Въведение:

„Intel VTune Amplifier XE for Linux” (предишно наименование „Intel VTune Performance Analyzer“) е предназначен за откриване на „тесни места” в 32- и 64-битови Linux приложения, а също така в Java-приложения за системи с архитектура IA-32. Той осигурява сбор и анализ на данни за интегрална производителност на процеси и приложения, а също и на техните съставни: функции, модули и инструкции. VTune Amplifier предлага интерфейс с команди за настройка и сбор на данни, съдържа допълнителен модул за визуално представяне на йерархията на модули и функции на приложенията, поддържа сценарни езици за управление, подобни на PERL.

Анализирането е извършено върху три приложения, две от тях компилирани под Linux, като за компилатор е използван GCC и като допълнение, библиотеката на MPI (Message Passing Interface). Третото приложение е базирано върху POSIX Threads, отновно компилирано под Linux, използвайки компилатора GCC като за целта е добавен флага „-pthread“ преди компилация без допълнителни библиотеки. Целта е да се анализира поведението на различните методи за разпределение на натоварването по отделните ядра на процесора. Трите приложения изпълняват сходни действия, като основното им действие е да извършват умножение на матрици.

Първото приложение (*mpi-mm-instant*) е реализирано с динамично заделяне на памет за матриците, без добро разпределение и освобождаване на паметта с риск за така наречения *„memory leak“*. То приема един входен параметър, който отговаря за броя на редовете и колоните.

Второто приложение (*mpi-mm-instant-better)* е реализирано със статични променливи, строго дефинирани размери и стойности без рискове за разпределение на паметта. Входни параметри не се използват.

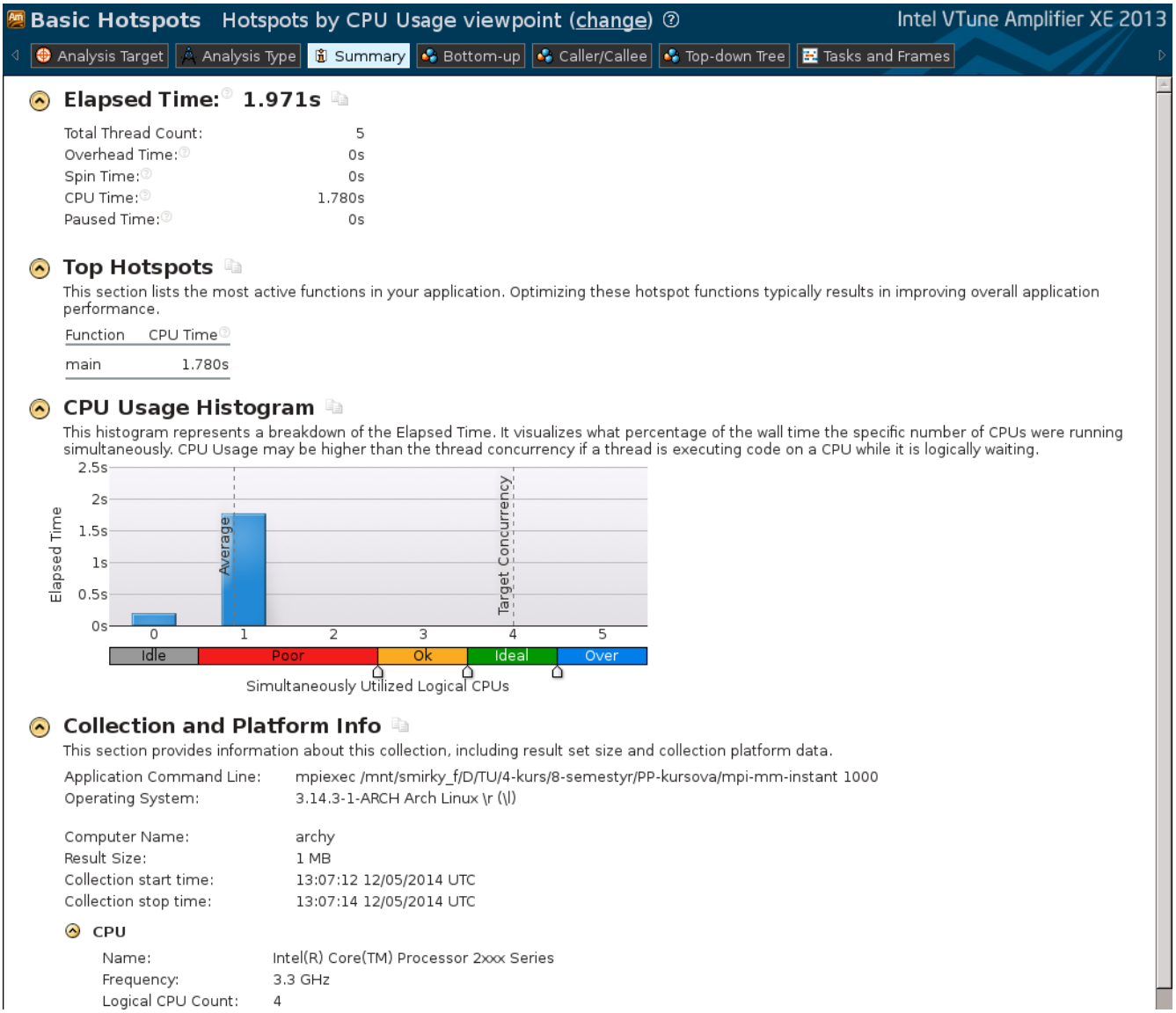
Третото приложение (*posix-mm)* е реализирано с POSIX функции за нишки (*pthreads)*, като за размера на матриците се заема памет зависеща от входните параметри. Умножението на матриците се разпределя по нишките. Освобождаването на паметта също не е реализирано по най-подходящия начин с риск за *„memory leak“*.

