Programming Techniques for Supercomputers Tutorial

Erlangen National High Performance Computing Center

Department of Computer Science

FAU Erlangen-Nürnberg

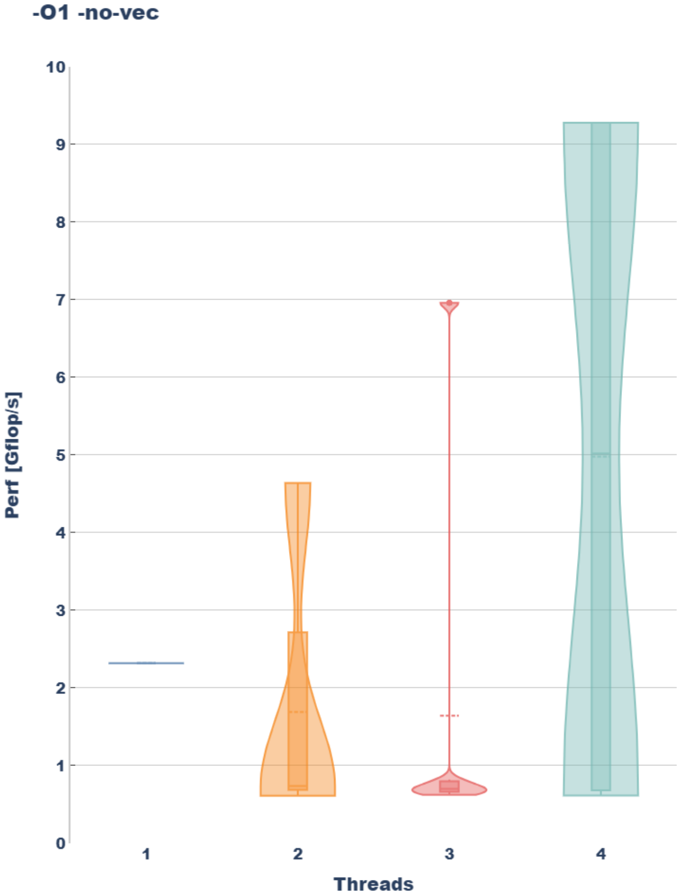
Sommersemester 2024

Assignment 7 – Task 1a)

Weird π

```
#pragma omp parallel for reduction(+:sum)
for(int i=0; i<N; i++) {
  x = (i + 0.5) * delta;
  sum = sum + 4.0 * sqrt(1.0 - x * x);
}
```

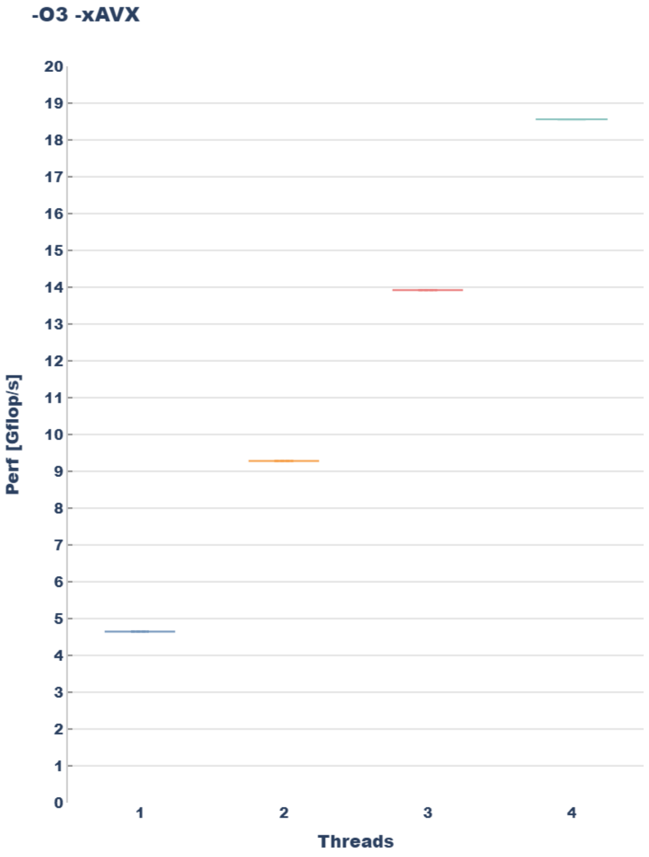
threads	Min	Lower Q	Mean	Upper Q	Max
1	2.32	2.32	2.32	2.32	2.32
2	0.61	0.69	1.69	2.72	4.64
3	0.62	0.67	1.64	0.80	6.96
4	0.62	0.68	4.98	9.28	9.28



Assignment 7 – Task 1b)

