetic.

- execl Throughput

This test measures the number of execl calls that can be performed per second. execl is part of the exec family of functions that replaces the current process image with a new process image. It and many other similar commands are front ends for the function execve().

- File Copy

This measures the rate at which data can be transferred from one file to another, using various buffer sizes. The file read, write and copy tests capture the number of characters that can be written, read and copied in a specified time (default is 10 seconds).

- Pipe Throughput

A pipe is the simplest form of communication between processes. Pipe throughput is the number of times (per second) a process can write 512 bytes to a pipe and read them back. The pipe throughput test has no real counterpart in real-world programming.

- Pipe-based Context Switching

This test measures the number of times two processes can exchange an increasing integer through a pipe. The pipe-based context switching test is more like a real-world application. The test program spawns a child process with which it carries on a bi-directional pipe conversation.

- Process Creation

This test measure the number of times a process can fork and reap a child that immediately exits. Process creation refers to actually creating process control blocks and memory allocations for new processes, so this applies directly to memory bandwidth. Typically, this benchmark would be used to compare various implementations of operating system process creation calls.

- Shell Scripts

The shells scripts test measures the number of times per minute a process can start and reap a set of one, two, four and eight concurrent copies of a shell scripts where the shell script applies a series of transformation to a data file.

- System Call Overhead

This estimates the cost of entering and leaving the operating system kernel, i.e., the overhead for performing a system call. It consists of a simple program repeatedly calling the getpid (which returns the process id of the calling process) system call. The time to execute such calls is used to estimate the cost of entering and exiting the kernel.

- Graphical Tests

Both 2D and 3D graphical tests are provided; at the moment, the 3D suite in particular is very limited, consisting of the ubgears program. These tests are intended to provide a very rough idea of the system's 2D and 3D graphics performance. Bear in mind, of course, that the reported performance will depend not only on hardware, but on whether your system has appropriate drivers for it.

The reference scores of test items for each chip are as follows:

- the performance of your system when running a single task

Item	RK3399/RK3399Pro	RK3288	RK3326/PX30
Dhrystone 2 using register variables	19191210.4	10626086.6	5704897.1
Double-Precision Whetstone	3303.5	1718.9	1565.2
Execl Throughput	2730.8	1538.1	787.9
File Copy 1024 bufsize 2000 maxblocks	263262.3	163001.5	125333.2
File Copy 256 bufsize 500 maxblocks	98335.8	50635.1	37871.9
File Copy 4096 bufsize 8000 maxblocks	677993.2	384632.9	321189.7
Pipe Throughput	775302.3	357578.5	300305.5
Pipe-based Context Switching	87345.3	54247.5	37434.5
Process Creation	4274.2	3512.1	2086.0
Shell Scripts (1 concurrent)	2944.0	2973.3	1474.2
Shell Scripts (8 concurrent)	832.4	703.2	431.7
System Call Overhead	721899.8	624614.1	568868.6
System Benchmarks Index Score	654.7	421.7	290.6

- the performance of your system when running multiple tasks

Item	RK3399/RK3399Pro	RK3288	RK3326/PX30
Dhrystone 2 using register variables	61892645.4	41527276.3	22821903.2
Double-Precision Whetstone	13192.2	6870.1	6265.7
Execl Throughput	6638.9	4127.0	2449.4
File Copy 1024 bufsize 2000 maxblocks	253903.6	265838.2	194293.5
File Copy 256 bufsize 500 maxblocks	74647.0	74156.4	54107.4
File Copy 4096 bufsize 8000 maxblocks	715699.5	709343.4	565091.8
Pipe Throughput	3159789.0	1323176.0	1191104.7
Pipe-based Context Switching	298324.6	134686.9	154652.2
Process Creation	11834.7	7412.8	5183.1
Shell Scripts (1 concurrent)	7420.0	5710.9	3587.9
Shell Scripts (8 concurrent)	952.5	744.6	477.4
System Call Overhead	2514699.7	2392234.7	2206337.3
System Benchmarks Index Score	1402.8	989.5	746.4

3. LMbench

lmbench is a suite of simple, portable, ANSI/C microbenchmarks for UNIX/POSIX. In general, it measures two key features: latency and bandwidth.

Things need to focus on LMbench include the following:

- Latency benchmarks
  - Context switching
  - Networking: connection establishment, pipe, TCP, UDP, and RPC hot potato
  - File system creates and deletes
  - Process Creation
  - Signal handling
  - System call overhead
  - Memory read latency
- Bandwidth benchmarks.
  - Cached file read
  - Memory copy (bcopy)
  - Memory read
  - Memory write
  - Pipe
  - TCP
- Miscellanious
  - Processor clock rate calculation
- the performance of your system when running latency benchmarks. (microseconds)

Item	RK3399/RK3399Pro	RK3288	RK3326/PX30
Simple syscall	0.2228	0.2126	0.2176
Simple read	0.3385	0.5989	0.6965
Simple write	0.2828	0.3777	0.5637
Simple stat	1.3041	2.9836	4.1538
Simple fstat	0.3456	0.6527	0.6581
Simple open/close	3.0633	5.2514	7.7892
Signal handler installation	0.4103	0.6593	0.6882
Signal handler overhead	1.9096	4.3894	4.5098
Pipe latency	14.3626	23.8091	36.1158
AF_UNIX sock stream latency	19.0106	21.9593	50.0182
Process fork+exit	254.8182	390.5714	668.1250
Process fork+execve	281.0526	417.2308	754.7143
UDP latency	28.8465	53.3952	64.4354
TCP latency	36.1384	65.4973	77.5462
STREAM2 sum latency	1.89 nanoseconds	3.35 nanoseconds	5.96 nanoseconds

- the performance of your system when running bandwidth benchmarks. (MB/sec)

Item	RK3399/RK3399Pro	RK3288	RK3326/PX30
AF_UNIX sock stream bandwidth	3751.28	2545.44	1181.90
Pipe bandwidth	1390.20	804.06	805.99
STREAM2 sum bandwidth	4237.37	2385.57	1342.32