



Анализ и оптимизация на софтуерни приложения

Александър Пенев

Васил Василев

Пример – Tachyon

Съдържание

1. Сваляне на примерите
2. Компилиране
3. Изпълнение на примера
4. Компиляция с различни флагове

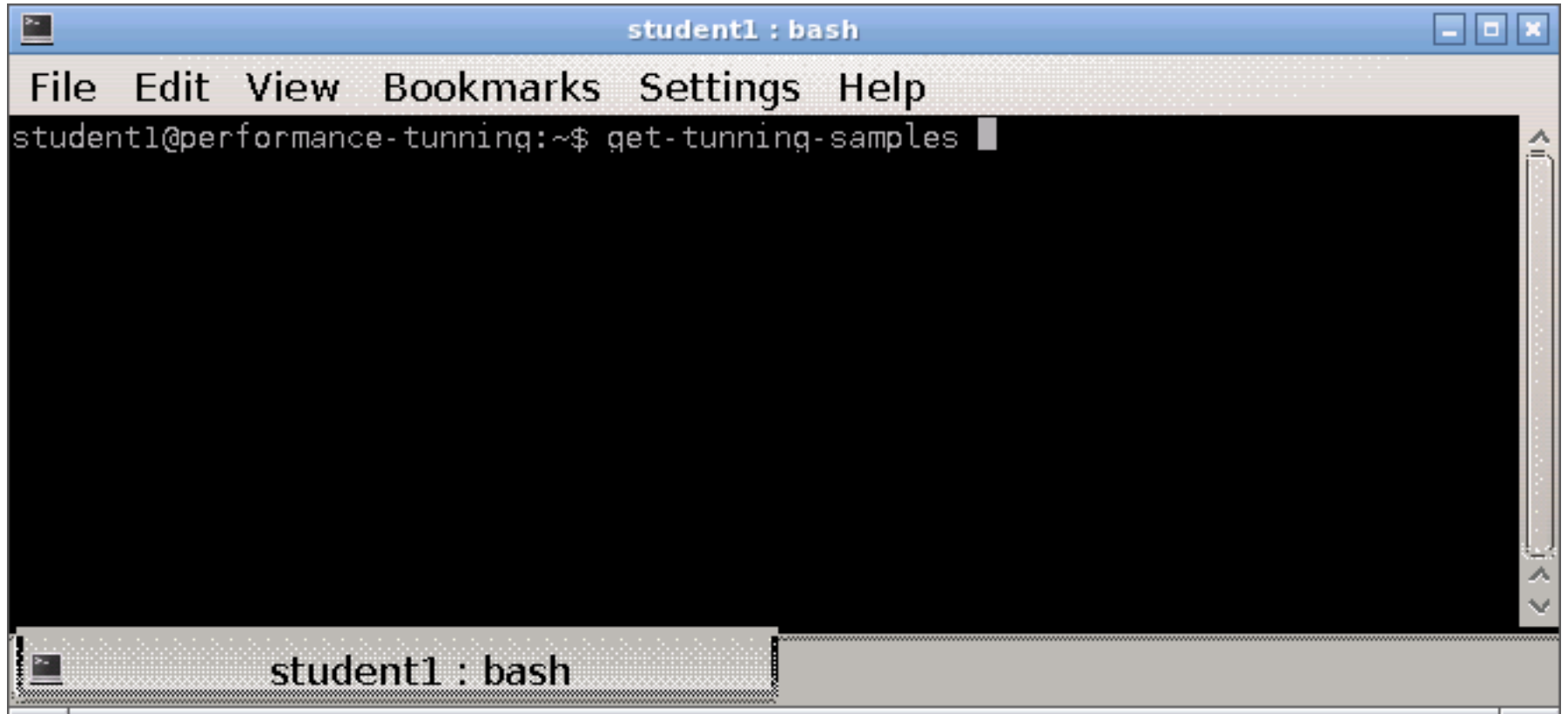


Как да свалите примерите във вашата работна среда

1. Свързвате се със сървъра по SSH и/или VNC, използвайки вашия акаунт
2. В SSH конзолата напишете **get-tuning-samples**
3. Получавате копие на сорс кода на примерите за упражненията във подпапка `samples` на вашата потребителска папка
4. Примера `samples/intel/tachyon` ще се добави към вече съществуващите примери в папката ви



Допълнете примерите си



```
student1 : bash
File Edit View Bookmarks Settings Help
student1@performance-tunning:~$ get-tunning-samples
```

Нов пример Ray tracer Tachyon

- ❖ Реализиран на C++
- ❖ Намира се в подпапката
`~/samples/intel/tachyon/...`
- ❖ Влизате в папката с
`cd ~/samples/intel/tachyon`
- ❖ Компилира се с
`make UI=con`



Стартиране на Tachyon

1. След компилацията сте в
`~/samples/intel/tachyon`
2. Стартирайте програмата
`./tachyon_find_hotspots`
3. След 10-11 секунди изпълнение програмата приключва визуализацията на сцената по подразбиране
(dat/balls.dat)
4. За да изпълним трейсъра с друга сцена
`./tachyon_find_hotspots dat/teapot.dat`