Резултатите са описани и изобразени с данни и графики в следващите страници.

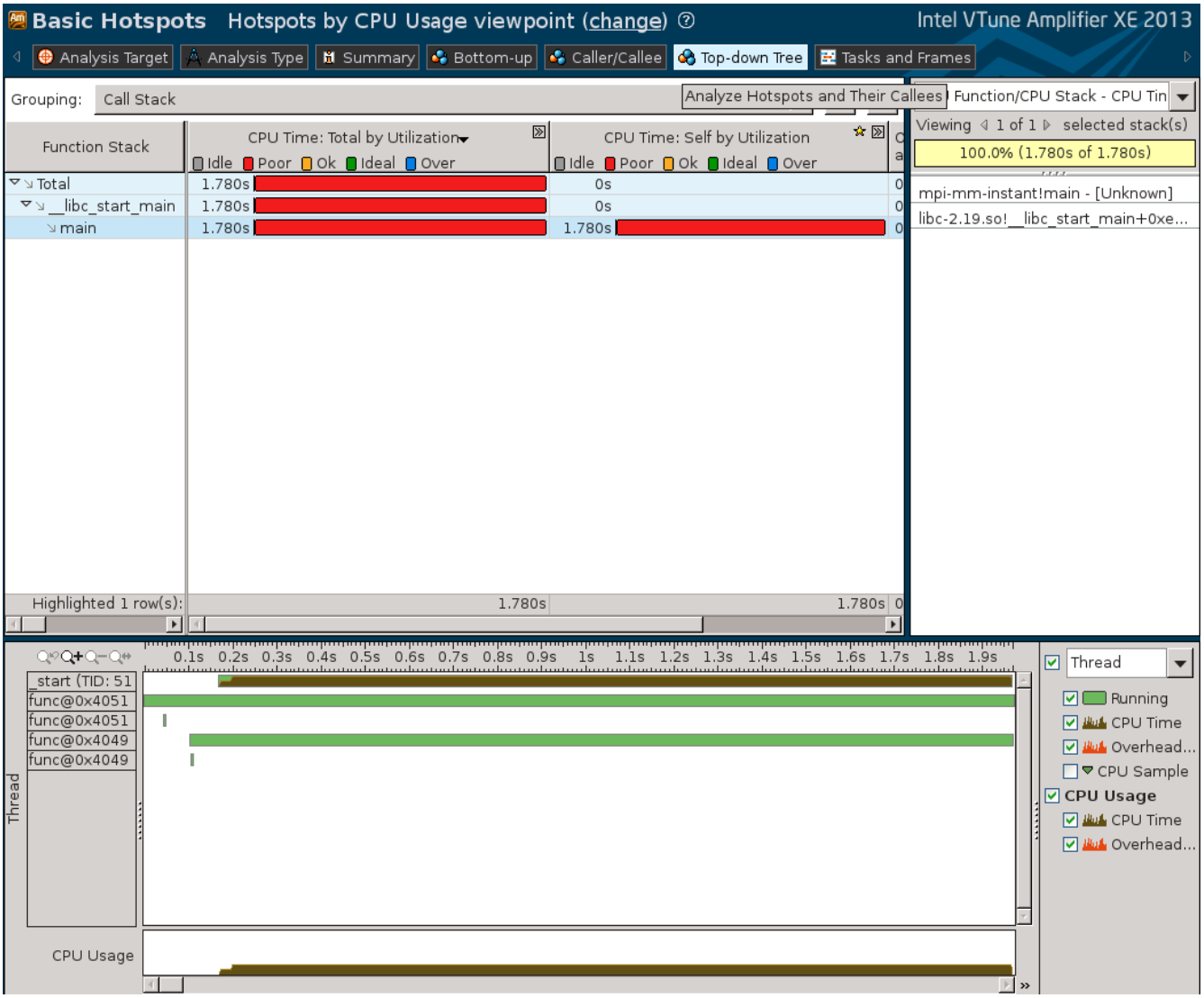
Резултати:

програма: ***mpi-mm-instant***
входни параметри: ***1000***

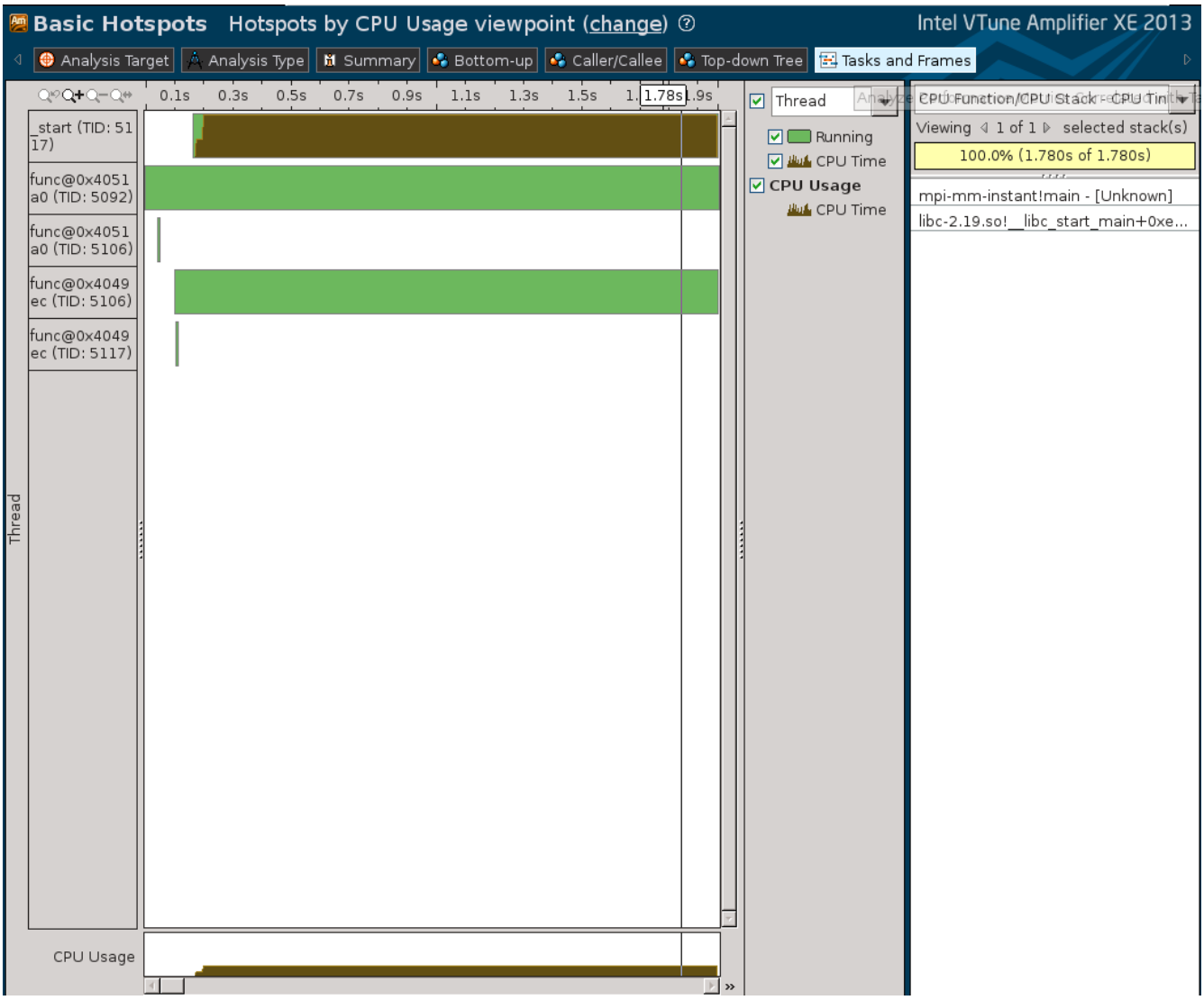
* Времетраене и графика на зареждането на приложението с 1000 елемента.



* Времетраене на отделните функции на програмата (включително и библиотеки).

