



Анализ и оптимизация на софтуерни приложения

Александър Пенев

Васил Василев

Пример - Кеш

Съдържание

1. Сваляне на примерите
2. Компилиране
3. Изпълнение на теста
4. Задаване на различни размери на BLOCK
5. Влияние на кеша върху изпълнението

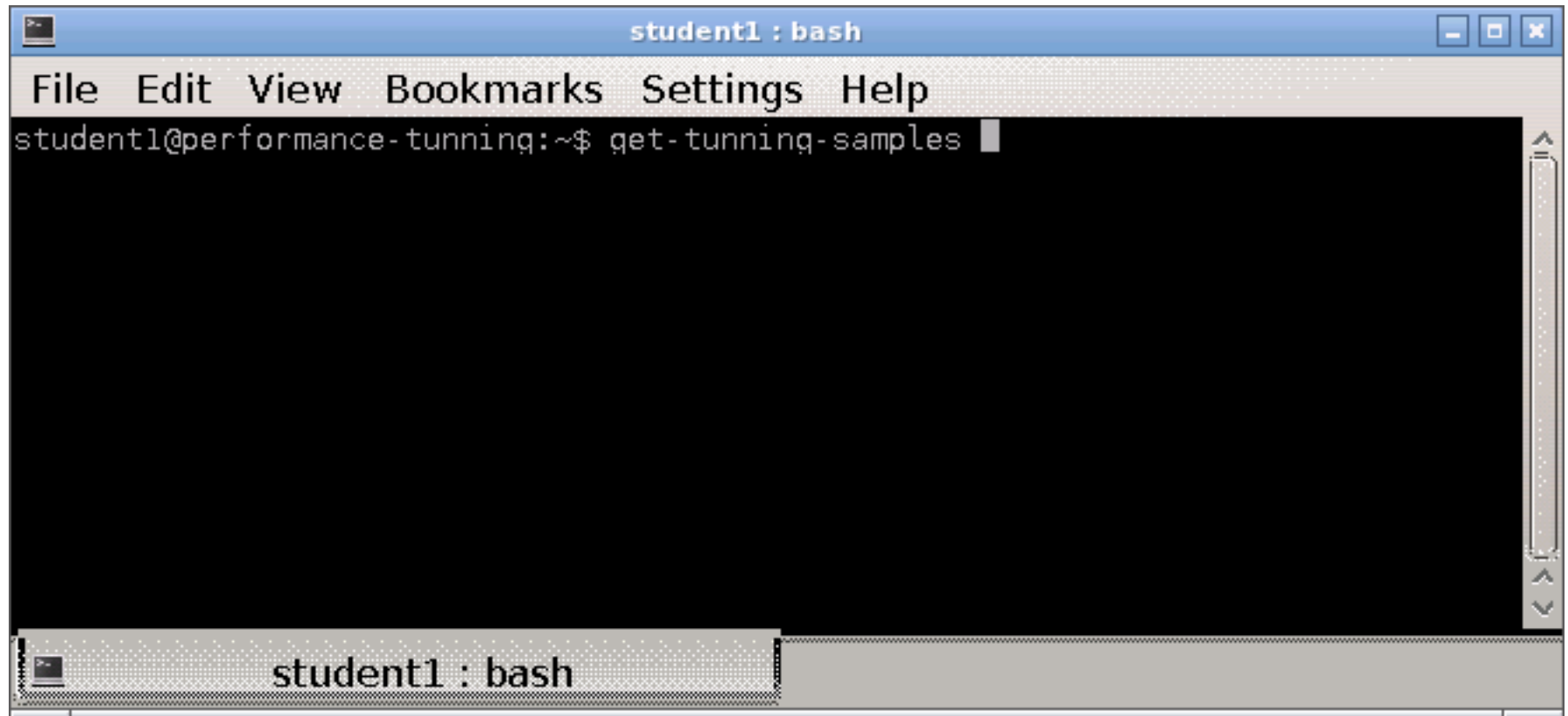


Как да свалите примерите във вашата работна среда

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 в `~/samples/tuning/matrix/C/TiledCache` е подходящ за експерименти на влиянието на големината на кеша



