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-openjdk-landscape.sh

#!/bin/sh

JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64

JAVA\_OPTS="-XX:CompileThreshold=10 -XX:Tier2CompileThreshold=10 -XX:+CITime"

SPECJVM\_OPTS="-coe -ict -ikv -it 1m -i 2 -wt 1m -bt 1"

testcases="compiler.compiler compiler.sunflow compress crypto.aes crypto.rsa crypto.signverify derby mpegaudio scimark.fft.large scimark.lu.large scimark.sor.large scimark.sparse.large scimark.fft.small scimark.lu.small scimark.sor.small scimark.sparse.small scimark.monte\_carlo serial sunflow xml.transform xml.validation"

if [ -z "$JAVA\_HOME" ] ; then

echo "Please point the system variable JAVA\_HOME to where your jdk is installed."

exit

fi

if [ ! -d "$JAVA\_HOME" ] ; then

echo "Can not find folder $JAVA\_HOME"

echo "Please check your JAVA\_HOME variable."

exit

fi

if [ -f "$JAVA\_HOME/jre/bin/java" ] ; then

JAVA\_EXE="$JAVA\_HOME/jre/bin/java"

fi

if [ -f "$JAVA\_HOME/bin/java" ] ; then

JAVA\_EXE="$JAVA\_HOME/bin/java"

fi

if [ -z "$JAVA\_EXE" ] ; then

echo "Expected to find file $JAVA\_HOME/jre/bin/java or $JAVA\_HOME/bin/java"

echo "Please check your JAVA\_HOME variable or edit the script to match your JDK."

exit

fi

if [ `uname -s` = 'Darwin' ]; then

if [ ! -f "$JAVA\_HOME/lib/tools.jar" ] ; then

EXTRA\_BOOTCLASSPATH\_ARG="-Xbootclasspath/p:lib/javac.jar"

fi

fi

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

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

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

{

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

$JAVA\_EXE $EXTRA\_BOOTCLASSPATH\_ARG $JAVA\_OPTS -jar SPECjvm2008.jar $SPECJVM\_OPTS $testcase

done

} 2>&1 | tee "$outputfile"

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

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

{

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

$JAVA\_EXE $EXTRA\_BOOTCLASSPATH\_ARG $JAVA\_OPTS -jar SPECjvm2008.jar $SPECJVM\_OPTS $testcase

done

} 2>&1 | tee "$outputfile"

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

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

{

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

$JAVA\_EXE $EXTRA\_BOOTCLASSPATH\_ARG $JAVA\_OPTS -jar SPECjvm2008.jar $SPECJVM\_OPTS $testcase

done

} 2>&1 | tee "$outputfile"