Weird π

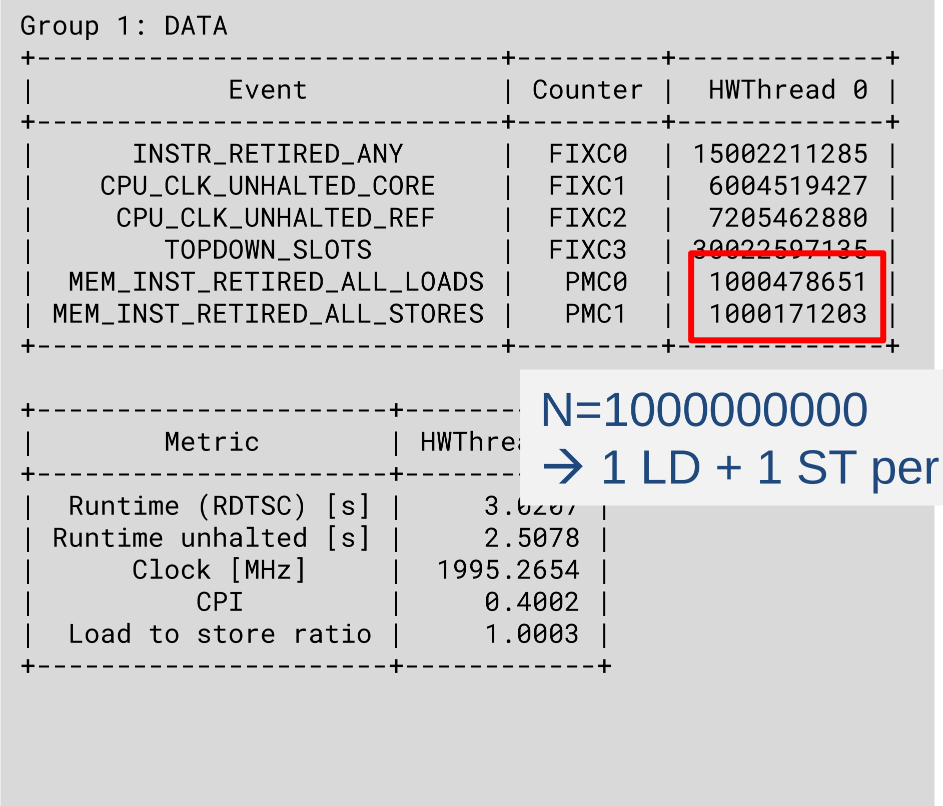
```
#pragma omp parallel for reduction(+:sum)
for(int i=0; i<N; i++) {
  x = (i + 0.5) * delta;
  sum = sum + 4.0 * sqrt(1.0 - x * x);
}
```



threads	Min	Lower Q	Mean	Upper Q	Max
1	4.65	4.65	4.65	4.65	4.65
2	9.28	9.28	9.28	9.28	9.28
3	13.92	13.92	13.92	13.92	13.92
4	18.56	18.56	18.56	18.56	18.56

Assignment 7 – Task 1c)