Допълнете примерите си

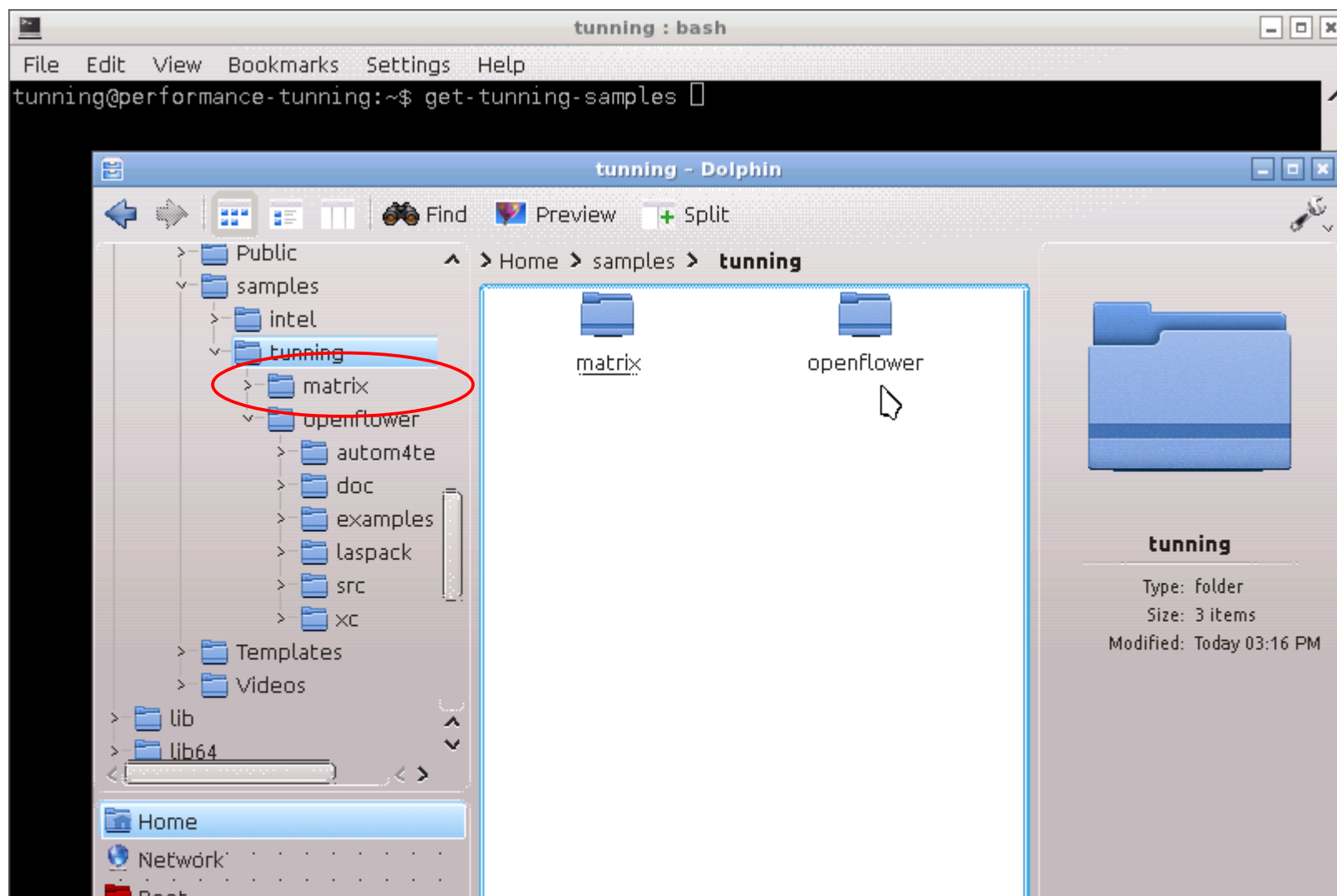


The image shows a terminal window titled "student1 : bash". The window has a menu bar with "File", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal content shows the prompt "student1@performance-tuning:~\$" followed by the command "get-tuning-samples" and a cursor. A vertical scrollbar is visible on the right side of the terminal area. At the bottom of the window, there is a status bar that also displays "student1 : bash".

```
student1@performance-tuning:~$ get-tuning-samples
```



Структура на папките в ~/samples



Нов пример *TiledCache*

- ❖ Реализиран на C
- ❖ Намира се в подпапката
~/samples/tuning/matrix/C/TiledCache
- ❖ Компилира се с:
make



Стартиране на *TiledCache*

1. От терминала влезте в
`~/samples/tuning/matrix/C/TiledCache`
2. Стартирайте програмата
`time ./TiledCache.gcc`
3. След няколко секунди изпълнение програмата приключва изпълнението си
4. Запишете времето за изпълнение



Промяна на *TiledCache*

1. Модифицирайте **TiledCache.c**
2. Променете BLOCK на 1024
3. Изпълнете **make clean**
4. Изпълнете **make**
5. Стартирайте програмата
time ./TiledCache.gcc
6. След няколко секунди изпълнение програмата приключва изпълнението си
7. Запишете времето за изпълнение



Стартиране на *TiledCache* (втори път)

1. Компилирайте отново
2. Стартирайте програмата
`time ./TiledCache.gcc`
3. След няколко секунди изпълнение програмата приключва изпълнението си
4. Запишете времето за изпълнение



Анализ с VTune Amplifier

- ❖ **Анализирайте бързодействието с VTune**
- ❖ **Намерете влиянието на големината на блока в зависимост от размера на кеша**
- ❖ **Запишете резултатите при различни размери на блока**



Как влияе BLOCK?

