Recitation 6: Cache Access Patterns

Andrew Faulring 15213 Section A 14 October 2002

Today's Plan

- Optimization
 - Amdahl's law
- · Cache Access Patterns
 - Practice problems 6.4, 6.15-17
- · Lab 4
 - Horner's Rule, including naïve code

Andrew Faulring

- faulring@cs.cmu.edu
- Office hours:
 - NSH 2504 (lab) / 2507 (conference room)
 - Wednesday 5-6
- Lab 4
 - due Thursday, 24 Oct @ 11:59pm

Amdahl's law

Old program (unenhanced)

Old time: $T = T_1 + T_2$

T₂ = time that can be enhanced.

be enhanced.

 T_1 = time that can NOT

New program (enhanced)

 $T_1' = T_1$ $T_2' <= T_2$ New time: $T' = T_1' + T_2'$ T₂' = time after the enhancement.

Speedup: $S_{overall} = T / T'$

Key idea: Amdahl's law quantifies the general notion of diminishing returns. It applies to any activity, not just computer programs.

Example: Amdahl's law

 You plan to visit a friend in Normandy France and must decide whether it is worth it to take the Concorde SST (\$3,100) or a 747 (\$1,021) from NY to Paris, assuming it will take 4 hours Pgh to NY and 4 hours Paris to Normandy.

```
time NY->Paris total trip time speedup over 747 8.5 hours 16.5 hours 1 SST 3.75 hours 11.75 hours 1.4
```

 Taking the SST (which is 2.2 times faster) speeds up the overall trip by only a factor of 1.4!

Locality

- Temporal locality: a memory location that is referenced once is likely to be reference again multiple times in the near future
- Spatial locality: if a memory location is referenced once, then the program is likely to reference a nearby memory location in the near future

Amdahl's law (cont)

 Trip example: Suppose that for the New York to Paris leg, we now consider the possibility of taking a rocket ship (15 minutes) or a handy rip in the fabric of space-time (0 minutes):

```
time NY->Paris total trip time speedup over 747
747 8.5 hours 16.5 hours 1
SST 3.75 hours 11.75 hours 1.4
rocket 0.25 hours 8.25 hours 2.0
rip 0.0 hours 8 hours 2.1
```

Moral: It is hard to speed up a program.

Moral++: It is easy to make premature optimizations.

Practice Problem 6.4

```
int summary3d(int a[N][N][N])
{
    int i, j, k, sum = 0;
    for (i = 0; i < N; i++) {
        for (j = 0; k < N; j++ ) {
            for (k = 0; k < N; k++ ) {
                 sum += a[k][i][j];
            }
        }
    }
    return sum;
}</pre>
```

Answer

```
int summary3d(int a[N][N][N])
{
    int i, j, k, sum = 0;
    for (k = 0; k < N; k++) {
        for (i = 0; i < N; i++ ) {
            for (j = 0; j < N; j++ ) {
                 sum += a[k][i][j];
                 }
        }
    }
    return sum;
}</pre>
```

Cache Access Patterns

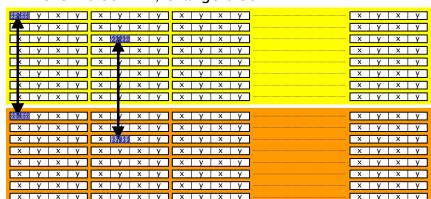
- Spend the next fifteen minutes working on Practice Problems 6.15-17
- Handout is a photocopy from the text

Practice Problem 6.15-17

- sizeof(algae position) = 8
- Each block (16 bytes) holds two algae_position structures
- The 16×16 array requires 2048 bytes of memory
 - Twice the size of the 1024 byte cache

Practice Problem 6.15-17

- Rows: 16 items (8 blocks, 128 bytes)
- Columns: 16 items
- Yellow block: 1k; Orange block 1k



