



I		5
2	가?	3
3	가?	6
4	가?	9
5	?	11
6		13





#### http://www.intel.com/products/processor\_number

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead., Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, VTune, Xeon Xeon Inside 7t Intel Corporation

가

Copyright © 2008, Intel Corporation.

Microsoft

. Microsoft Corporation

: 320487-001US





2				<i><b>7</b></i> <b>!</b> ?
		<install-dir></install-dir>	>/samples/matrix	
		. ®	가	
		가		(Hotspot)
			가	
	가			
:		가 .		

2.1

가

3



:

#### 1. Visual Studio

가	

R

 7i

 2.

 Image: C:\WINDOWS\system32\cmd.exe

 Image: C:\work\samples\matrix\_AD\matrix\Debug>matrix.exe

 Matrix transformation algorithms

 Number of Threads = 1

 Overall execution time =
 21.31 secs

 C:\work\samples\matrix\_AD\matrix\Debug>\_

### 2.2

:



: 320487-001US



r-008-hs matrix.cpp Start Page			-	×	Call Stack 🗸 🕂 🗙	t
🕾 🚯 Hotspots: Bottom-up 🚯 H	lotspots: Top-o	lown Tree		~	CPU time 🕑 4 1of 1 D	
Function - Call Stack	Module	CPU Time 🔻	*	^	100% of selection (18.960s of 18.960s) # matrix.exe!algorithm_3(struct MATRIX,int) - mat . matrix.exe!do_mm(void *) - matrix.cpp: 142	
algorithm_3(struct MATRIX,int)	matrix.exe	18.960s			kernel32.dl!GetModuleFileNameA+0x1b3	
Stack: <- do_mm(void *) <- GetMo	2 rix.exe	18.960s				
■algorithm_2(void)	matrix.exe	1.891s	1	=		
algorithm_1(void)	matrix.exe	0.760s	L		4	
RtEnterCriticalSection	ntdl.dl	0.1095	2			
⊞memset	MSVCR80D.dll	0.021s	-			
_RTC_CheckEsp	matrix.exe	0.016s	1	-11		
ra 11.0 Calestada	1.1		10.050-			
Selected:	<		> 10.9008	<u> </u>		
Filter: 100% is shown Module: <al></al>	~		終	₽	Solution Explorer 🔣 Summary 💽 Call Stack	

1	– (Function – Call Stack)
-	(+) . 가
2	
	CPU (CPU Time)
3	(Data of Interest) . CPU
	CPU 가 .
4	CPU .
	(Summary) .
5	CPU (CPU Time) CPU . (Elapsed Time)
J	

7ŀ?



## 2.3

### : (Hotspots: Bottom-up)

algorithm \_3 .

algorithm \_3

.

# . 222 가 가 CPU

가

·



#### 词 🖓 🧐 👲 🔲 4 3 Line 210 211 212 213 214 215 216 217 218 219 220 221 222 Source **CPU Time** \* for (int ii=0/ ii<30; ii++) { for (int i=myid; i<N; i==NumThreads) { for (int j=0; j<N; j++) { int ij = i=N + j; // Protect data.cc initialization from multiple thread contention Transformed contention(initialization from multiple thread contention) Transformed contention(initialization continue) } }</pre> 0.0156 // Protect data.cc initialization from multiple EnterCriticalSection(&initialization\_section); data.cc(i) = 0.0; LeaveCriticalSection(&initialization\_section); for (int k=0, k<N; k+=stride) { int ik = i+N + k; int k) = k+N + 0; data.cc(i) += data.as(k)\*data.bb(0); 0.0315 | 1.6856 -0.861s -1.0015 -223 224 225 226 227 2.202s 🥅 } 1 ) - } } } 228 229 230 // Cet floating point value for number of seconds since system started Total Selected: 13.1649

R

(Source)

1	. 가 CPU .
2	
3	
4	

.

가

.

.

.

3 7/? (Concurrency) 가 가 .