1. Експериментирайте с различни стойности на BLOCK
2. Намерете стойността водеща до най-голямо бързодействие
3. Защо?



Възможни причини

L1D кеш: 32 КБ

L2 кеш: 6 МБ

sizeof(double) = 8 байта

1. $3 \cdot 16 \cdot 16 \cdot 8 = 6 \text{ КБ}$	L1	L2
2. $3 \cdot 32 \cdot 32 \cdot 8 = 24 \text{ КБ}$	L1	L2
3. $3 \cdot 64 \cdot 64 \cdot 8 = 96 \text{ КБ}$	L1	L2
4. $3 \cdot 128 \cdot 128 \cdot 8 = 384 \text{ КБ}$	L1	L2
5. $2 \cdot 256 \cdot 256 \cdot 8 = 1.5 \text{ МБ}$	L1	L2
6. $3 \cdot 512 \cdot 512 \cdot 8 = 6 \text{ МБ}$	L1	L2
7. $3 \cdot 1024 \cdot 1024 \cdot 8 = 24 \text{ МБ}$	L1	L2
8. $3 \cdot 2048 \cdot 2048 \cdot 8 = 32 \text{ МБ}$	L1	L2

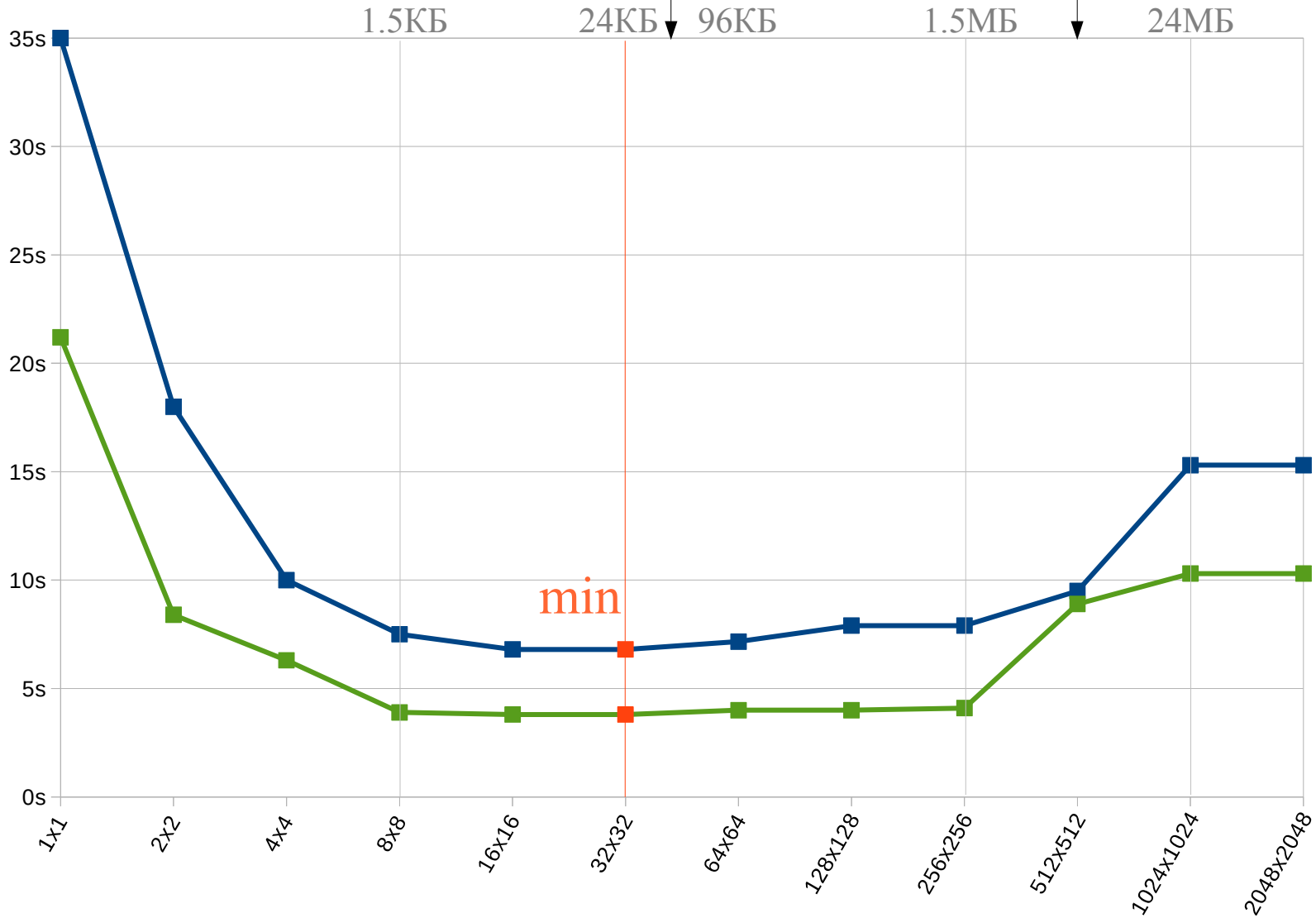


TiledCache – 1024x1024

i7 (32КБ, 256КБ, 6МБ)

