Performance\_counter.sh

#! /bin/sh

# $1: Complete execution command $2: store folder for generated file

# eg: ./performance\_counter.sh "./hackbench -s 512 -l 200 -g 15 -f 25 -P" /home

if [ $# -ne 2 ]; then

echo "Usage: ./performance\_counter.sh parameter1 parameter2"

exit 1

fi

echo "parameter1=$1"

result=$(echo "$1" | sed 's:.\*/::')

file\_name=$(echo "$result" | sed 's/ //g')

echo "file name : $file\_name"

if [ -f "performance.txt" ]; then

rm -f performance.txt

echo "performance.txt has been deleted"

fi

perf stat --sync -e duration\_time,task-clock,cycles,instructions,cache-references,cache-misses,branches,branch-misses,L1-dcache-loads,L1-dcache-load-misses,LLC-load-misses,LLC-loads -r 1 -o performance.txt $1

awk '{print $1, $2, $3}' performance.txt > performance\_tmp.txt

mv performance\_tmp.txt performance.txt

duration\_time=`cat performance.txt | grep "duration\_time" | awk '{print $1}' | sed 's/,//g'`

task\_clock=`cat performance.txt | grep "task-clock" | awk '{print $1}' | sed 's/,//g'`

cpu\_cycle=`cat performance.txt | grep "cycles" | awk '{print $1}' | sed 's/,//g'`

instruction=`cat performance.txt | grep "instructions" | awk '{print $1}' | sed 's/,//g'`

cache\_references=`cat performance.txt | grep "cache-references" | awk '{print $1}' | sed 's/,//g'`

cache\_misses=`cat performance.txt | grep "cache-misses" | awk '{print $1}' | sed 's/,//g'`

branches=`cat performance.txt | grep "branches" | awk '{print $1}' | sed 's/,//g'`

branch\_misses=`cat performance.txt | grep "branch-misses" | awk '{print $1}' | sed 's/,//g'`

L1\_dcache\_loads=`cat performance.txt | grep "L1-dcache-loads" | awk '{print $1}' | sed 's/,//g'`

L1\_dcache\_load\_misses=`cat performance.txt | grep "L1-dcache-load-misses" | awk '{print $1}' | sed 's/,//g'`

LLC\_load\_misses=`cat performance.txt | grep "LLC-load-misses" | awk '{print $1}' | sed 's/,//g'`

LLC\_loads=`cat performance.txt | grep "LLC-loads" | awk '{print $1}' | sed 's/,//g'`

branches=`cat performance.txt | grep "branches" | awk '{print $1}' | sed 's/,//g'`

branch\_misses=`cat performance.txt | grep "branch-misses" | awk '{print $1}' | sed 's/,//g'`

L1\_dcache\_loads=`cat performance.txt | grep "L1-dcache-loads" | awk '{print $1}' | sed 's/,//g'`

L1\_dcache\_load\_misses=`cat performance.txt | grep "L1-dcache-load-misses" | awk '{print $1}' | sed 's/,//g'`

LLC\_load\_misses=`cat performance.txt | grep "LLC-load-misses" | awk '{print $1}' | sed 's/,//g'`

LLC\_loads=`cat performance.txt | grep "LLC-loads" | awk '{print $1}' | sed 's/,//g'`

printf "\n\n"

echo "Avg 10 times duration time: $duration\_time"

printf "Avg 10 times task clock: %.3f\n" $task\_clock

echo "Avg 10 times cpu-cycles: $cpu\_cycle"

echo "Avg 10 times instructions: $instruction"

echo "Avg 10 times cache references: $cache\_references"

echo "Avg 10 times cache misses: $cache\_misses"

echo "Avg 10 times branches: $branches"

echo "Avg 10 times branch misses: $branch\_misses"

echo "Avg 10 times L1 dcache loads: $L1\_dcache\_loads"

echo "Avg 10 times L1 dcache load misses: $L1\_dcache\_load\_misses"

echo "Avg 10 times LLC load misses: $LLC\_load\_misses"

echo "Avg 10 times LLC load: $LLC\_loads"

IPC=`echo "scale=3; $instruction / $cpu\_cycle" | bc`

printf "Avg 10 times IPC: %.3f\n" $IPC

if [ -f "$file\_name.txt" ]; then

rm -f $file\_name.txt

echo "$file\_name.txt has been deleted"

fi

echo "=================================== landscape basic\_software $1 Start ===================================">> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"duration\_time\":$duration\_time >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"task\_clock\":$task\_clock >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"cpu\_cycle\":$cpu\_cycle >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"instruction\":$instruction >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"cache\_references\":$cache\_references >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"cache\_misses\":$cache\_misses >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"branches\":$branches >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"branch\_misses\":$branch\_misses >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"L1\_dcache\_loads\":$L1\_dcache\_loads >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"L1\_dcache\_load\_misses\":$L1\_dcache\_load\_misses >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"LLC\_load\_misses\":$LLC\_load\_misses >> $file\_name.txt

echo \"landscape\":\"basic\_software\":\"$1\":\"LLC\_loads\":$LLC\_loads >> $file\_name.txt

printf "\"landscape\":\"basic\_software\":\"%s\":\"IPC\":%.3f\n" "$1" "$IPC" >> $file\_name.txt

echo "=================================== landscape basic\_software $1 Finsish ===================================">> $file\_name.txt

cat $file\_name.txt

mv $file\_name.txt $2

rm -f performance.txt

run-pyperformance-landscape.sh

#!/bin/bash

testcases="2to3 async\_generators async\_tree async\_tree\_cpu\_io\_mixed \

async\_tree\_cpu\_io\_mixed\_tg async\_tree\_io async\_tree\_io\_tg async\_tree\_memoization \

async\_tree\_memoization\_tg async\_tree\_tg asyncio\_tcp asyncio\_tcp\_ssl asyncio\_websockets \

chameleon chaos comprehensions concurrent\_imap coroutines coverage crypto\_pyaes dask \

deepcopy deltablue django\_template docutils dulwich\_log fannkuch float gc\_collect \

gc\_traversal generators genshi go hexiom html5lib json\_dumps json\_loads logging mako mdp \

meteor\_contest nbody nqueens pathlib pickle pickle\_dict pickle\_list pickle\_pure\_python \

pidigits pprint pyflate python\_startup python\_startup\_no\_site raytrace regex\_compile regex\_dna \

regex\_effbot regex\_v8 richards richards\_super scimark spectral\_norm sqlalchemy\_declarative \

sqlalchemy\_imperative sqlglot sqlglot\_optimize sqlglot\_parse sqlglot\_transpile sqlite\_synth sympy telco tomli\_loads \

tornado\_http typing\_runtime\_protocols unpack\_sequence unpickle unpickle\_list unpickle\_pure\_python xml\_etree"

read -r -a array <<< "$testcases"

timestamp=$(date +"%Y-%m-%d\_%H-%M-%S")

outputfile="pyperformance\_${timestamp}.log"

{

for testcase in "${array[@]}"; do

./performance\_counter.sh "pyperformance run --benchmarks=$testcase" "./perf-sh.log"

done

} 2>&1 | tee "$outputfile"