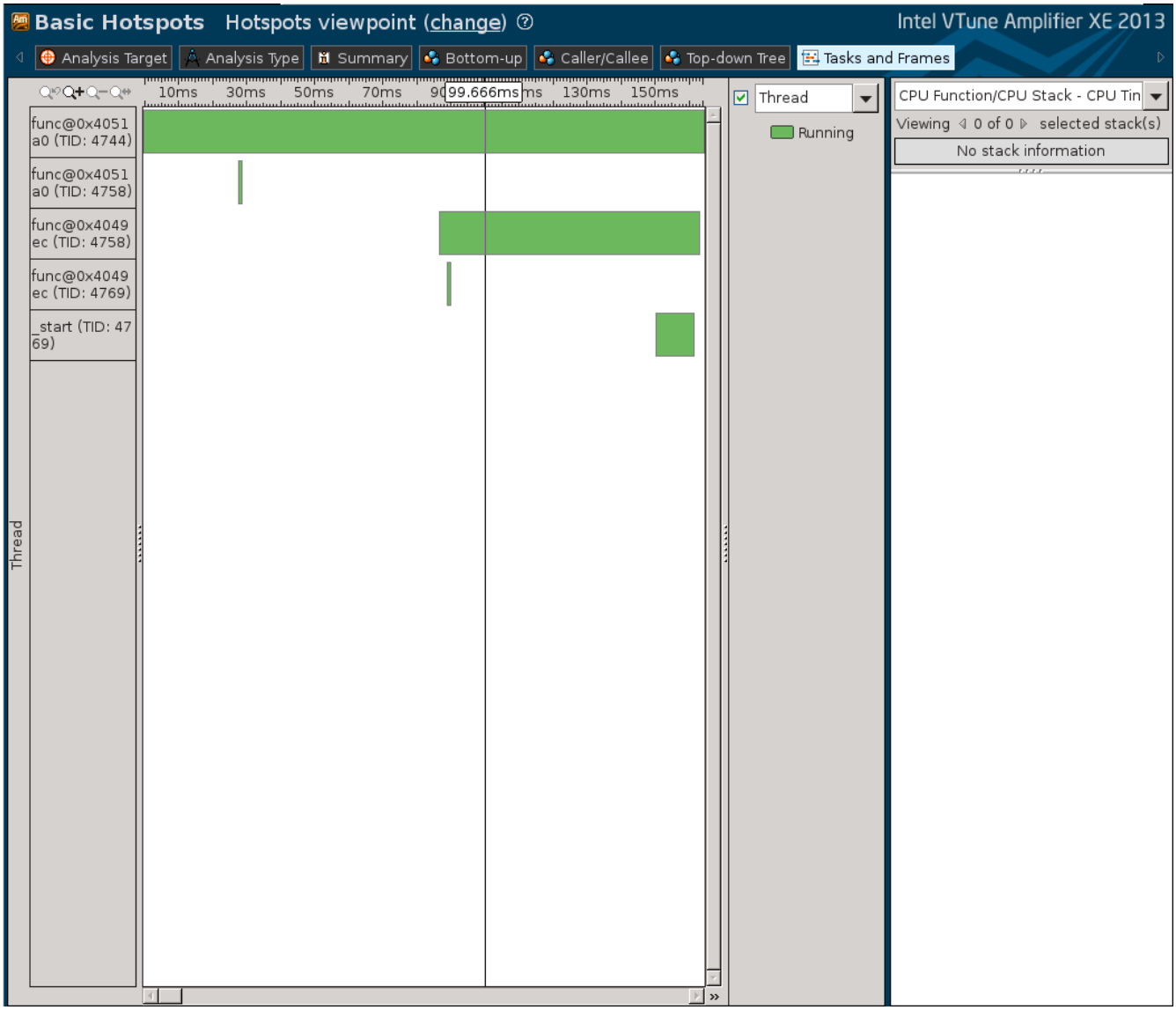


* Разпределяне на работата по отделните нишки.



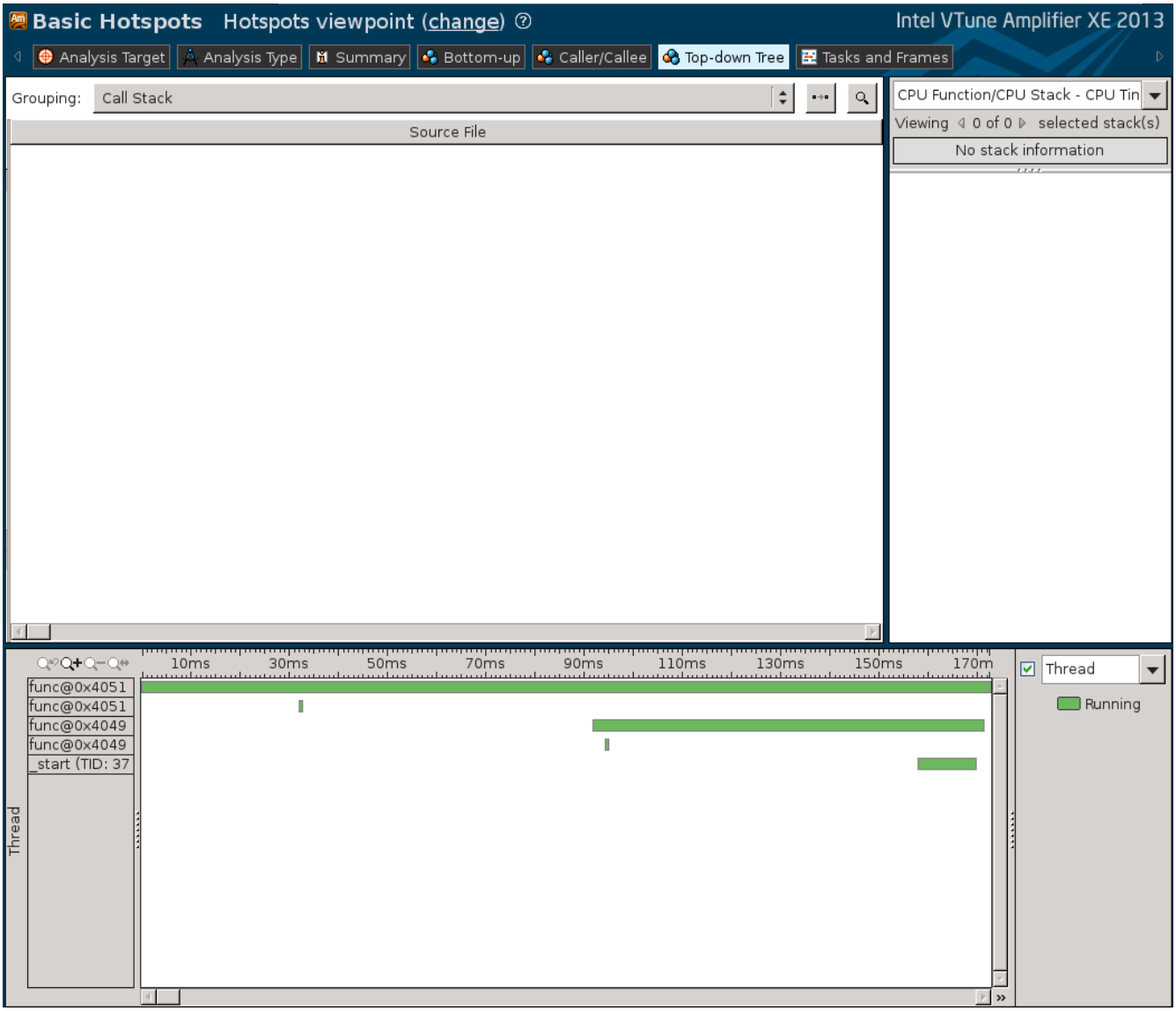
програма: ***mpi-mm-instant***
входни параметри: ***9***