-O1 -no-vec

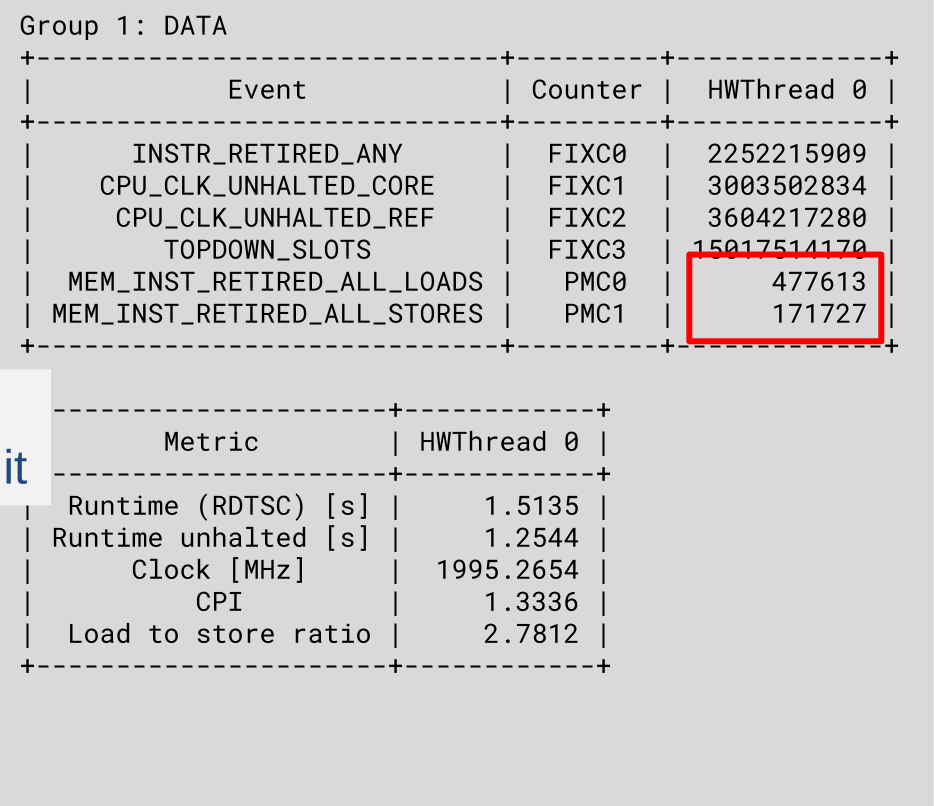


```
Group 1: DATA
+------------------------------+---------+--------------+
|            Event             | Counter |  HWThread 0  |
+------------------------------+---------+--------------+
|       INSTR_RETIRED_ANY      |  FIXC0  |  15002211285 |
|     CPU_CLK_UNHALTED_CORE    |  FIXC1  |   6004519427 |
|      CPU_CLK_UNHALTED_REF    |  FIXC2  |   7205462880 |
|         TOPDOWN_SLOTS        |  FIXC3  |  30022597135 |
|  MEM_INST_RETIRED_ALL_LOADS  |   PMC0  |   1000478651 |
| MEM_INST_RETIRED_ALL_STORES  |   PMC1  |   1000171203 |
+------------------------------+---------+--------------+

+----------------------+------------+
|        Metric        | HWThread 0 |
+----------------------+------------+
|  Runtime (RDTSC) [s] |     3.0207 |
| Runtime unhalted [s] |     2.5078 |
|      Clock [MHz]     |  1995.2654 |
|          CPI         |     0.4002 |
|  Load to store ratio |     1.0003 |
+----------------------+------------+
```

N=1000000000

→ 1 LD + 1 ST per it

-O3 -xAVX



```
Group 1: DATA
+------------------------------+---------+--------------+
|            Event             | Counter |  HWThread 0  |
+------------------------------+---------+--------------+
|       INSTR_RETIRED_ANY      |  FIXC0  |   2252215909 |
|     CPU_CLK_UNHALTED_CORE    |  FIXC1  |   3003502834 |
|      CPU_CLK_UNHALTED_REF    |  FIXC2  |   3604217280 |
|         TOPDOWN_SLOTS        |  FIXC3  |  15017514170 |
|  MEM_INST_RETIRED_ALL_LOADS  |   PMC0  |       477613 |
| MEM_INST_RETIRED_ALL_STORES  |   PMC1  |       171727 |
+------------------------------+---------+--------------+

+----------------------+------------+
|        Metric        | HWThread 0 |
+----------------------+------------+
|  Runtime (RDTSC) [s] |     1.5135 |
| Runtime unhalted [s] |     1.2544 |
|      Clock [MHz]     |  1995.2654 |
|          CPI         |     1.3336 |
|  Load to store ratio |     2.7812 |
+----------------------+------------+
```

Assignment 7 – Task 1d)

-O3 -xAVX

```
.LBB4_12:
  leal                (%rdx,%r8), %r9d
  vpbroadcastd        %r9d, %ymm7
  vpaddd              %ymm1, %ymm7, %ymm8
  vextracti128        $1, %ymm8, %xmm7
  vcvtdq2pd           %xmm7, %ymm7
  vcvtdq2pd           %xmm8, %ymm8
  vfmadd213pd         %ymm3, %ymm2, %ymm8
  vfmadd213pd         %ymm3, %ymm2, %ymm7
  vmovapd             %ymm7, %ymm9
  vfnmsub213pd        %ymm4, %ymm7, %ymm9
  vfnmsub213pd        %ymm4, %ymm8, %ymm8
  vsqrtpd             %ymm8, %ymm8
  vsqrtpd             %ymm9, %ymm9
  vfmadd231pd         %ymm9, %ymm5, %ymm6
  vfmadd231pd         %ymm8, %ymm5, %ymm0
  addl                $8, %r8d
  cmpl                %edi, %r8d
  jle .LBB4_12
# ...
# horizontal ADD
# and remainder loop
```

calculate i

```
#pragma omp parallel for reduction(+:sum)
for(int i=0; i<N; i++) {
  x = (i + 0.5) * delta;
  sum = sum + 4.0 * sqrt(1.0 - x * x);
}
```

Assignment 7 – Task 1d)