7

### 3.1

#### 가?(Concurrency – Where is my concurrency poor?) (Profile)

(Concurrency)

*7*}?

r-008-hs r-001-cc matrix.c	op Start Page		- ×	Summary	* ‡ ×
🕾 🚯 Concurrency: Bottom	up 🚷 Concurr	ency: Top-down Tree		Elapsed Time:	
	1	CPU Time by Utilization 🐨	* 🗐	CPU Time:	21.543s
- Call Stack	Module	Poor	图	Logical CPU Count:	2 ∞
Malgorithm_3(struct MATRIX,int)	matrix.exe	19.2946		p	
(#i algorithm_2(void)	matrix.exe	1.5635		2 /	
3 algorithm_1(vold)	matrix.exe	0.6938 🚦		1 ÷ /	
almemset.	MSVCR80D.dll	0.003s			
Selected:			19.2845	er o i	
< >	<		>	Smilteneous Running	liveati.
Filter: 100% is shown Module: <	el>	*	26 5	Solution Expl	Call Stack

.

(Concurrency) (Summary) 가 . CPU (CPU Time by Utilization) 가 . (Summary) 0 1 CPU .

: (Hotspots: Bottom-up) 가 algorithm\_3 · 가 가 . algorithm\_3 가 가 .

## 3.2

1. Visual Studio matrix.cpp .

- 2. 22 USE \_MULTIPLE \_THREADS TRUE
- 3. 23 USE \_MULTIPLE \_THREADS FALSE



4.	algorithm_3	2	216	218	Enter	LeaveCriticalSection
5.	가 0				. Visual Stud	io

### 3.3



21.31 14.01

🛎 🚯 Concurrency: Bottom-up	Concurre	ncy: Top-down Tree		w.	Elapsed Time:	
Function -Cell Stack	Madule	CPU Time by Utilizati	n.+ *⊡	*	68.2695 CPU Time: Logical CPU Count:	73.457s 2
# algorithm_3(struct MATRIX, mt)	matrix.exe	61.324			P	
EIOFastSystemCalRet	ntdi.di	9.375s			1 2 / N	
fialgorithm_2(void)	mabrix.exe	1,844 0			1 - A - A - A - A - A - A - A - A - A -	
E algorithm_1(void)	matrix, exe	0.7028		-		
#RtiLeaveCriticaSection	ntdl.dl	0.121s			3 /	
#_RTC_CheckEsp	matrix.exe	0.060s				
Selected:		(**** 1).	61.3246	×	In the second	
c )	<		>		Smultaneous Running Threads	





: (Locks and Waits: Bottom-up)

.

🔤 😪 Locks and Waits: Bottom	-up 🚷 Locks a	and Waits: Top	down Tree						$\nabla$
Wait Sync Object - Wait Function - Wait Call Stack - Signal Call Stack	Sync Object Type	Creation Module	Creation Source File	Cre Line	Мо	Wait Time by Utilization 💌	* 测	Wait Count by Utilization	>
Thread 0xa7e3525c	Thread	matrix.exe	matrix.cpp	113		85.249s			1
■Critical Section 0xccbb490b	Critical Section	matrix.exe	matrix.cpp	96		40.965s		19389	259
Stream 0xf9d187f1	Stream	MSVCR80D.dll	write.c	297		0.000s			4
Selected:							85.249s		1
Filter: 100% is shown Wait Module:	<all></all>	▼ Thread:	<all></all>		۷U	tilization: <all></all>	*	装	₽

.

가



<b>70</b> 4	) 🚸 🔔 🗏		
Line	Source	Wait Time by Utilization	☆ ≫ Wait Count by Utilization
115	printf("CreateThread %d failed %d\n",myid[i],GetL		
116	<pre>exit(1);</pre>		
117	}		
118	}		
119			
120	// Wait for all "algorithm_3" threads to finish		
121	int done = WaitForMultipleObjects(NumThreads, h, TRUE, IN	85.249s	85.2487
122			
123	// Display overall execution time		
124	<pre>double overall_end_time = GetSeconds();</pre>		
125	<pre>printf("Overall execution time = %10.2f secs\n",</pre>		
126	<pre>overall_end_time - overall_start_time);</pre>		
127			
128	DeleteCriticalSection (&initialization_section);		

	가		가
		가	
가			

.