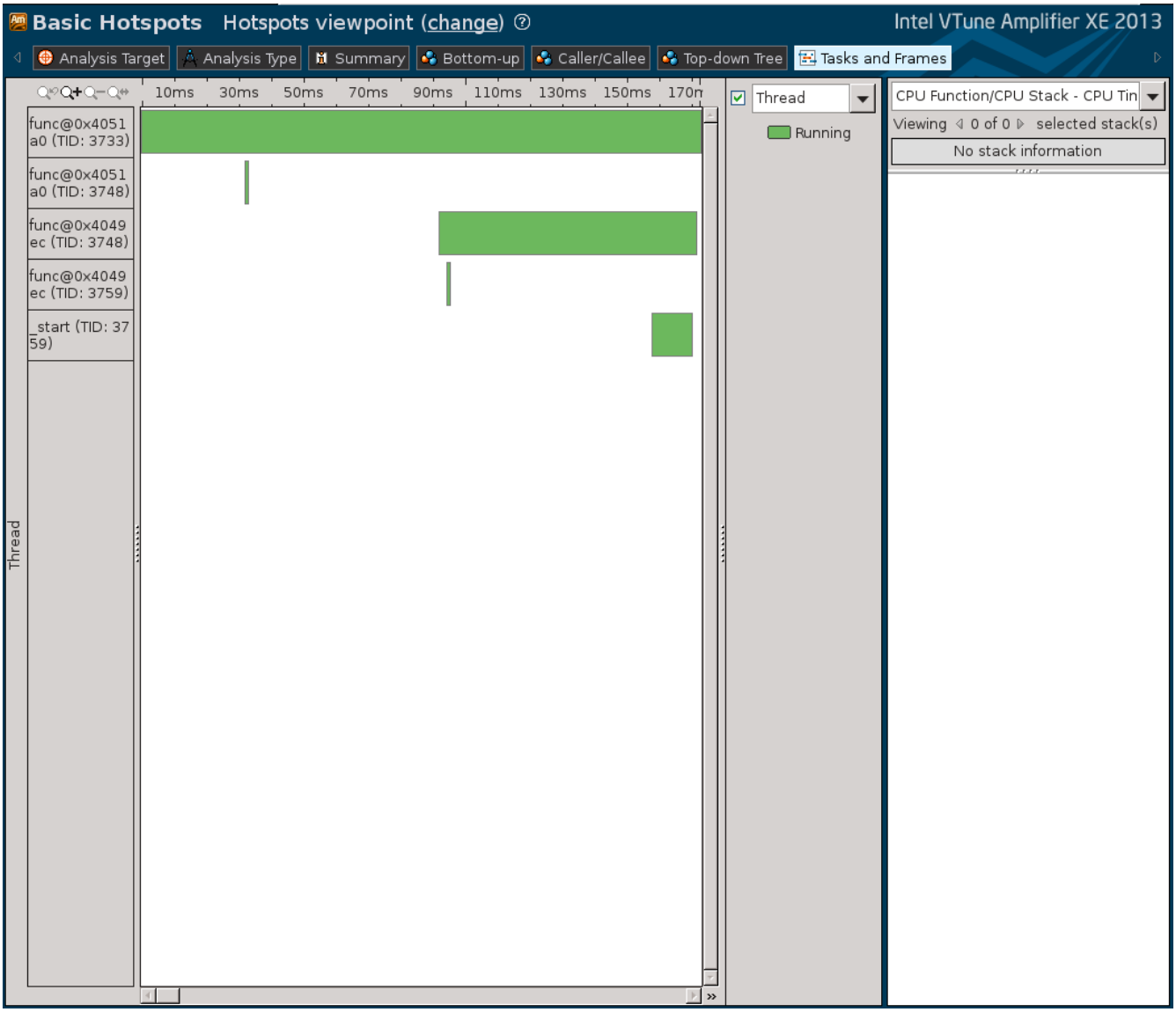
* Разпределяне на работата по отделните нишки.