-O1 –no-vec

```
.LBB4_3:
  xorps      %xmm4, %xmm4
  cvtsi2sd   %eax, %xmm4
  addsd      %xmm1, %xmm4
  mulsd      (%r15), %xmm4
  movsd      %xmm4, (%r14)
  mulsd      %xmm4, %xmm4
  movapd     %xmm2, %xmm5
  subsd      %xmm4, %xmm5
  xorps      %xmm4, %xmm4
  sqrtsd     %xmm5, %xmm4
  mulsd      %xmm3, %xmm4
  addsd      %xmm4, %xmm0
  incl       %eax
  cmpl       %eax, %ecx
  jne .LBB4_3
```

calculate i

```
#pragma omp parallel for reduction(+:sum)
for(int i=0; i<N; i++) {
  x = (i + 0.5) * delta;
  sum = sum + 4.0 * sqrt(1.0 - x * x);
}
```

Each iteration, we

- load **delta** and
- store **x**

Assignment 7 – Task 2a)

Ressource-driven modeling

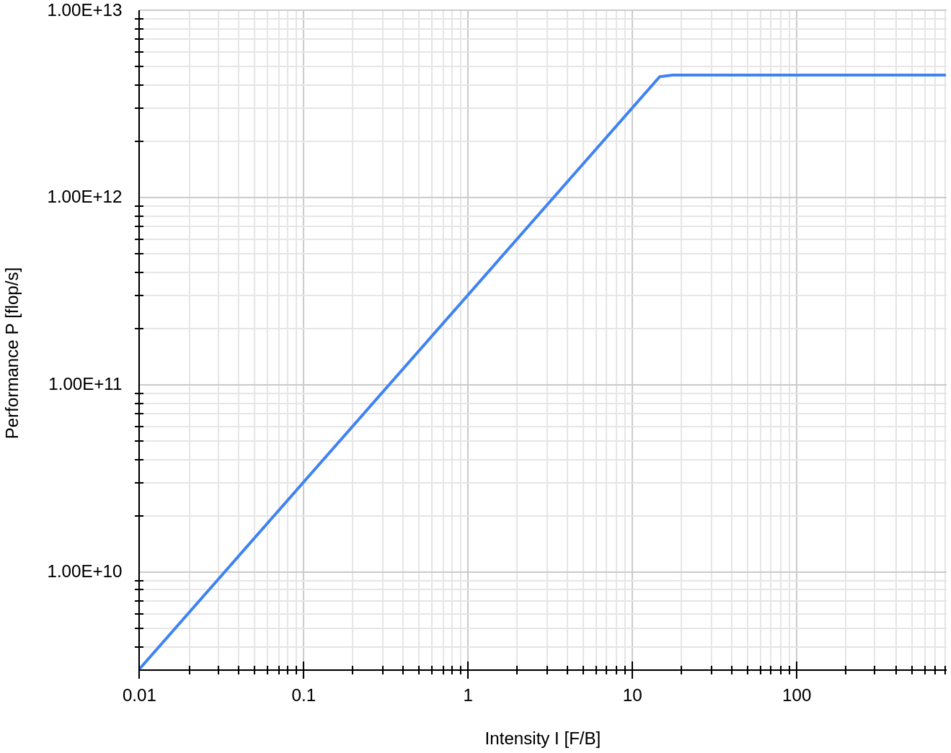
$$P_{peak} = 4.5\,\frac{TFlop}{s} \qquad b_s = 300\,\frac{GB}{s}$$

$$P = \min(P_{peak}, I \times b_s)$$

“Knee” at

$$P_{peak} = I_{crit} \times b_s$$

$$\rightarrow I_{crit} = \frac{P_{peak}}{b_s} = 15\,\frac{flop}{B}$$



Assignment 7 – Task 2b), c)

