NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

Programming Techniques for Supercomputers Tutorial

Erlangen National High Performance Computing Center

Department of Computer Science

FAU Erlangen-Nürnberg

Sommersemester 2026

Assignment 9 – Task 1 a)

Optimizing loop code

```
int N = 16384;
double mat[N][N], s[N][N];
// ...
srand(1); // random seed

for(i=0; i<N; ++i) {
    val = (double)rand();
  for(j=0; j<N; ++j)
    mat[j][i] = s[j][i]*sqrt(val);
  }
}
```

move sqrt outside of inner loop

```
int N = 16384;
double mat[N][N], s[N][N];
// ...
srand(1); // random seed

for(i=0; i<N; ++i) {
    val = sqrt((double)rand());
  for(j=0; j<N; ++j)
    mat[j][i] = s[j][i]*val;
  }
}
```

Problems:

- wrong loop order → strided access
- outer dimension power of 2 w/ strides
- sqrt in inner loop

Assignment 9 – Task 1 a)

Optimizing loop code

interchange loops

```
int N = 16384;
double mat[N][N], s[N][N];
double table[N];
// ...
srand(1); // random seed

for(i=0; i<N; ++i)
  table[i] = sqrt((double)rand());

for(j=0; j<N; ++j)
  for(i=0; i<N; ++i)
    mat[j][i] = s[j][i]*table[i];
```

Compiler might move the sqrt but does not apply loop interchange or perform tabulation

Assignment 9 – Task 1 b)

Optimizing loop code

```
for(j=0; j<N; ++j)
  for(i=0; i<N; ++i)
    mat[j][i] = s[j][i]*table[i];
```

Assuming 2.4 GHz (base freq)

Assuming no sqrt → sqrt would take 3 cy → 1F/3cy → 28.8 GF/s

$\boldsymbol{P_{max}} = 1\frac{AVX512-it}{cy}$ per core ➔ **691 $\boldsymbol{GFLOP/s}$ per socket**

$\boldsymbol{B_C} = 24\frac{Byte}{Flop}$ with $b_S = 160\frac{GB}{s}$

Assuming WA → NT stores lead to 16 B/F

$$\boldsymbol{P} = \min\left(P_{\max}, \frac{b_S}{B_C}\right) = \min\left(691, \frac{160}{24}\right) GF/s = 6.67\ GF/s$$

memory-bound!

Assignment 9 – Task 2

Hardware Performance Counters a) Instrumentation

Marker calls have overhead!

```c
#include <likwid-marker.h>

int main(...) {
 LIKWID_MARKER_INIT;
#pragma omp parallel
 {
  LIKWID_MARKER_REGISTER("triad");
 }

 <… benchmark …>

 LIKWID_MARKER_CLOSE;
}
```

```c
#pragma omp parallel
{

 LIKWID_MARKER_START("triad");

 for(int j=0; j<niter; j++){
  #pragma omp for schedule(static)
  for(i=0; i<size; ++i)
   a[i] = b[i] + s * c[i];
  if(a[5]<0.0) // dummy
   cout<<a[3]<<b[5]<<c[1]<<d[9];
 }

 LIKWID_MARKER_STOP("triad");

}
```

Assignment 9 – Task 2

Hardware Performance Counters b)

```
for(i=0; i<size; ++i)
  a[i] = b[i] + s * c[i];
```

- Compile with

```
icpx -DLIKWID_PERFMON ... $LIKWID_INC $LIKWID_LIB ... -llikwid
```

- Single core `srun -cpu-freq=performance likwid-perfctr -g MEM -C S0:0 -m ./a.out`

  - Expect a read to write data volume ratio of **3 : 1**
  - Per iteration we expect $10^8 * (4 * 8\,byte) = 3.2\,Gbyte$
  - 3 iterations with expected result

Metric	HWThread 0
Runtime (RDTSC) [s]	0.4453
Runtime unhalted [s]	0.6465
Clock [MHz]	3500.0080
CPI	1.2930
Memory read bandwidth [MBytes/s]	16332.6790
Memory read data volume [GBytes]	7.2735
Memory write bandwidth [MBytes/s]	5406.6741
Memory write data volume [GBytes]	2.4078
Memory bandwidth [MBytes/s]	21739.3531
Memory data volume [GBytes]	9.6812

Assignment 9 – Task 2

Hardware Performance Counters b)

- Single NUMA domain (18 cores)

```
srun -cpu-freq=performance likwid-perfctr -g MEM -C S0:0-17 -m ./a.out
```

  - We still expect 3.2 Gbyte per iteration; w/ 15 iterations: 48 GB

Metric	Sum	Min	Max	Avg
Runtime (RDTSC) [s] STAT	8.7005	0.4827	0.4834	0.4834
Runtime unhalted [s] STAT	12.5907	0.6982	0.7003	0.6995
Clock [MHz] STAT	62921.5005	3495.6223	3495.6692	3495.6389
CPI STAT	84.2222	4.0695	5.0256	4.6790
Memory read bandwidth [MBytes/s] STAT	51802.4294	0	51802.4294	2877.9127
Memory read data volume [GBytes] STAT	25.0025	0	25.00	
Memory write bandwidth [MBytes/s] STAT	24905.1534	0	24905.15	
Memory write data volume [GBytes] STAT	12.0205	0	12.02	
Memory bandwidth [MBytes/s] STAT	76707.5829	0	76707.58	
Memory data volume [GBytes] STAT	37.0231	0	37.02	

Due to Icelake's automatic write-allocate evasion mechanism

  - Oberserved read to write data volume ratio of **2 : 1**; → 1 load less per iteration