програма: ***mpi-mm-instant-better***
входни параметри: ***няма***

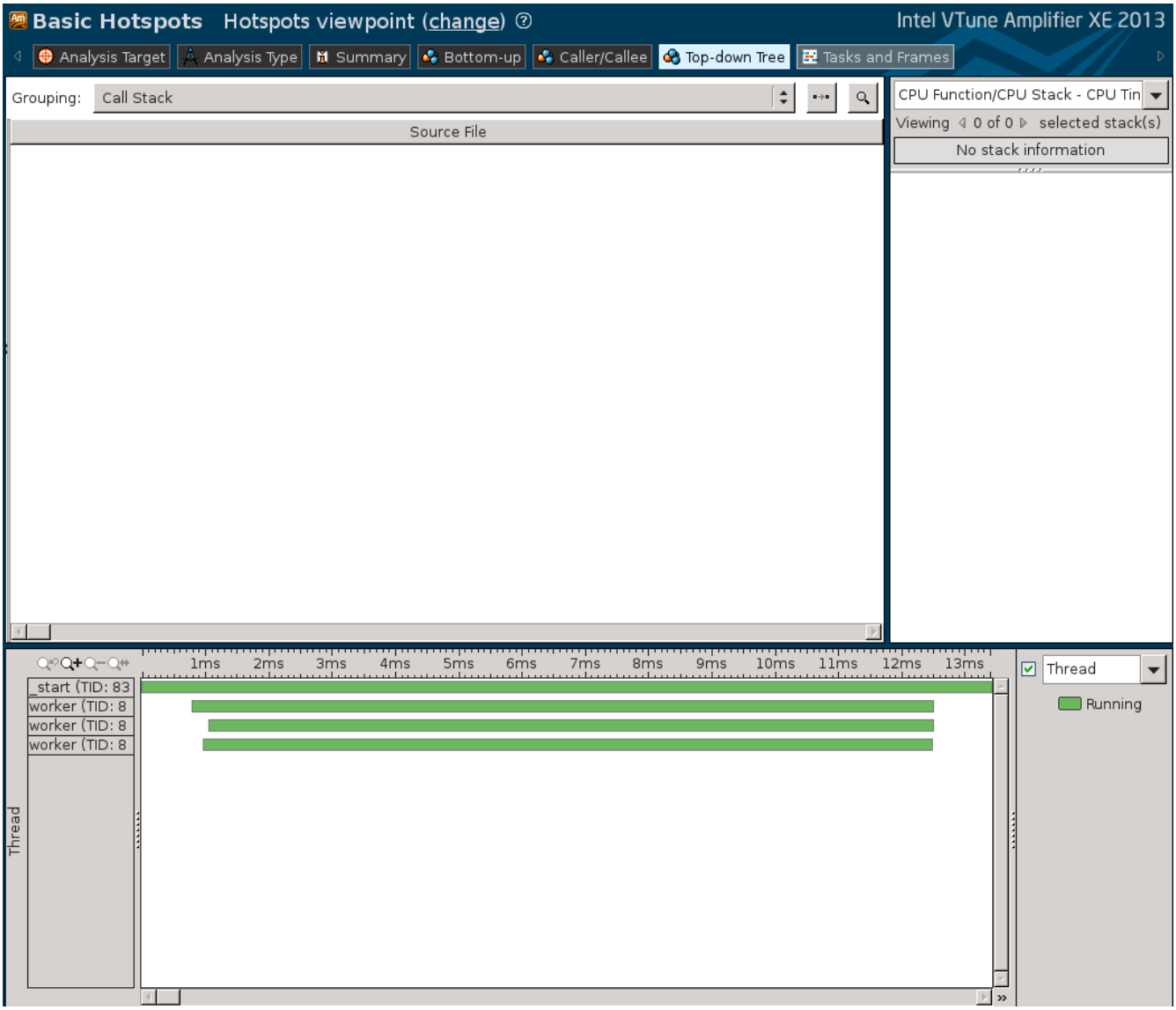
* Разпределяне на работата по отделните нишки.