6.15: Row major access pattern

6.15: Stride of 2 words

First loop, accessing just x's



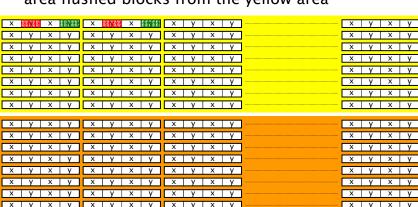
6.15: Stride of 2 words

First loop, accessing just x's

88888	у	88288	У	000000	у	88288	у	38838	у	33333	У		383381	у	882881	У
aaaaa	У	RRCRRI	У	555551	У	BBCBBI	у	2000年1	У	28233	У		335333	У	883881	У
464461	У	88588	У	33233	У	187,381	У	18 18 18 18 18 18 18 18 18 18 18 18 18 1	У	88288	У		1212 1212	У	88088	У
	У	445441	У	33333	У	88288	у	33233	у	11:11	у		11111	У	227.22	У
11111	У	**0**	У	*****	У	20000	у	11:11:	У	88288	у		88288	У	22022	У
£££££	у	55(55)	У	888881	У	88588	у	SESSE	У	22.22	у		888881	У	22222	У
188288	У	888881	у	RRRRR	У	RRERRI	у	888888	У	122222	у		888881	У	88888(У
882881	У	RRRRR	У	BRRES	У	88888(У	BBBBBB	У	100000	У		RECER	У	33,33	У
33733	У	33733	У	257.55	У	55755	у	38038	У	20728	У		25000	У	33233	У
\$\$255	У	33233	У	22.22	У	35735	у	38088	У	88488 88488	У		ie ie ie ie ie ie ie ie ie ie	У	22222	У
35255	У	55055	у	86066	У	22.22	У	20220	У	44444	У		00000	У	22.22	У
888881	У	\$\$2\$\$!	У	200001	У	287.48	у	88388	У	新野产的原; 新野市中野縣	У		000001	У	992991	У
888881	У		У	88288	У	2121212121 212121212121	у	20222	У	55255	у		66666	У	88388	У
88288	У	335 335 B	у	22223	У	44544	у	88888	у	88388	у		88888	У	44.44	У
88288	У	建建设建建	У	333333	У	34534	У	22223	У	88388	У		22722	У	\$\$33\$	У
66666	·/	884880	٧/	999991	- 1/	SESENCE SES	V	WWWWW	- 1/	(gr gr (r gr gr	- 1/	l	20020	٧/	\$0.50 to \$0.50 \$	- 1/

- Second loop, accessing just the y's
- Same miss pattern because accessing the orange area flushed blocks from the yellow area

6.15: Stride of 2 words



6.15: Stride of 2 words

- Second loop, accessing just the y's
- Same miss pattern because accessing the orange area flushed blocks from the yellow area

Х	X	88788	Х	26768	Х	186788	Х	88788	Х	88788	 Х	44044	Х	REVER
Х	X	88788	Х	88888	Х	88788	Х	222221	Х	\$ \$ (* \$ fs.	 Х	22722	Х	88,788
Х	X	88488	Х	22722	Х	127.22	Х		Х		 Х	22722	Х	88788
Х	X	\$\$7.5\$	Х	\$\$7\$\$	Х	\$\$7.\$\$	Х	は 本 が な な な な な な な な な な な な な な な な な な	Х	K K 4.8 K	 Х	55755	Х	\$\$\\\\$\$
Х	X	22722	Х	\$\$2\$\$1	х	\$\$\\\$\$	Х	586581	х	SEPE	 X	20000	Х	88788
Х	X	55.056	Х	KETEE KELEE	Х	22022	Х	000000	х	28,088	 Х	88888	Х	88788
Х	X	20000	Х	Seves	Х	88788	Х	882881	Х	66,788	X	999991	Х	00300
Х	X X	28.28	Х	36666	х					99039:	×	909091	х	gayaa
	10000000	200000		- CALMARIA		<u> </u>		- CALMICAL D		i de la constante de la consta				to the latest and the
Х	X X	88788 88788	Х	88888	Х	SERES	Х	44744	Х	38738	Х	11211	Х	88788
Х	X X	22,22	Х	55555	Х	\$\$\\$\$.	Х	44044	Х	33738	 Х	##X##	Х	£57.55
Х	X X	22.22	Х	#####	Х	38788	Х	\$ \$ \$ \$ \$ \$ \$ \$	Х	22222	Х	\$\$7.5\$1	Х	88788
Х	X	00000	Х	E E V E E	Х	38138	Х	88788	Х	88088	X	RRVER	Х	88788
Х	X	22722	Х	BARBA	Х	88088	Х	888888	Х	22722	X	RECER	Х	BBZBB
Х	X	88788	Х	88888	Х	BENER	Х	000000	Х	22720	X	MANAGE	Х	88788
Х	X	22723	Х		Х	HNI	Х			22/22	X	ERRERE	Х	88788
Х	X			933333			_	SSESSI				88788		
	and the second s			. oxovátotok				constitution.		- Allelen		on contribution		

Answers to 6.15

- A: 512
- 2 for each of 256 array elements
- B: 256
 - Every other array element experiences a miss
- · C: 50%