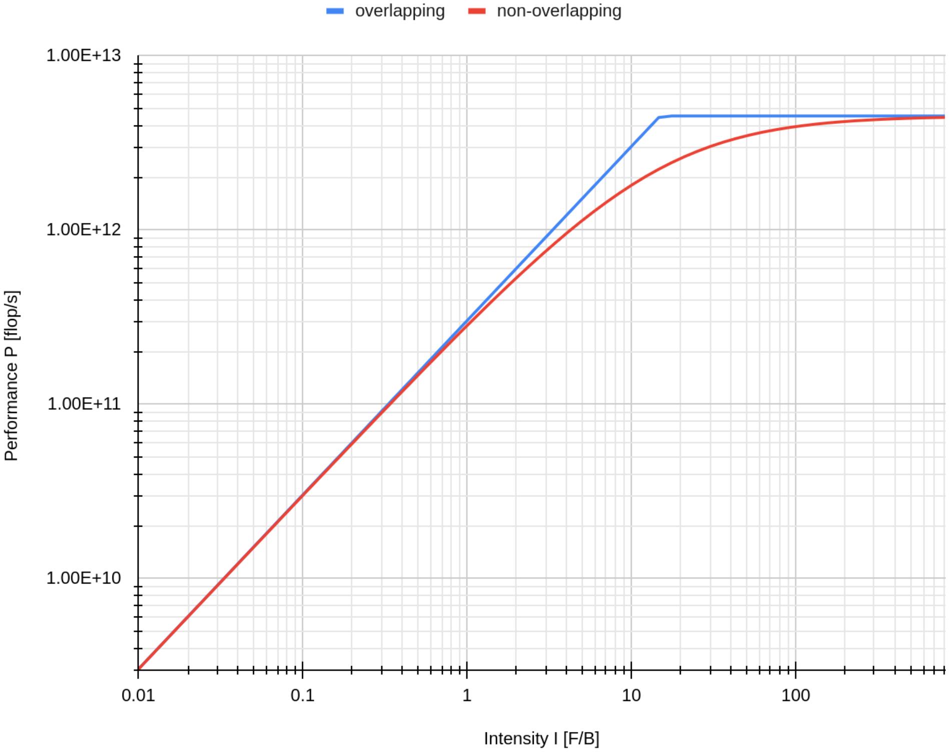
Ressource-driven modeling

$$P_{peak} = 4.5\,\frac{TFlop}{s} \qquad b_s = 300\,\frac{GB}{s}$$

$$P_{nOL} = \frac{W}{T_{flop} + T_{data}}$$

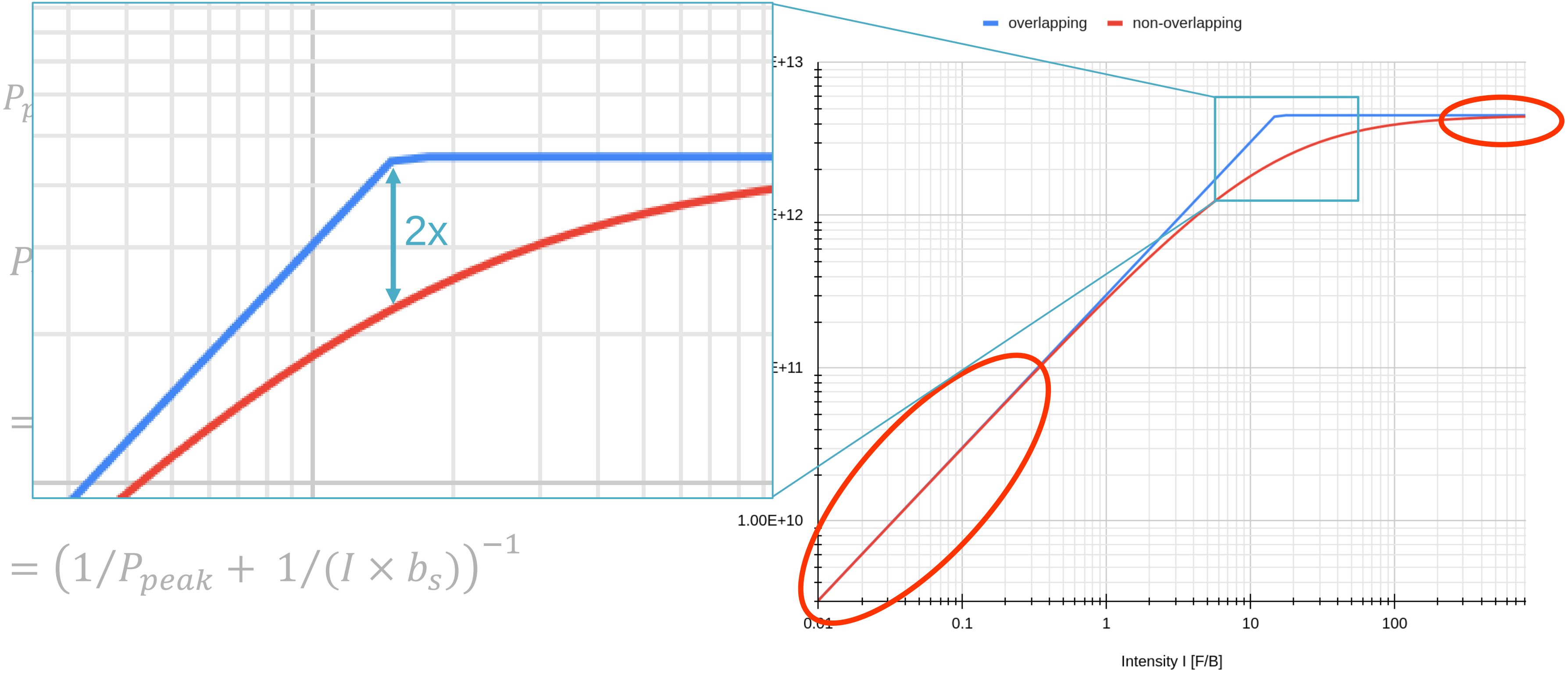
$$= \frac{W}{W/P_{peak} + V/b_s}$$

$$= \left(1/P_{peak} + 1/(I \times b_s)\right)^{-1}$$



Assignment 7 – Task 2b), c)

