

Lab 1 Java

1. Sequential run

No.	N	M	k	Time (nano)
1	10	10	3	18220
2	1000	1000	5	37004220
3	10	10000	5	2133450
4	10000	10	5	2458520
5	10000	10000	5	3714556210

2. Paralel Horizontal run

No.	N	M	k	p	Time
1	10	10	3	4	663120
2	1000	1000	5	2	22880960
3	1000	1000	5	4	10611820
4	1000	1000	5	8	9386820
5	1000	1000	5	16	10423540
6	10	10000	5	2	259400
7	10	10000	5	4	432590
8	10	10000	5	8	744950
9	10	10000	5	16	1443780
10	10000	10	5	2	2624470
11	10000	10	5	4	1110040
12	10000	10	5	8	1328410
13	10000	10	5	16	2086380
14	10000	10000	5	2	2078448270
15	10000	10000	5	4	1636274540
16	10000	10000	5	8	1410654510
17	10000	10000	5	16	1400457260

3. Paralel Vertical run

No.	N	M	k	p	Time
1	10	10	3	4	703160
2	1000	1000	5	2	21364900
3	1000	1000	5	4	11112900
4	1000	1000	5	8	11971780
5	1000	1000	5	16	12154860
6	10	10000	5	2	1304900
7	10	10000	5	4	1151100
8	10	10000	5	8	1165960
9	10	10000	5	16	1757600
10	10000	10	5	2	271520
11	10000	10	5	4	429910
12	10000	10	5	8	725080
13	10000	10	5	16	1491470
14	10000	10000	5	2	2332650150
15	10000	10000	5	4	1615137880
16	10000	10000	5	8	1480072180
17	10000	10000	5	16	1487302450

C++

1. Static sequential run

No.	N	M	k	Time
1	10	10	3	4780
2	1000	1000	5	101552370
3	10	10000	5	5754570
4	10000	10	5	8015850
5	10000	10000	5	7201215300

2. Dynamic sequential run

No.	N	M	k	Time
1	10	10	3	4830
2	1000	1000	5	102661270
3	10	10000	5	5963480
4	10000	10	5	7905970
5	10000	10000	5	7140726400

3. Static paralel horizontal run

No.	N	M	k	p	Time
1	10	10	3	4	3102450
2	1000	1000	5	2	329183100
3	1000	1000	5	4	525489450
4	1000	1000	5	8	688697610
5	1000	1000	5	16	860680380
6	10	10000	5	2	30565210
7	10	10000	5	4	54136880
8	10	10000	5	8	78887650
9	10	10000	5	16	131996770
10	10000	10	5	2	26686810
11	10000	10	5	4	44988420
12	10000	10	5	8	64072750
13	10000	10	5	16	88802340
14	10000	10000	5	2	26816662740
15	10000	10000	5	4	47071927350
16	10000	10000	5	8	64837434590
17	10000	10000	5	16	81818178500

4. Static paralel vertical run

No.	N	M	k	p	Time
1	10	10	3	4	3270130
2	1000	1000	5	2	324490680
3	1000	1000	5	4	523229330
4	1000	1000	5	8	695730880
5	1000	1000	5	16	948401590
6	10	10000	5	2	38851370
7	10	10000	5	4	62676250
8	10	10000	5	8	86197320
9	10	10000	5	16	113926850
10	10000	10	5	2	40388340
11	10000	10	5	4	100650660
12	10000	10	5	8	178275830
13	10000	10	5	16	303616400
14	10000	10000	5	2	31512232780
15	10000	10000	5	4	54890889200
16	10000	10000	5	8	73294157300
17	10000	10000	5	16	92328666460

5. Dynamic paralel horizontal run

No.	N	M	k	p	Time
1	10	10	3	4	4107020
2	1000	1000	5	2	243749880
3	1000	1000	5	4	341485390
4	1000	1000	5	8	443160750
5	1000	1000	5	16	538666180
6	10	10000	5	2	16020740

No.	N	M	k	p	Time
7	10	10000	5	4	31485080
8	10	10000	5	8	49049550
9	10	10000	5	16	85258340
10	10000	10	5	2	20566450
11	10000	10	5	4	37513820
12	10000	10	5	8	54224170
13	10000	10	5	16	77550850
14	10000	10000	5	2	13187906290
15	10000	10000	5	4	23109035470
16	10000	10000	5	8	31528821880
17	10000	10000	5	16	39530570510

6. Dynamic paralel vertical run

No.	N	M	k	p	Time
1	10	10	3	4	3846310
2	1000	1000	5	2	207266550
3	1000	1000	5	4	316607130
4	1000	1000	5	8	432649690
5	1000	1000	5	16	547559240
6	10	10000	5	2	19353640
7	10	10000	5	4	33291130
8	10	10000	5	8	49722260
9	10	10000	5	16	80444140
10	10000	10	5	2	23734640
11	10000	10	5	4	44211350
12	10000	10	5	8	80038010
13	10000	10	5	16	122756470
14	10000	10000	5	2	20599673640

No.	N	M	k	p	Time
15	10000	10000	5	4	35276428180
16	10000	10000	5	8	46566492490
17	10000	10000	5	16	57578150070

Concluzii

Java

Executia paralela este mai rapida pentru cazurile 1000x1000 sau 10000x10000. Depinde de dimensiuni, cea verticala este mai rapida decat cea orizontala. De asemenea, cresterea paralelismului scade timpul necesar, pana intr un anumit prag. Se poate observa ca pentru diferite modalitati de rulare, si diferiti parametri, exista un numar optim de thread-uri. La parametrii mici, paralelizarea influenteaza negativ timpul de executie.

C++

Executile secventiale statice/dinamice au timp similari de executie, dar in mod paralel, cea dinamica aduce cele mai bune rezultate. Paralelizarea nu are efectul dorit, timpul creste odata cu numarul de thread-uri alocat. Performanta scade odata cu cresterea parametrilor. Pentru paralelism, versiunile dinamice sunt mai rapide. Costul de initializare poate impacta negativ timpul de executie.

Java vs C++

Timpii de executie sunt mai buni pentru java. Spre deosebire de c++, in java paralelismul isi arata efectul odata cu cresterea numarului de thread-uri.

Orizental vs Vertical

Diferenta depinde de la caz la caz, deoarece thread-urile sunt imprevizibile, conteaza mult si forma matricii si a kernelului.

Paralelizarea aduce beneficii timpului de executie doar pana la o anumita dimensiune si numar de threaduri. Se pot observa **sweet spot-uri** pentru unele dimensiuni. De exemplu: pentru 10000x10000 in java cu k=5x5, cel mai mic timp de executie este cu p = 8, pe vertical; dar pe orizental, pentru aceleasi specificatii, este cu p=16.