Assignment 9 – Task 2

Hardware Performance Counters c) Cache Traffic

- L2 Cache

```
srun –cpu-freq=performance likwid-perfctr -g L2 -C S0:0-17 -m ./a.out
```

  - Read to write data volume ratio of **3 : 1**
  - Cache line claim does not help us here

- L3 Cache

```
srun –cpu-freq=performance likwid-perfctr -g L3 -C S0:0-17 -m ./a.out
```

  - Read to write data volume ratio of **1 : 1**
    Due to Fritz ICX exclusive victim L3 cache
  - E.g., every array is written back into L3, even if they aren’t modified
  - Read == evict from L2

Assignment 9 – Task 2

DAXPY code

```
a[i] = a[i] + s * b[i]
```

```
srun --cpu-freq=2000000-2000000:performance \
    likwid-pin -q -C E:N:1 ./bench-serial
Running with 1 threads.
Total walltime: 0.226606  NITER: 1048576  N: 2000
Perf [GFlop/s]: 17.651783
Cycles per parallel region: 453.212
```

a) Serial Performance

- With $N = 2000$, $P \approx 17.65\ [GF/s]$
- $T = \frac{4000[F]}{17.65\ [GF/s]} * 2 * 10^9[cy/s] = 453\ [cy]$

b) Parallel Performance ($nt :=$ number of threads)

- $nt = 1$: T = 1423 [cy]
- $nt = 2$: T = 3639 [cy] with $N = 4000$
- $nt = 4$: T = 4754 [cy] with $N = 8000$
- $nt = 18$: T = 6890 [cy] with $N = 36000$

Increase problem size with nt

Assignment 9 – Task 2

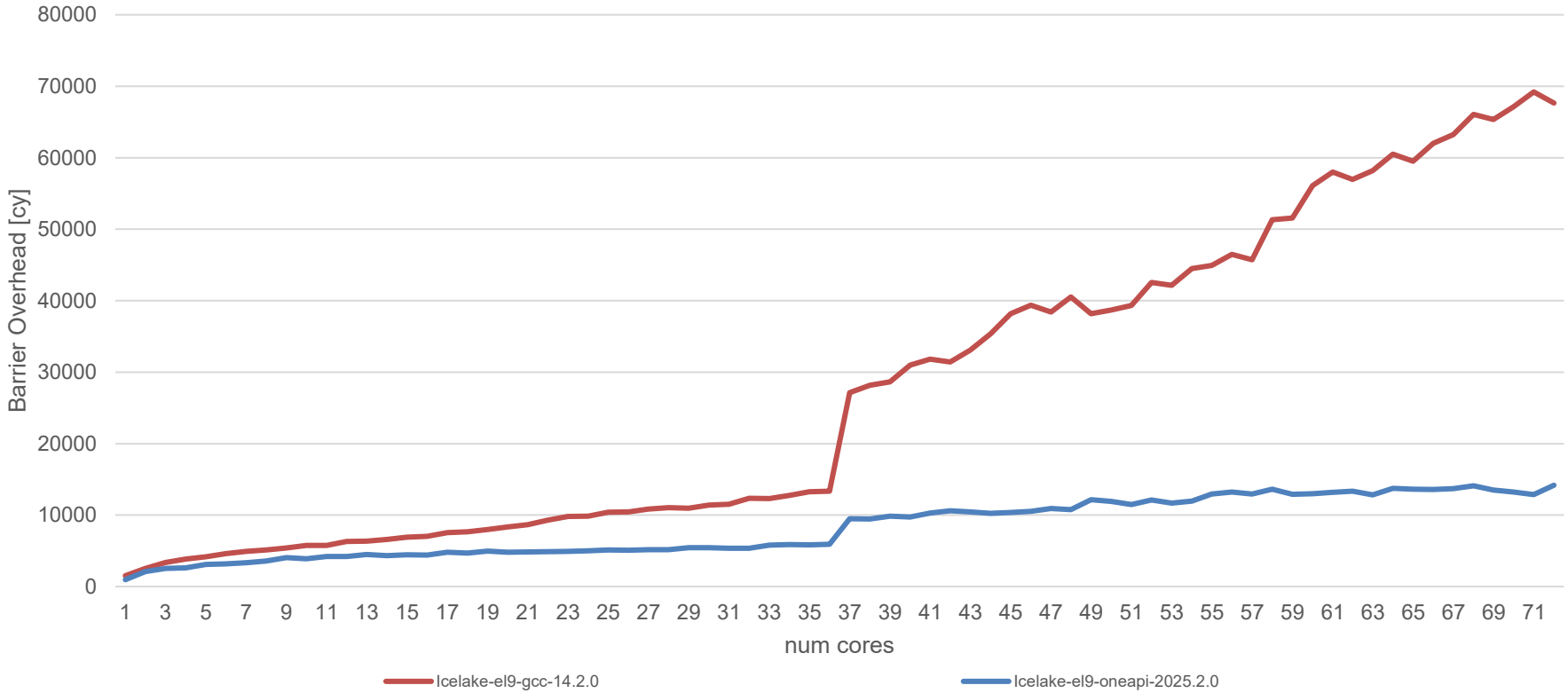
OpenMP Overhead

c) Overhead in cycles

- $nt = 1$:
  - Overhead: $1423 - 453 = \mathbf{970}\ [cy]$
- $nt = 2$:
  - Overhead: $3639 - 453 = \mathbf{3186}\ [cy]$
- $nt = 4$:
  - Overhead: $4754 - 453 = \mathbf{4301}\ [cy]$
- $nt = 18$:
  - Overhead: $6890 - 453 = \mathbf{6437}\ [cy]$

etc.

Assignment 9 – Task 3



Assignment 9 – Task 3