

## بخش اول آزمایش: بررسی Hotspotها

Source	CPU Time: Total	CPU Time: Self
23 // Output a start message		
24 printf("None Parallel Timings for %d iterations\n", VERYBIG);		
25 // repeat experiment several times		
26 for (i = 0; i < 1; i++)		
27 {		
28 // get starting time56 x CHAPTER 3 PARALLEL STUDIO XE FOR THE IMPATIENT		
29 starttime = timeGetTime();		
30 // reset check sum & running total		
31 sum = 0;		
32 total = 0.0;		
33 // Work Loop, do some work by looping VERYBIG times		
34 //pragma omp parallel for private(sumx, sumy, k) reduction(+: sum, total)		
35 for (j = 0; j < VERYBIG; j++)		
36 {		
37 // increment check sum		
38 sum += 1;		
39 // Calculate first arithmetic series		
40 sumx = 0.0;		
41 for (k = 0; k < j; k++)		
42 sumx = sumx + (double)k;	1.2%	6.404usec
43 // Calculate second arithmetic series		
44 sumy = 0.0;		
45 for (k = j; k > 0; k--)		
46 sumy = sumy + (double)k;	98.8%	512.342usec
47 if (sumx > 0.0) total = total + 1.0 / sqrt(sumx);		
48 if (sumy > 0.0) total = total + 1.0 / sqrt(sumy);		
49 }		
50 // get ending time and use it to determine elapsed time		
51 elapsedtime = timeGetTime() - starttime;		
52 //printf("%d", omp_get_num_threads());		
53 // report elapsed time		
54 printf("Time Elapsed %10d mSecs Total = %1f Check Sum = %1d\n",		
55 (int)elapsedtime, total, sum);		
56 }		
57 // return integer as required by function header		
58 return 0;		
59 }		

Function / Call Stack	CPU Time	Module	Function (Full)	Source File	Start Address
main	518.747usec	MultiCore-Lab-3.exe	main	MultiCore-Lab-3.cpp	0x140011840

CPU Time: 100.0% (0.519ms of 0.519ms)

MultiCore-Lab-3.exe!main - MultiCore-Lab-3.cpp

MultiCore-Lab-3.exe!func@0x140012450+0x38 - [unknown source file]

MultiCore-Lab-3.exe!func@0x140012200+0x12d - [unknown source file]

MultiCore-Lab-3.exe!func@0x1400121e0+0xd - [unknown source file]

MultiCore-Lab-3.exe!mainCRTStartup+0x8 - exe\_main.cpp:16

KERNEL32.DLL!BaseThreadInitThunk+0x13 - [unknown source file]

ntdll.dll!RtlUserThreadStart+0x20 - [unknown source file]

Thread: func@0x14001100f (TID: 15...)

CPU Utilization: 100.0%

# بررسی Deadlock ها:

**Locate Deadlocks and Data Races**

Problems

ID	Type	Sources	Modules	State
P1	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New
P2	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New
P3	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New

Filters

Severity	Count
Error	3 item(s)

Timeline

- OMP Worker Thread #2 (4768)
- OMP Worker Thread #3 (17012)

**Locate Deadlocks and Data Races**

Problems

ID	Type	Sources	Modules	State
P1	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New
P2	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New
P3	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New

Filters

Severity	Count
Error	3 item(s)

Timeline

- OMP Worker Thread #2 (4768)
- OMP Worker Thread #3 (17012)

**Locate Deadlocks and Data Races**

Problems

ID	Type	Sources	Modules	State
#P1	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New
#P2	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New
#P3	Data race	MultiCore-Lab-3.cpp	multicore-lab-3.exe	New

Filters

Severity	Count
Error	3 item(s)
Type	
Data race	3 item(s)
Source	
MultiCore-Lab-3.cpp	3 item(s)
Module	
multicore-lab-3.exe	3 item(s)
State	
New	3 item(s)
Suppressed	
Not suppressed	3 item(s)
Investigated	
Not investigated	3 item(s)

Code Locations: Data race

Description	Source	Function	Module	Variable
Read	MultiCore-Lab-3.cpp:38	mainSomp\$parallel@37	multicore-lab-3.exe	j
Write	MultiCore-Lab-3.cpp:38	mainSomp\$parallel@37	multicore-lab-3.exe	j

Timeline

- OMP Master Thread #0 (11544)
- OMP Worker Thread #1 (17004)

## بررسی Static Threading

**Threading**

Analysis Configuration | Collection Log | Summary | Bottom-up | Caller/Callee | Top-down Tree | Platform

Elapsed Time<sup>®</sup>: 73.696s  
Paused Time<sup>®</sup>: 0s

Effective CPU Utilization<sup>®</sup>: 56.7% (2.269 out of 4 logical CPUs) 📊

Effective CPU Utilization Histogram<sup>®</sup>  
This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.

