

修订说明

- CPU PhotoWorxx: CPU 的整数运算能力、多个 CPU 核心运算能力, 通过模拟数位影像的处理过程, 来对 CPU 处理速度进行评估。
- CPU ZLib: 通过计算 CPU 在处理压缩和解压缩文件时的时间, 来检测 CPU 在处理文件方面的速度。
- CPU AES: 通过此项测试, 反映 CPU 在进行 AES 加密演算法时的效能。

利用c++程序进行测试

- CPU PhotoWorxx:
 - CPU 的整数、浮点数运算能力测试, 利用Pthread进行多线程测试
 - 多个 CPU 核心运算能力: 利用SSE/AVX/NEON工具集编写并行代码, 测试多核心运算能力
- CPU ZLib: 通过计算 CPU 在处理压缩和解压缩文件时的时间, 来检测 CPU 在处理文件方面的速度: 编写c++随机处理压缩和解压缩文件程序进行测试
- CPU AES: 通过此项测试, 反映 CPU 在进行 AES 加密演算法时的效能: 编写c++随机 AES 加密演算法进行测试

测量指令性能:

1. 选择一个测试框架: 选择一个C++测试框架, 例如Google Test或Catch2, 来方便地编写和运行测试。
2. 选择一个性能测试库: 选择一个C++性能测试库, 例如Google Benchmark或Celer, 来测量函数或代码段的性能。
3. 编写测试用例: 使用测试框架编写测试用例, 为要测量的函数编写单元测试。
4. 编写性能测试: 使用性能测试库编写性能测试, 可以使用微秒计时器 (如 `std::chrono::high_resolution_clock`) 测量执行时间, 并根据需要重复多次执行测试以获得更准确的结果。
5. 运行测试: 使用测试框架和性能测试库运行测试并查看结果。可以将结果输出到控制台、文件或图表中。

利用 Python 实时检测 CPU 和 GPU 的功率, 并绘制曲线图。

要实时检测CPU和GPU的功率并绘制曲线图, 可以按照以下步骤进行:

1. 安装所需的Python库: 需要安装pandas、matplotlib、psutil和py3nvmI库。
2. 实时获取CPU和GPU功率数据: 可以使用psutil库获取CPU功率数据, 使用py3nvmI库获取GPU功率数据。
3. 将获取到的功率数据保存到CSV文件中: 可以使用pandas库将数据保存到CSV文件中。
4. 实时绘制功率曲线图: 可以使用matplotlib库实时绘制功率曲线图, 并实时更新曲线。