.

### : (Locks and Waits: Bottom-up) (Critical Section)

.

Line	Source	Wait Time by Utilization 🔶 🕅	Wait Count by Utilization
210			
211	for (int ii=0; ii<30; ii++) {		
212	<pre>for (int i=myid; i<n; i+="NumThreads)" pre="" {<=""></n;></pre>		
213	<pre>for (int j=0; j<n; j++)="" pre="" {<=""></n;></pre>		
214	<pre>int ij = i*N + j;</pre>		
215	<pre>// Protect data.cc initialization from multip</pre>		
216	EnterCriticalSection(&initialization section)	40.965s	40.965
217	data.cc[i] = 0.0;		
218	LeaveCriticalSection(&initialization_section)		
219	for (int k=0; k <n; k+="stride)" td="" {<=""><td></td><td></td></n;>		
220	int $ik = i*N + k;$		
221	int kj = k*N + j;		
222	<pre>data.cc[i] += data.aa[k]*data.bb[j];</pre>		
223	}		

algorithm \_3

		. <u>data.cc</u> [ i] = 0 . 0;	i	i 가	
	for		.for		
i	for	i+=NumThreads			,

.



?

.

Visual Studio가 EnterCriticalSection(216)LeaveCriticalSection(218)

### 4.3

matrix.exe .	
C:\WINDOWS\system32\cmd.exe	- 🗆 ×
C:\work\samples\matrix_AD\matrix\Debug>matrix.exe Matrix transformation algorithms	
Using Multiple Threads Number of Threads = 2	
Overall execution time = 13.83 secs	
C:\work\samples\matrix_AD\matrix\Debug>	<b>•</b>
•	• //

.

14.01 13.83





	•



: (Concurrency: Bottom-up)

Function	No.	CPU Time by Utilizatio	n:Result 1	CPU Time by Utiliz	ation:Result 2	CPU Time by Utilizato	n:Difference 👻 [
	163	Poor Ok		Poor Ok		Poor Ok	
algorithm_3(struct MATRIX, int)		19.284		2.955	0	16.3298	
algorithm_1(void)		0.693s	-	0.688s		0.006s	<b>A</b> 1
menset	-	0.0036	1	Ûs d	2	0.0036	<b>9</b> 1
algorithm_2(void)		1.563\$	1	1.797s	D	-0.234s	1
Selected:			19.2846		2.955s		16.329
S	5						>
Filter: 100% is shown Module:	d>	V Three	d: cal>	Utilization:	ab	*	彩 🌣

.



: 320487-001US



1	가 Poor	matrix.exe	CPU	
2	가 Ok	matrix.	exe C	CPU
3	<cpu> = &lt; CPU . 16 .</cpu>	1 CPU > - < 2 CPU , algorithm_3	> 2 CPU	
4	. 2) CPU CPU (Logica . 4) C	: 1) (CPU Time) I CPUs utilized) :PU (Logical CPU count)	(Elapsed Time) CPU CPU	. 3)
5		(< 1 > - <	2 >)	
	matrix eve		( 2)	

illa CI -	LA.EAC			( 2)	
	(Ideal)		(	115%)	
algorithm _3		16			

# 6

가 ® . \_\_\_\_\_

®	Microsoft* Visual Studio	)* .	
		Visual Studio (Help) (Intel Parallel	
	Amplifier) >	(Intel	
	Parallel Amplifier He	. (ql	
	(Context	- sensitive Help)	
	F1	conclute ricip)	
フト	<pre><installdir>\documentation\<locale></locale></installdir></pre>		
	71	71	
		21	
	>F	•	
	가	Visual Studio	
	(Help)	(Intel	
	Parallel Amplifier)	(initial	
	7L (Somple C	ada Cuida)	
	✓r (Sample C	ode Guide)	
	ытмі	P	



	Windows* Parallel Studio) > Parallel Amplifier) > (Intel Parallel Amp	(Intel (Intel lifier Documentation)
®	®	フト
	• ®	・ ア・ ・
	• ®	С/С++
	• ®	
	Studio) > >	(Intel Parallel