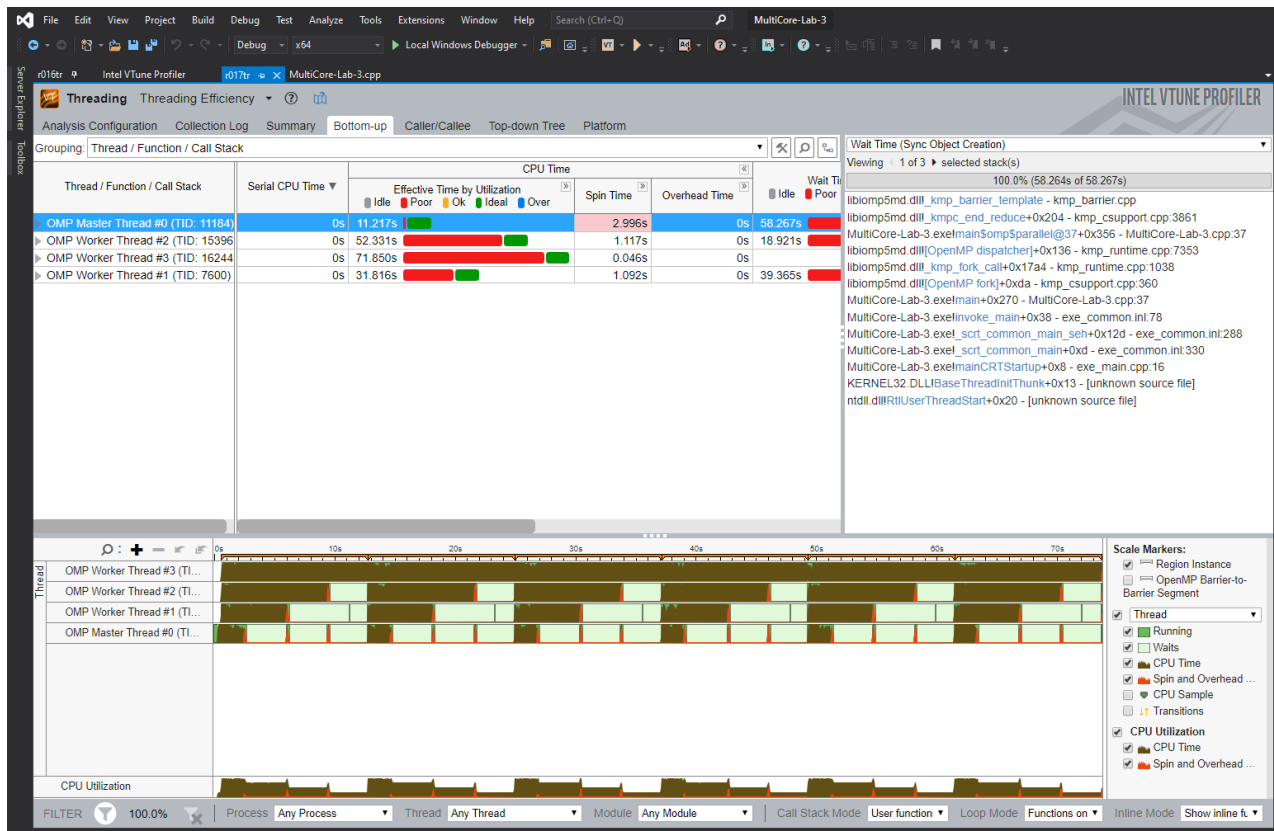
OpenMP Analysis. Collection Time<sup>®</sup>: 73.696s

- Serial Time (outside parallel regions)<sup>®</sup>: 0.640s (0.9%)
- Parallel Region Time<sup>®</sup>: 73.057s (99.1%)
  - Estimated Ideal Time<sup>®</sup>: 42.418s (57.6%)
  - OpenMP Potential Gain<sup>®</sup>: 30.638s (41.6%) 📈
- Top OpenMP Regions by Potential Gain  
This section lists OpenMP regions with the highest potential for performance improvement. The Potential Gain metric shows the elapsed time that could be saved if the region was optimized to have no load imbalance assuming no runtime overhead.

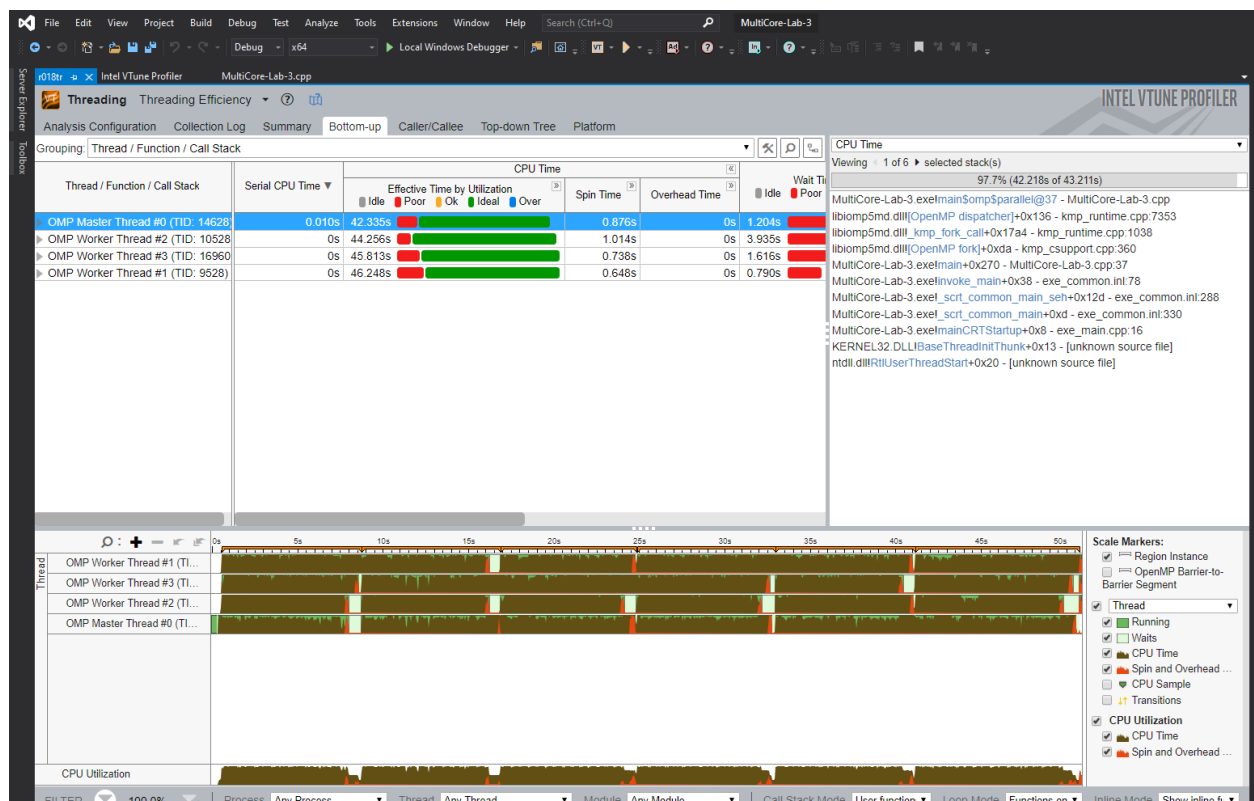
OpenMP Region	OpenMP Potential Gain <sup>®</sup> (%) <sup>®</sup>	OpenMP Region Time <sup>®</sup>
mainSomp\$parallel.4@unknown:37:38	30.638s 📈 41.6%	73.057s