Column major access pattern

New access removes first cache line contents before its were used



Column major access pattern

New access removes first cache line contents before its

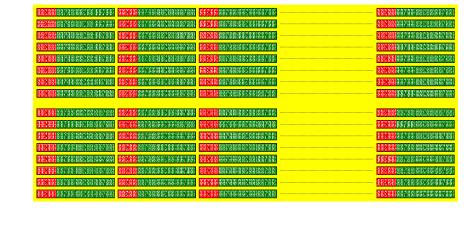
were asea				
anenniaay aalangaalaayaa	22222111111222211111111	RECREIREDEN BEZORIGADAS		Mastas edemestas
MANAMAN AN MANAMAN AN A		00000 886 886 880 880 880 880		A CARACAR BRINKING COE
BOCOBINANAN BARBARANAN	SS SS NATIONAL SAN SERVICES	55155155155155156156156		SE SAJAS BELGE 33733
######################################	\$\$5\$\$ BBDBB \$\$7\$\$ \$\$7\$\$	\$		SE SE SE MANAGER SE SE SE
######################################	22222 2222 2222 2222 2222 2222 2222 2222	22722		in er
	88288	eecesierpariancanina vari	<u></u>	erike de secretaria
88788 SS. SS 88588 SS. SS	RACOR SELECTED SELECTION	88288 887388888888888		88 23 33 <mark>35 25 </mark> 33 33
	882881007001 88288 FE7FF			arien er
TO A STREET OF THE STREET OF T		222220000000000000000000000000000000000		Gurrarisezeeix s.xx
PARTON DE LES ESTADE	BECOM THAT ESSENDED FOR I	88388 4 4 4 4 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6		
\$8088 \$8088 \$8788 \$ \$0 48	SECONDENS SECOND	######################################		eribbyrbij <mark>azaa</mark> ibbyrb
38288WEXENSESS \$5056	33533300000000000000000000000000000000	\$2222 BENEFICE (\$222)	889	arieer eelwoode er e
SECONISCIPATION TO TRANSPORT	BECSE NANANIE ECE E ISBURSI	30033 <u>BENESINON NAMARA</u>		SSICE CONTRACTOR
NECESION CONTROL OF SECTION SE	337333333333366766766700001	2272238878878878874444	NAT THE REST	Manadada in Amarika da Amarika d

Answers to 6.16

- A: 512
- B: 256
- · C: 50%

Column major access pattern

No misses on second access to each block, because the entire array fits in the cache.

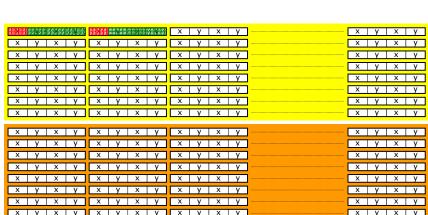


Answers to 6.16

- A: 512
- B: 256
- · C: 50%
- D: 25%

Stride of 1 word

Access both x and y in row major order



Stride of 1 word

Access both x and y in row major order

22722 22722 22722	NELSE SELECTION NO.	33238	
	BELER CONTRACTOR		88788 887 887 887 888 887 888 887 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
1000 10	er er artarissississississississississississississ	20220 1171 1272 1272 1272 1272 127	
	8808812172122122133733	AND	00000 SS SS 11 11 15 1 1 1 1 1 1 1 1 1 1 1 1
6201	## PE PE PE PE PE PE PE	2222 11711 22 22 22 22 22	
HICHHOOCO FFOFFEFEE	33233 8878888188188788	<mark>aacaa</mark> (BB?BB:BBCBB:RBCBK)	\$\$188 \$\$18818818818819
er en	86786 (88788)28738138738	BROOM PATTACION NO MATERIA	eeree navaaleeree nava
			99299333341315555
	20720 22.22.33.33.93.93	2020	
	30000 (KK) KK (AN) (AN) (AN) (AN)	2222	
	88288 88288 111 111 113 133 133		
	8808812222333333333333333333333333333333	Maria de Participa de Caracida	
S S S S S S S S S S S S S S S S S S S	00000100.00111111133733	BESSELVI VI	
		800000000000000000000000000000000000000	22221 XX
	\$8038 PATROLEE (#155755)	BECCE	
	20002 88. 88. 44. 44. 44. 53. 53.		\$\$255 dd. dd. 22755 (\$775)

Lab 4: Horner's Rule

Polynomial of degree d (d+1 coefficients)

$$P(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_d x^d$$

$$P(x)=a_0+(a_1+(a_2+(\cdots+(a_{d-1}+a_dx)x\cdots)x)x)x$$

Answers to 6.17

- A: 512
- B: 128
 - All are compulsory misses
- · C: 25%
- · D: 25%
 - Cache size does not matter since all misses are compulsory
 - Though the block size does matter

Naïve code for Horner's Rule

```
/* Horner's rule */
int poly_evalh(int *a, int degree, int x)
{
   int result = a[degree];
   int i;
   for (i = degree-1; i >= 0; i--)
      result = result*x+a[i];
   return result;
}
```