Ressource-driven modeling

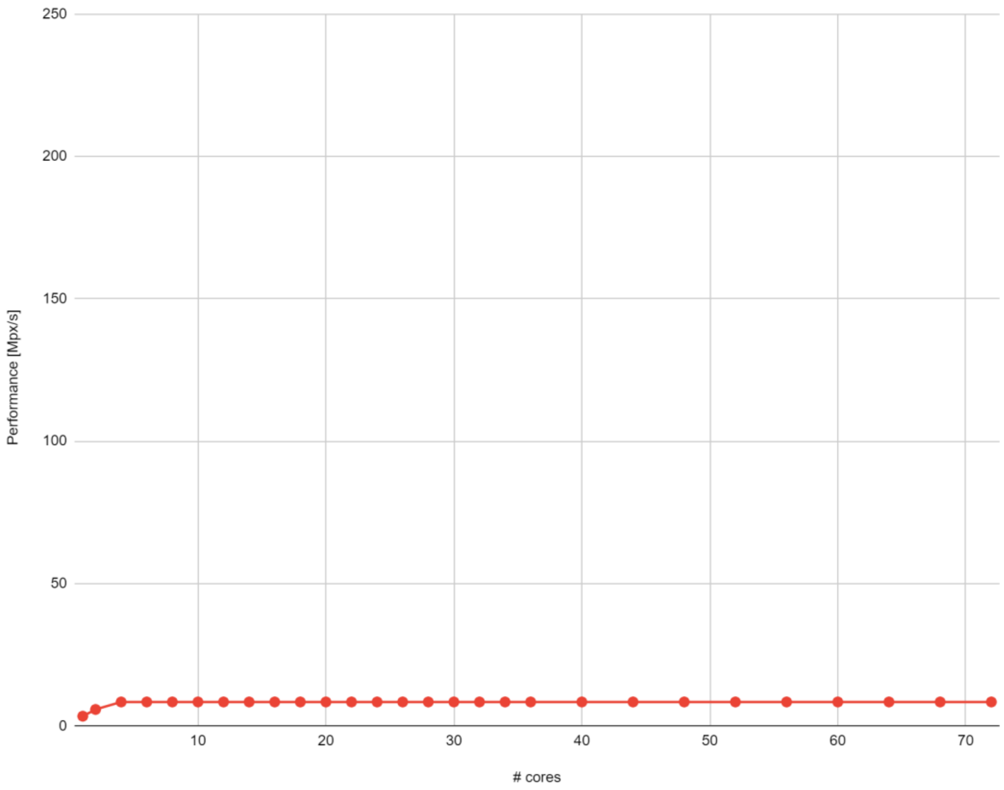


$$= \left(1/P_{peak} + \ 1/(I \times b_s)\right)^{-1}$$

Assignment 7 – Task 3

```
// ...
#pragma omp parallel private(tile)
{
tile=(char*)malloc(tilesize*tilesize*sizeof(char));

//..

count = 0;
#pragma omp for private(xc,i) reduction(+:count)
for(yc=0; yc<ytiles; yc++) {
  for(xc=0; xc<xtiles; xc++) {
     /* calc one tile */
     calc_tile(size, xc*tilesize, yc*tilesize, tilesize, tile);
     /* copy to picture buffer */
     for(i=0; i<tilesize; i++) {
         int tilebase = yc*tilesize*tilesize*xtiles+xc*tilesize;
         memcpy((void*)(picture+tilebase+i*tilesize*xtiles),
                (void*)(tile+i*tilesize),
                tilesize*sizeof(char));
     }
     count++;
   }
}
}
```