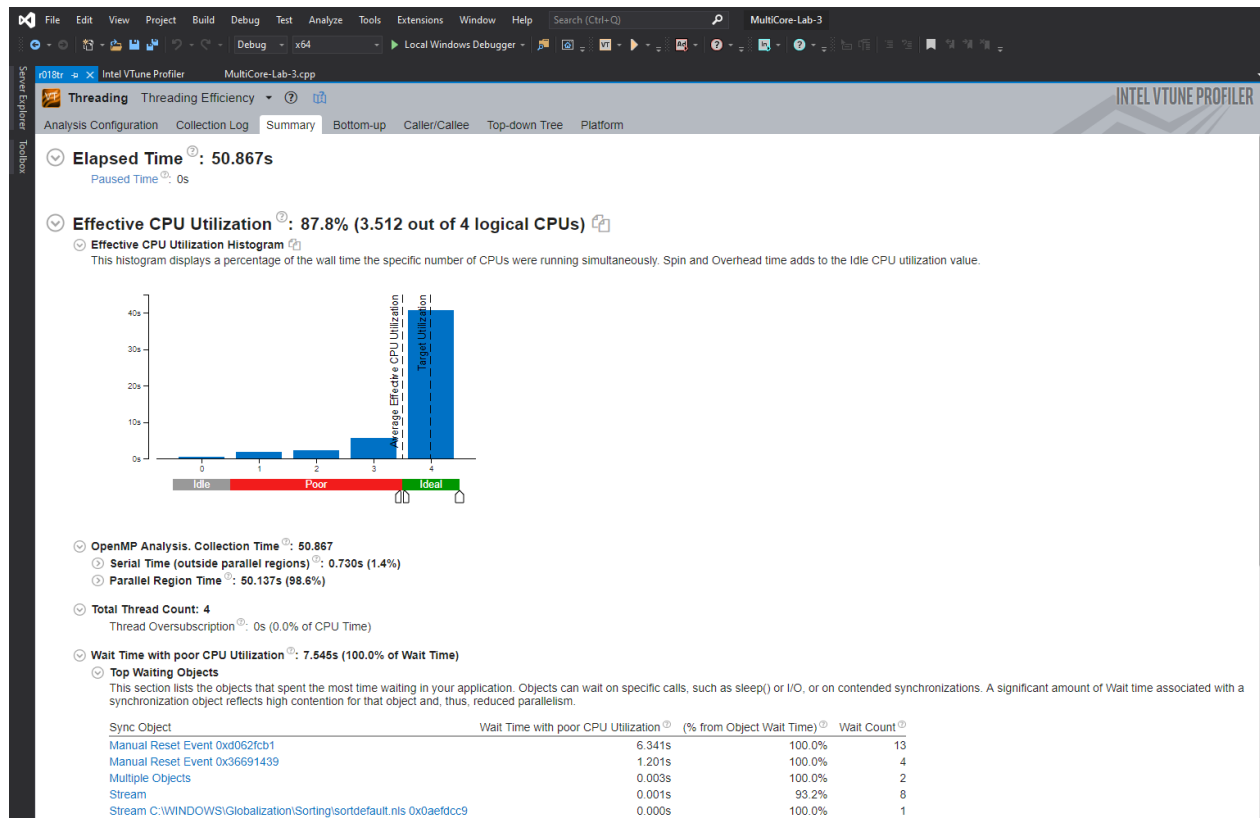
\*N/A is applied to non-summable metrics.

Total Thread Count: 4  
Thread Oversubscription<sup>®</sup>: 0s (0.0% of CPU Time)