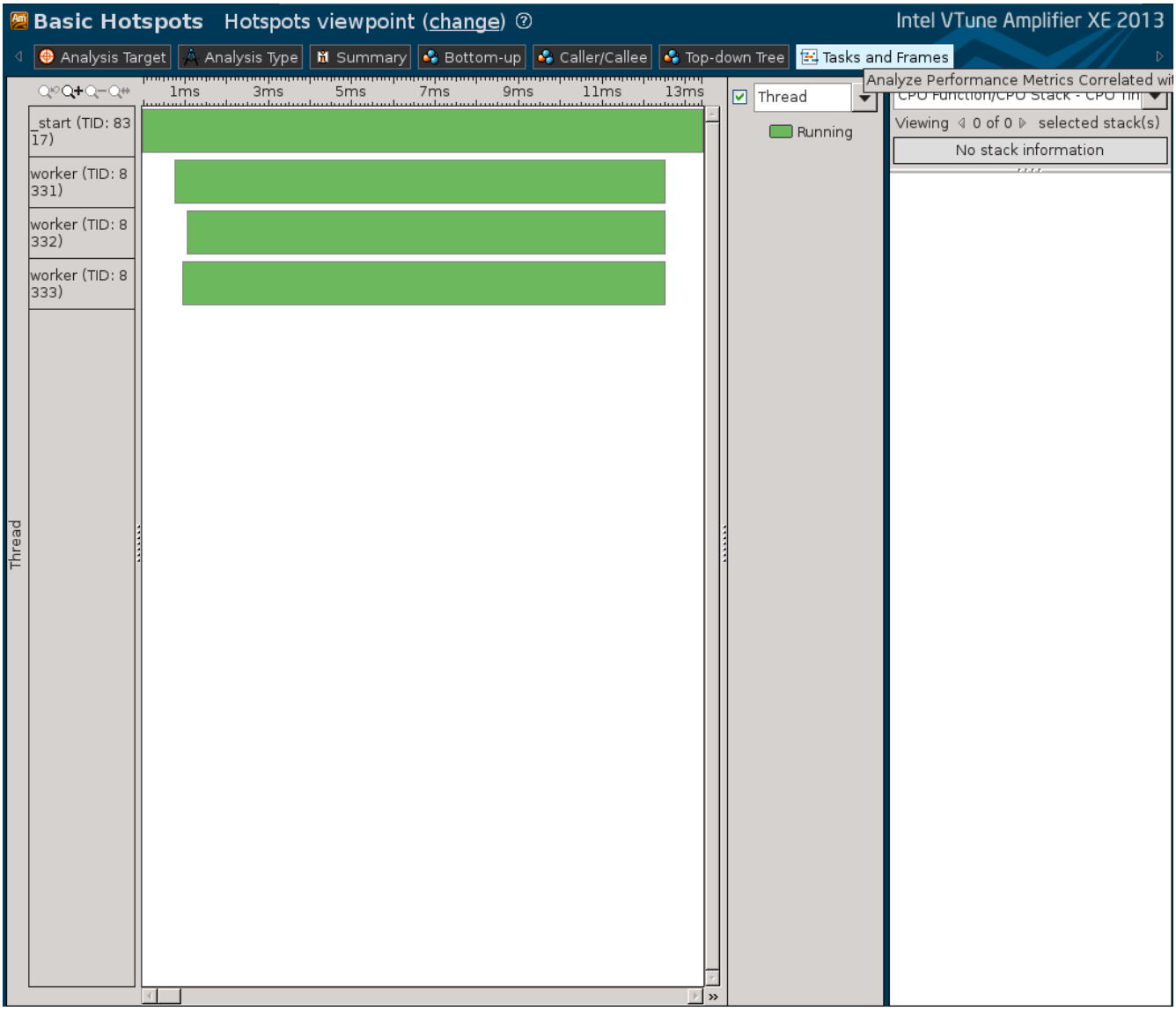



Basic Hotspots
Hotspots viewpoint (change)
Intel VTune Amplifier XE 2013
Analysis Target
Analysis Type
Summary
Bottom-up
Caller/Callee
Top-down Tree
Tasks and Frames
10ms
30ms
50ms
70ms
90ms
110ms
130ms
150ms
170m
func@0x4051a0 (TID: 3733)
func@0x4051a0 (TID: 3748)
func@0x4049ec (TID: 3748)
func@0x4049ec (TID: 3759)
_start (TID: 3759)
Thread
Thread
Running
CPU Function/CPU Stack - CPU Tin
Viewing 0 of 0 selected stack(s)
No stack information

програма: ***posix-mm***
входни параметри: ***9 3***

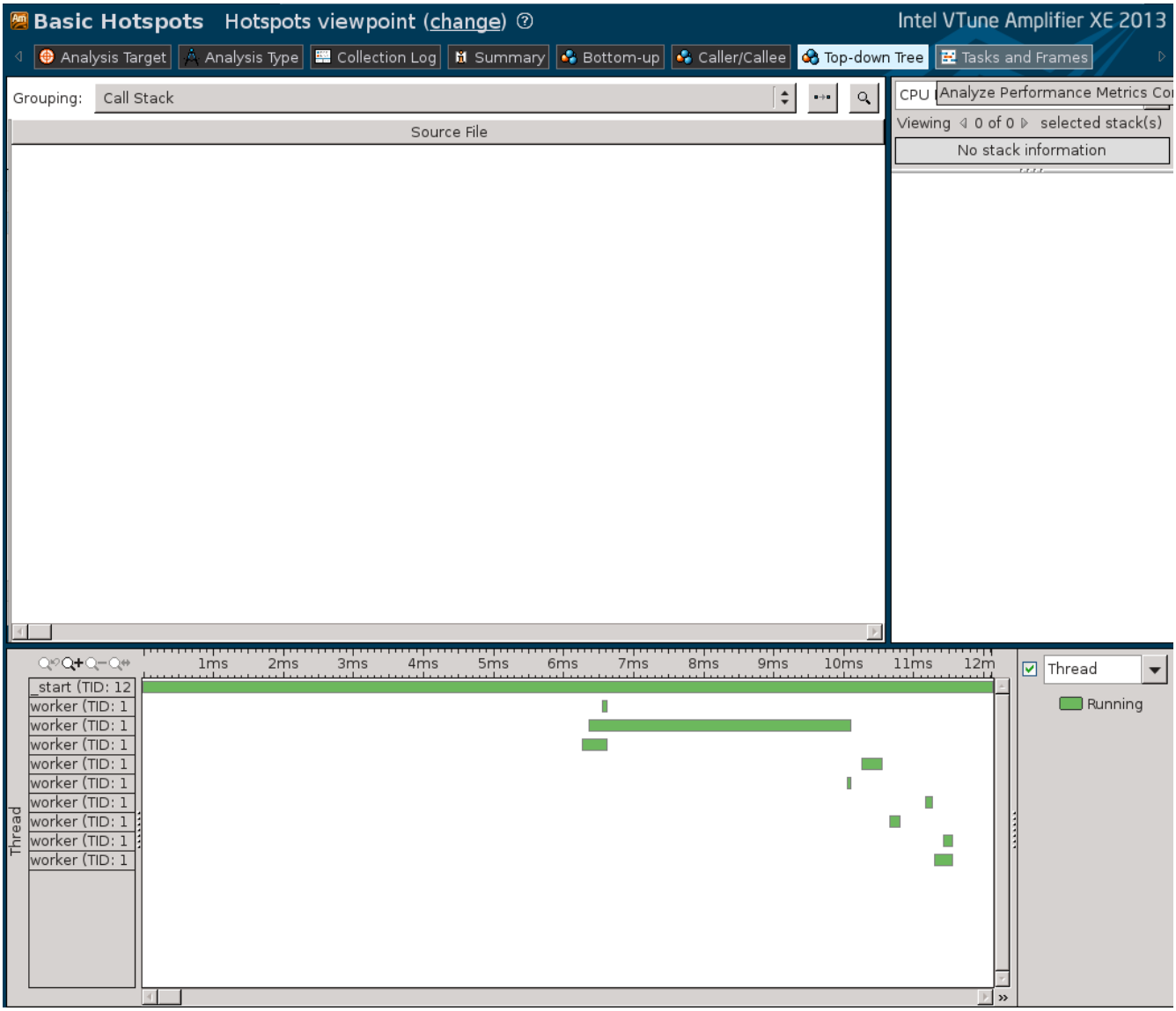
* Разпределяне на работата по **трите** нишки с 9 елемента.