Компиляция със и без интерфейс

1. От терминала изчистете старата компиляция с
make clean
2. Компилиайте наново с
make UI=x
3. Стартирайте отново програмата
./tachyon_find_hotspots



Изпълнение с показване на резултата



Изпълнение с показване на резултата



Компиляция с различен компилатор

1. От терминала изчистете старата компиляция с
make clean
2. Компилиайте наново с
make CXX=icc UI=con
3. Стартирайте отново програмата
./tachyon_find_hotspots
4. Засечете времето за изпълнение на програмата
time ./tachyon_find_hotspots
5. Експериментирайте и комбинация с други флагове:
CXXFLAGS="-O3" , CXXFLAGS="-xHost" , ...
може да се наложи редакция на Makefile
6. Разгледайте възможните флагове на icc или gcc с
icc --help или **gcc --help**

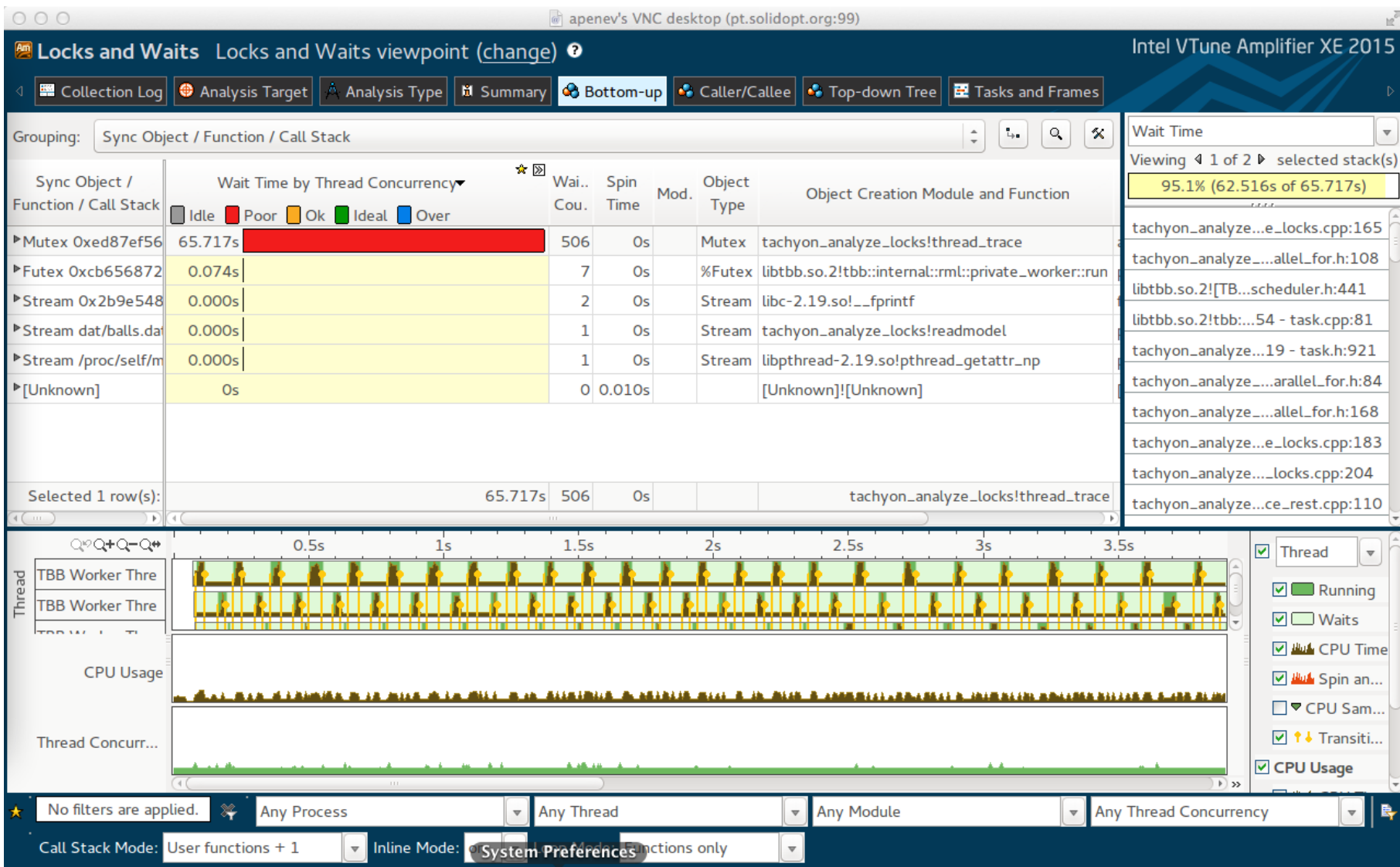


Анализ с VTune Amplifier

- ❖ Анализирайте бързодействието с VTune (tachyon_find_hotspots)
- ❖ Намерете проблемните за бързодействието на програмата места
- ❖ Изпробвайте ефекта на компилирането с различни флагове и компилатори
- ❖ Анализирайте паралелното изпълнение (tachyon_analyse_locks)



Анализ с VTune Amplifier на „Locks&Waits“



Анализ с VTune Amplifier

- ❖ След премахване на излишната критична секция в паралелния вариант

Анализ с VTune Amplifier на „Locks&Waits“