## بررسی Dynamic Threading:

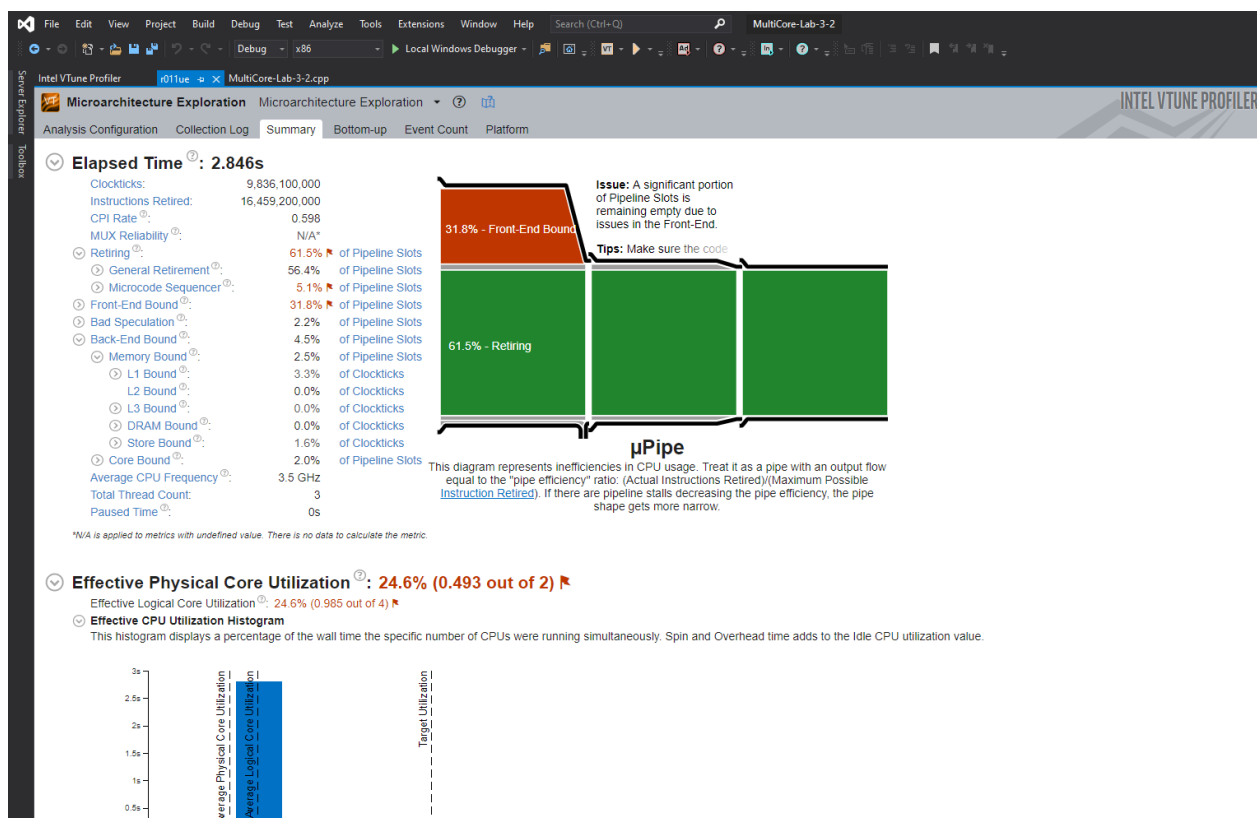




بخش دوم آزمایش:

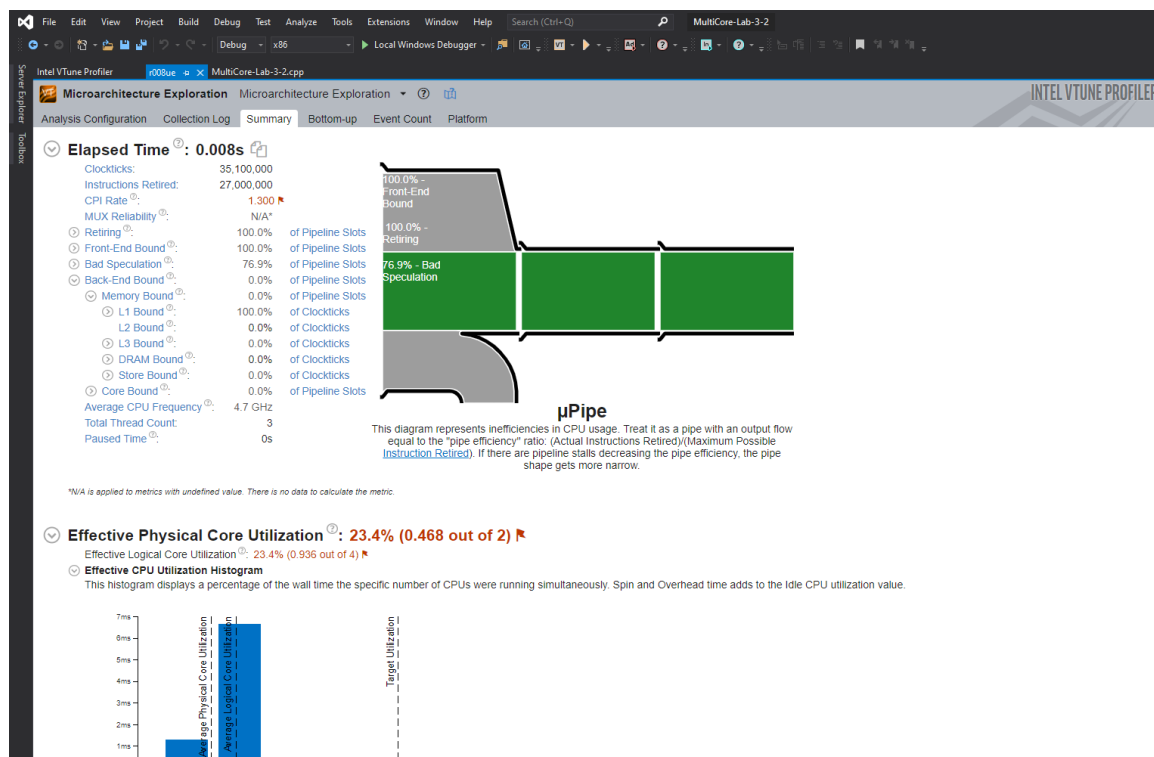
بررسی حالت ۱:

برای ۶۴:

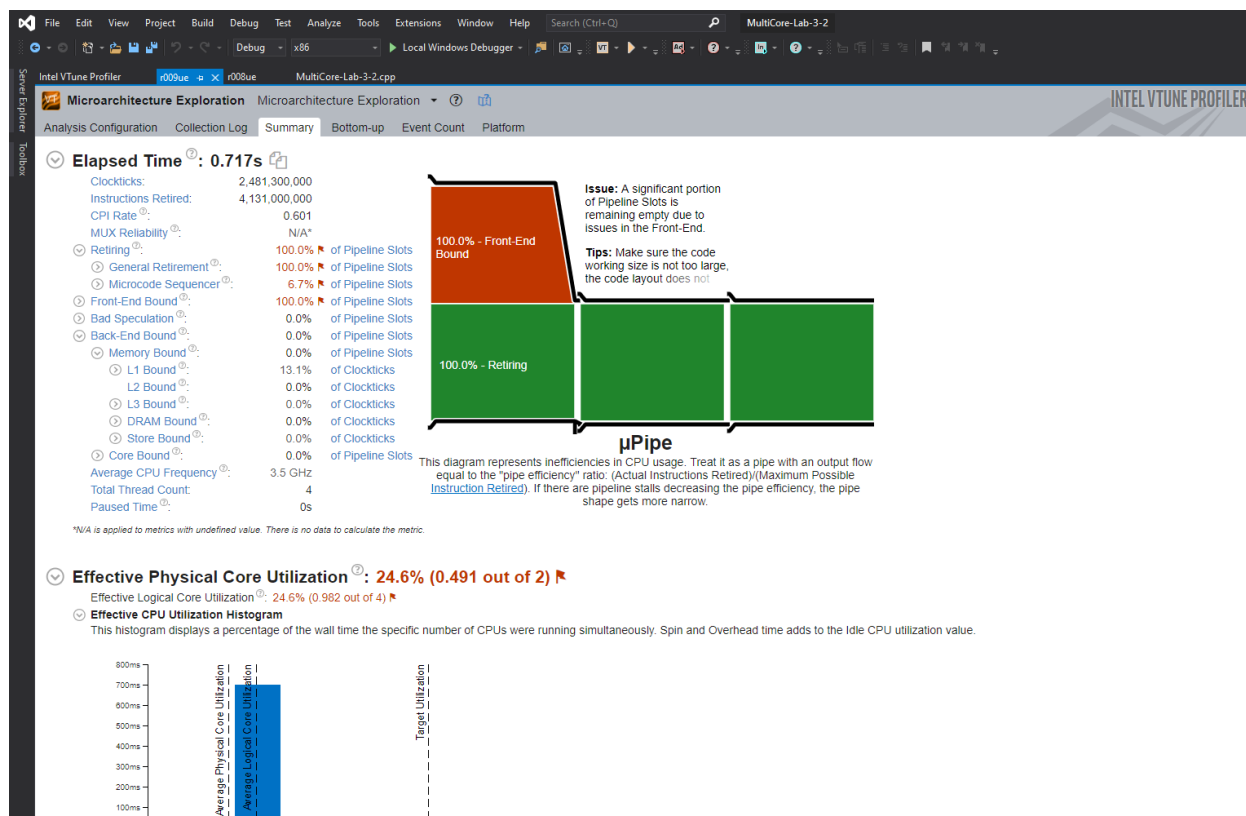


1

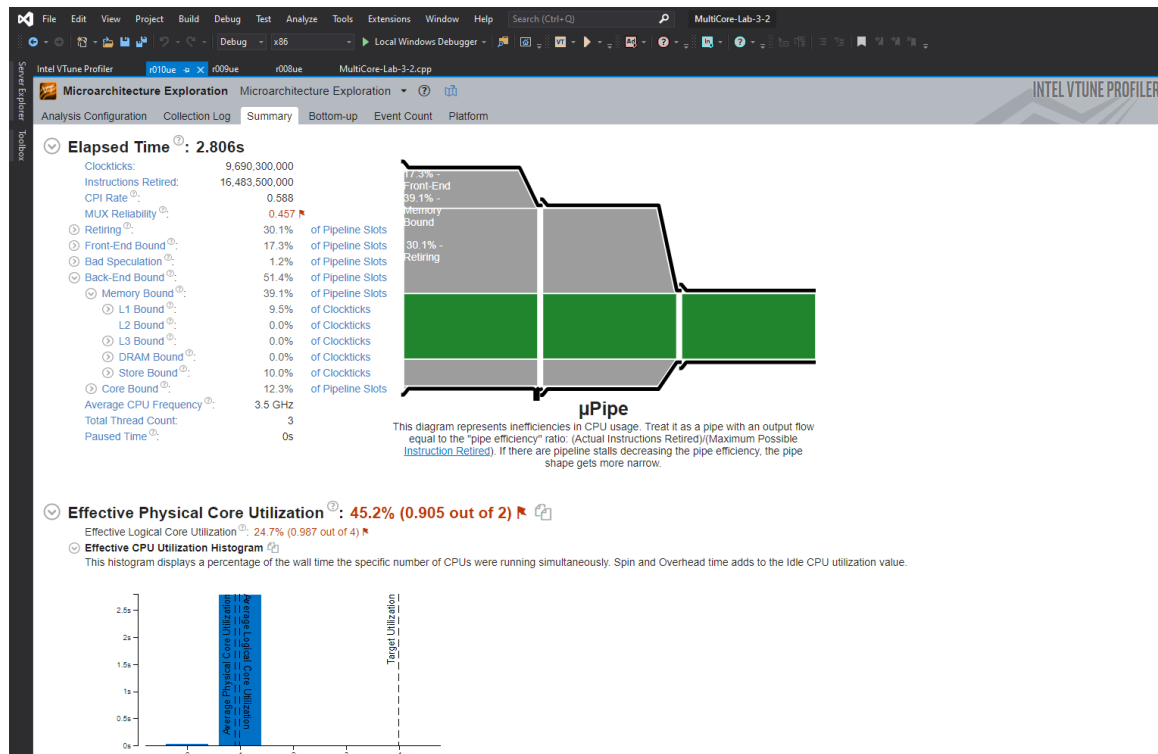
برای ۲۰۴۸:



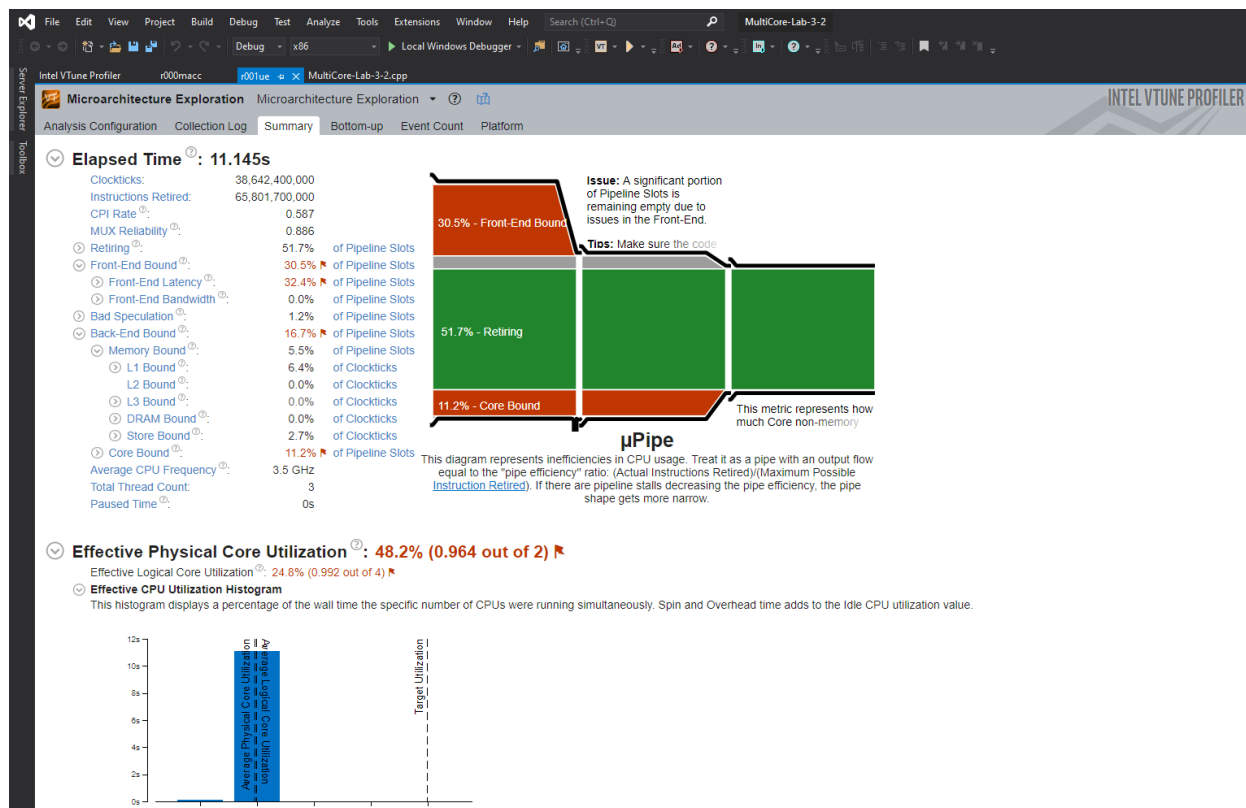
برای ۴۰۹۶:



برای ۸۱۹۲:

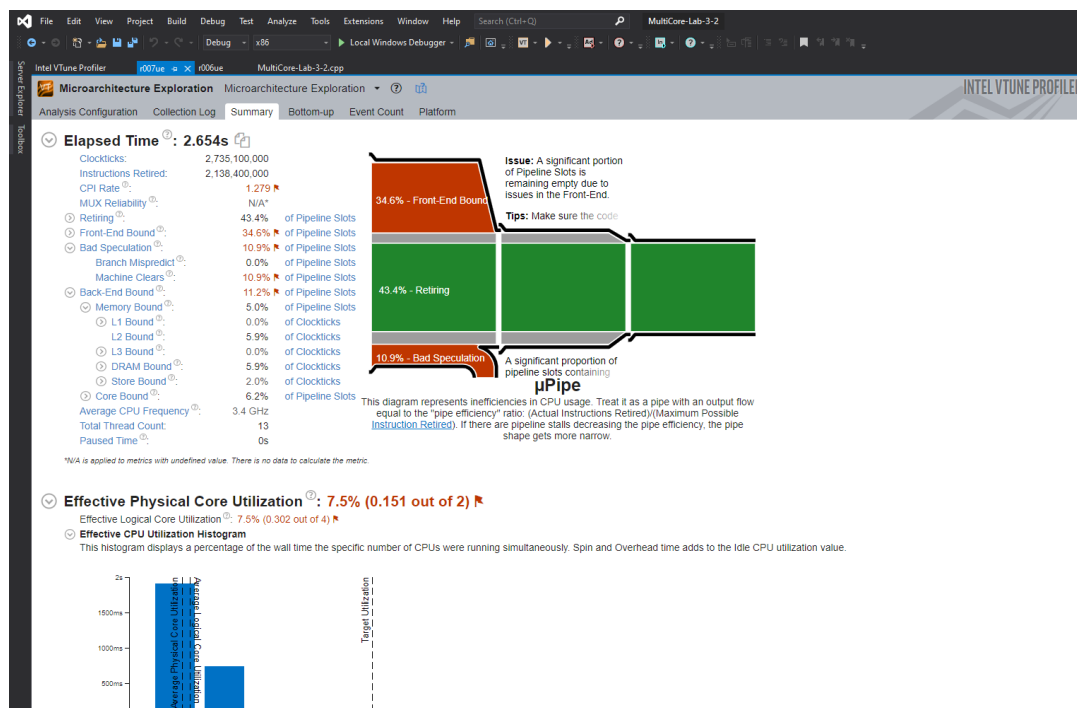


برای ۱۶۰۰۰:

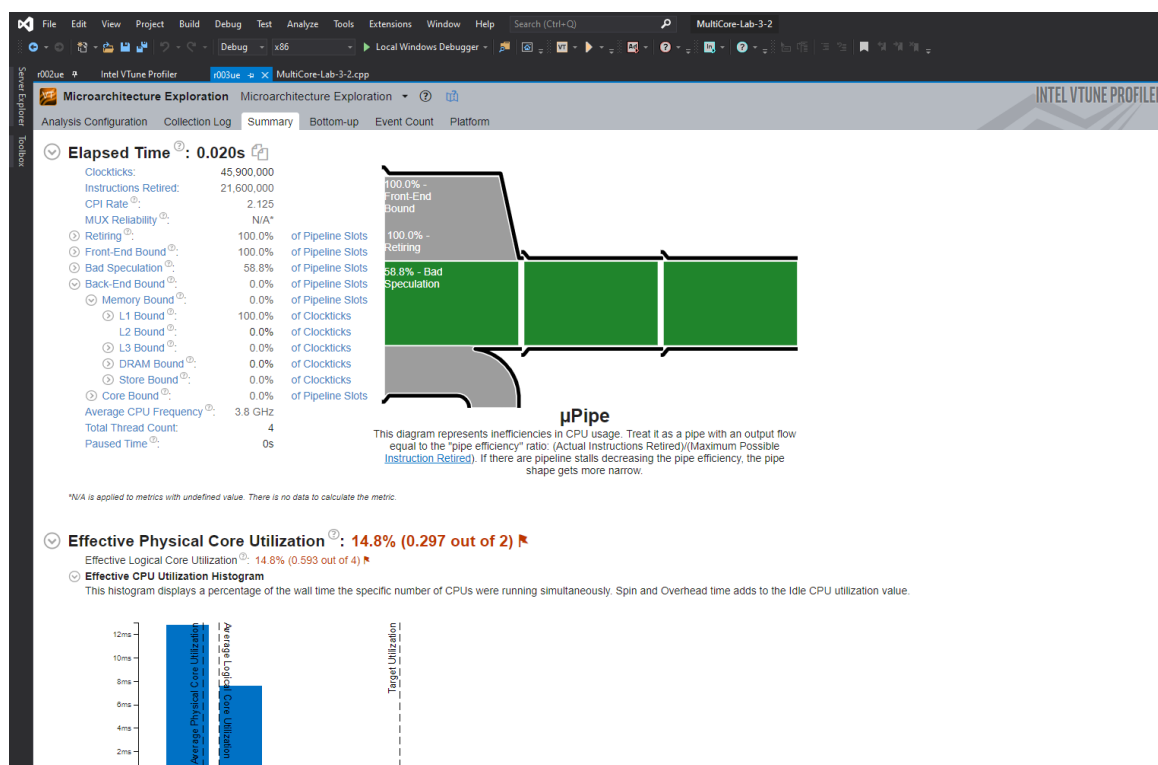


برای حالت ۲:

برای ۶۴:

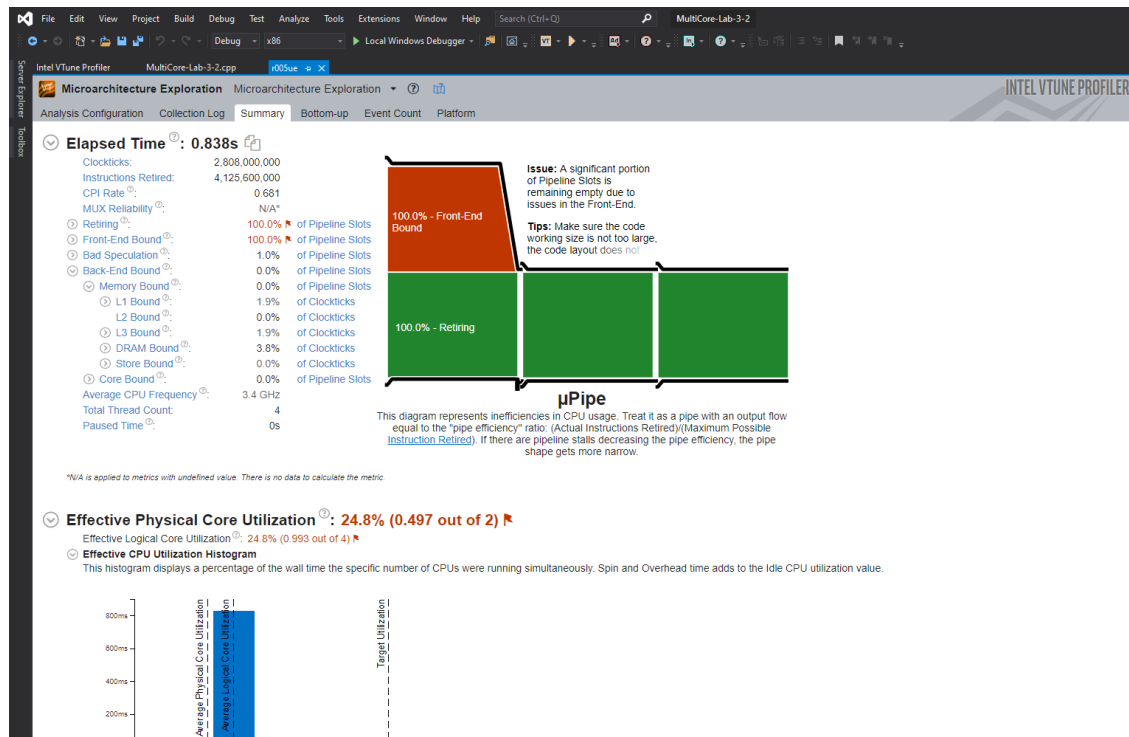


برای ۲۰۴۸:

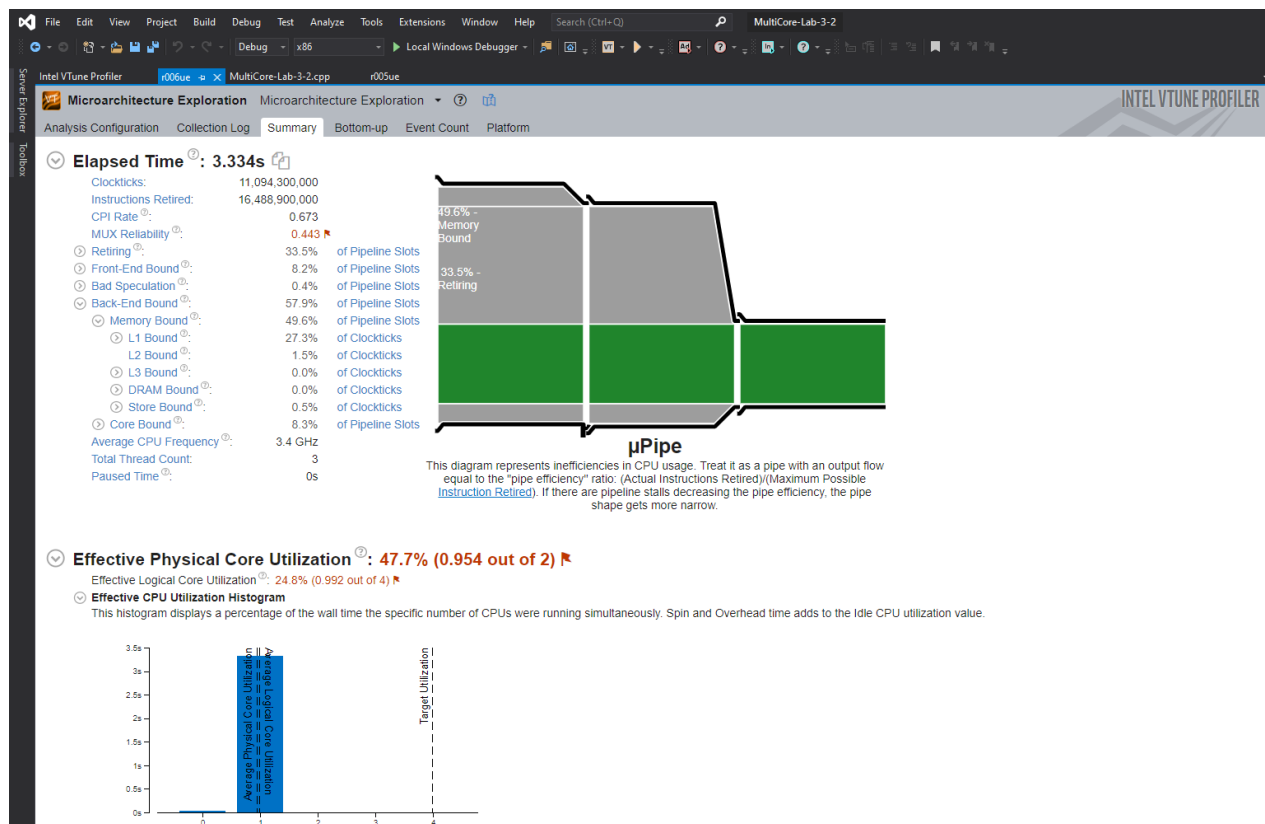




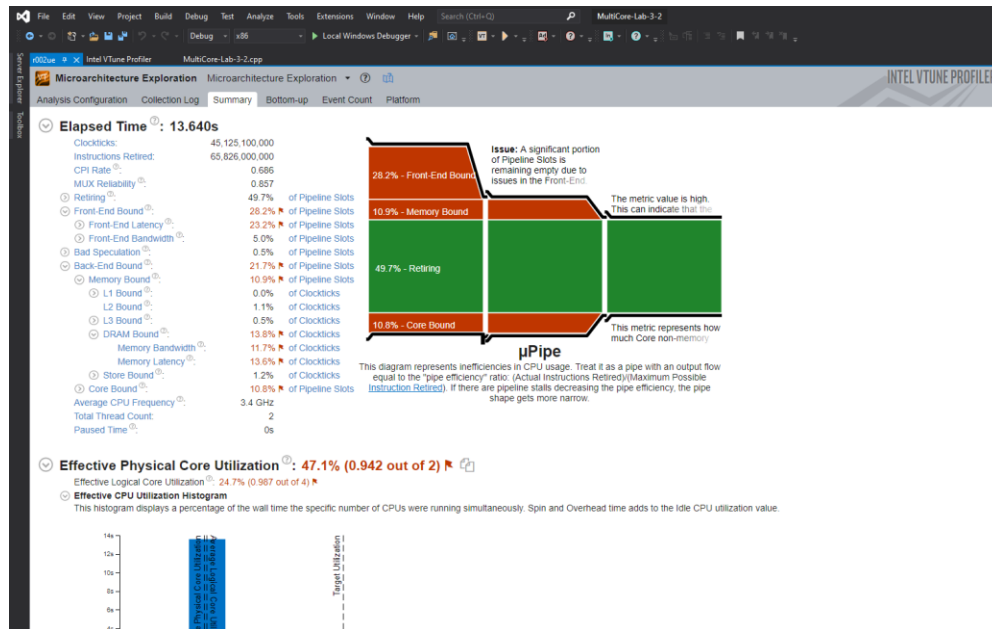
برای ۴۰۹۶:



برای ۸۱۹۲:



برای ۱۶۰۰۰:



هرچه قدر محاسبات ستونی تر و دیتای بزرگتری داشته باشیم affinity ما از کش L! به L2 به L3 به رم می‌رود، چون احتمال miss شدن و موجود نبودن دیتا در حافظه‌های کوچک‌تر بیش‌تر می‌شود.

در صورت کم‌تر بودن دیتای کش لاین نیز در سایز دیتاهای بزرگ‌تر زمان اجرا و شاخص‌های سطر و ستونی نزدیک هم می‌شوند، اما در دیتای با حجم پایین‌تر این امر کم‌تر مشهود است.