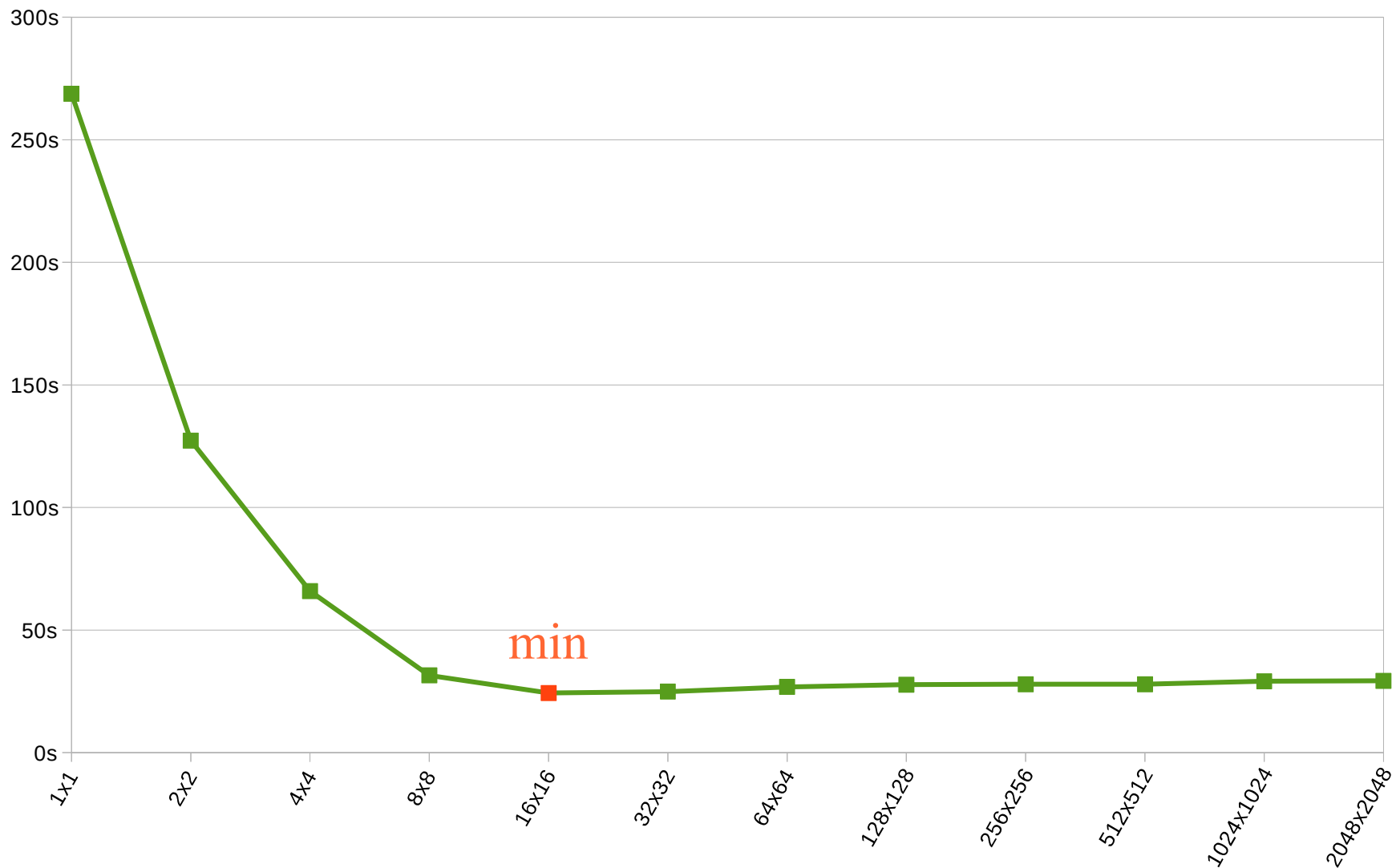
L1D=32КБ

L3=6МБ



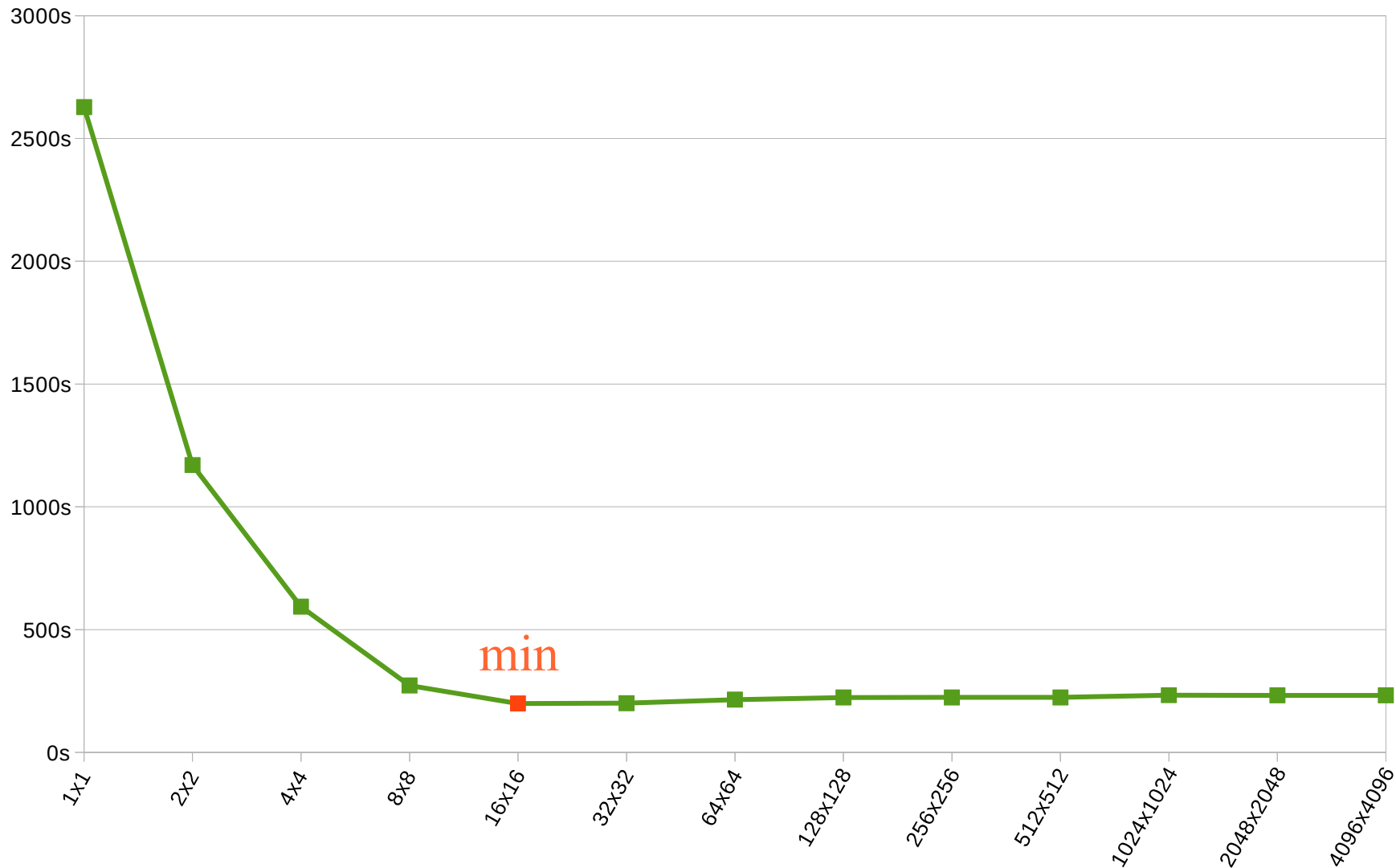
TiledTansposedCache – 2048x2048

i7 (32КБ, 256КБ, 6МБ)



TiledTansposedCache – 4096x4096

i7 (32КБ, 256КБ, 6МБ)



TiledTansposedCache – 8192x8192

i7 (32KB, 256KB, 6MB)