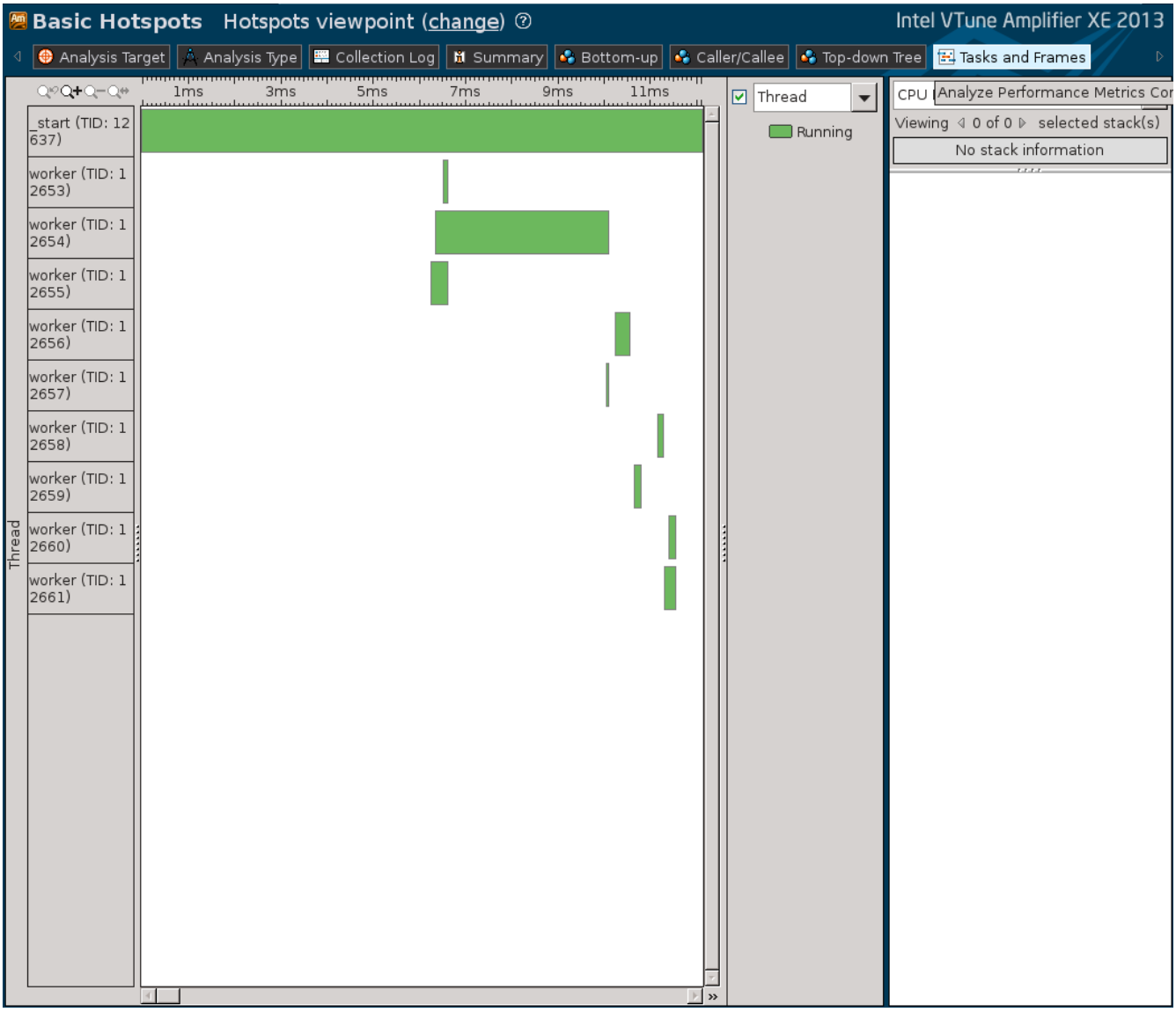



Basic Hotspots
Hotspots viewpoint (change)
Intel VTune Amplifier XE 2013
Analysis Target
Analysis Type
Summary
Bottom-up
Caller/Callee
Top-down Tree
Tasks and Frames
Analyze Performance Metrics Correlated wi
1ms
3ms
5ms
7ms
9ms
11ms
13ms
_start (TID: 8317)
worker (TID: 8331)
worker (TID: 8332)
worker (TID: 8333)
Thread
Thread
Running
Viewing 0 of 0 selected stack(s)
No stack information

програма: ***posix-mm***
входни параметри: ***9 9***

* Разпределяне на работата по **<u>деветте</u>** нишки с 9 елемента.





Изводи:

От предходните графики и данни можем да направим извод, че паралелното програмиране е доста сериозна част от компютърната сфера. То позволява равномерно разпределяне на товара по използвания хардуер, в резултат на което функционалността се повишава и ускорява. Паралелното програмиране се използва все повече и повече в съвременните технологии и програми като VTune ни посочва къде е най-наложително да подобрим кода на приложенията ни.