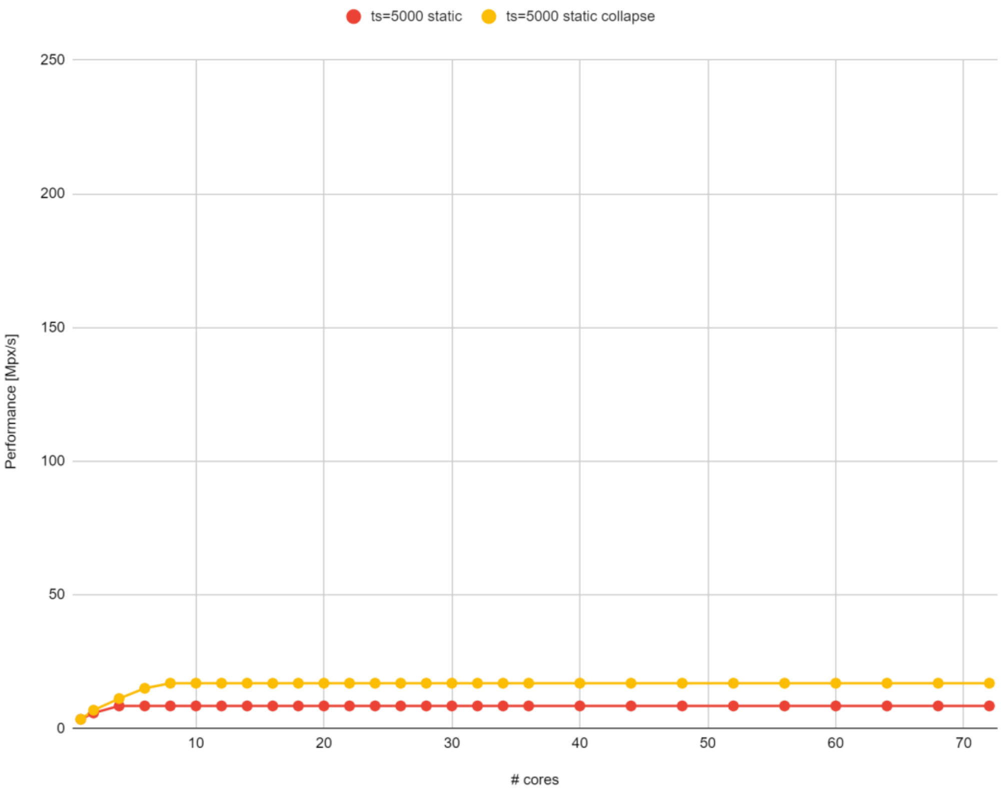
Assignment 7 – Task 3



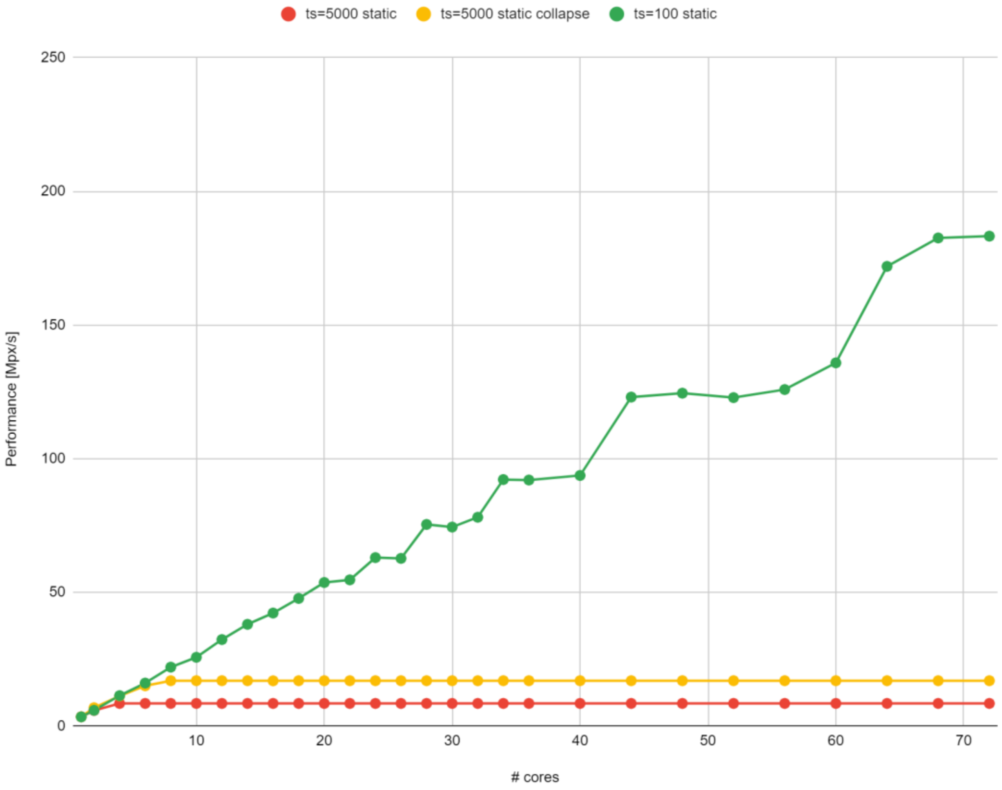
Assignment 7 – Task 3

```
// ...
#pragma omp parallel private(tile)
{
tile=(char*)malloc(tilesize*tilesize*sizeof(char));

//..

count = 0;
#pragma omp for collapse(2) private(xc,i) reduction(+:count)
for(yc=0; yc<ytiles; yc++) {
  for(xc=0; xc<xtiles; xc++) {
     /* calc one tile */
     calc_tile(size, xc*tilesize, yc*tilesize, tilesize, tile);
     /* copy to picture buffer */
     for(i=0; i<tilesize; i++) {
         int tilebase = yc*tilesize*tilesize*xtiles+xc*tilesize;
         memcpy((void*)(picture+tilebase+i*tilesize*xtiles),
                (void*)(tile+i*tilesize),
                tilesize*sizeof(char));
     }
     count++;
   }
}
}
```

Assignment 7 – Task 3



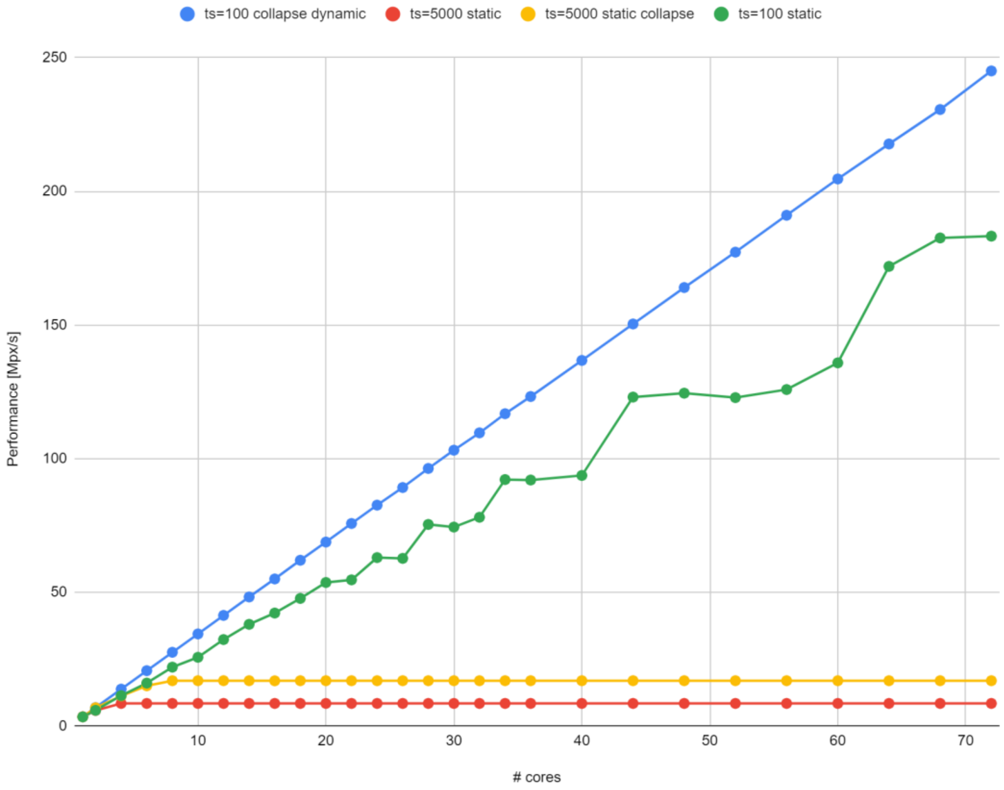
Assignment 7 – Task 3



Assignment 7 – Task 3

```
// ...
#pragma omp parallel private(tile)
{
tile=(char*)malloc(tilesize*tilesize*sizeof(char));

//..

count = 0;
#pragma omp for schedule(dynamic) collapse(2) private(xc,i) reduction(+:count)
for(yc=0; yc<ytiles; yc++) {
  for(xc=0; xc<xtiles; xc++) {
     /* calc one tile */
     calc_tile(size, xc*tilesize, yc*tilesize, tilesize, tile);
     /* copy to picture buffer */
     for(i=0; i<tilesize; i++) {
         int tilebase = yc*tilesize*tilesize*xtiles+xc*tilesize;
         memcpy((void*)(picture+tilebase+i*tilesize*xtiles),
                (void*)(tile+i*tilesize),
                tilesize*sizeof(char));
     }
     count++;
   }
}
}
```

Assignment 7 – Task 3



Assignment 7 – Task 3c)

Data transfer per pixel: 2B

With 123 Mpx/s (on 36 cores) → 246 MB/s