# Programming Techniques for Supercomputers

Friedrich-Alexander-Universität Erlangen-Nürnberg

Erlangen National High Performance Computing Center

Department of Computer Science

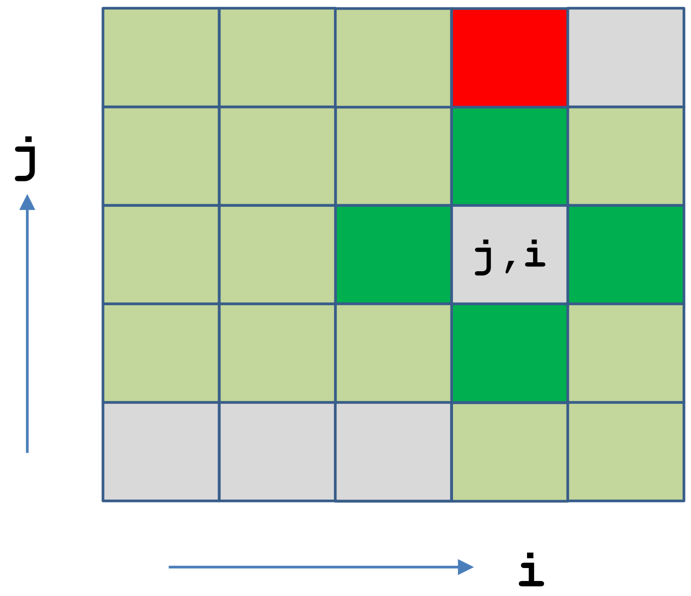
FAU Erlangen-Nürnberg

Sommersemester 2024

# Assignment 8 – Task 1

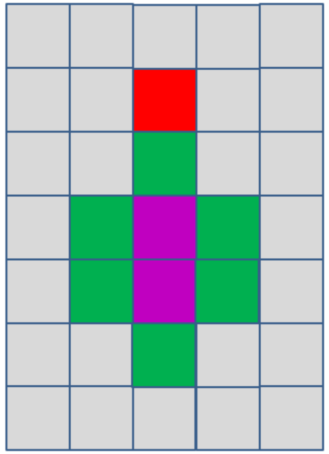
## 2D stencil code, double precision

```
#pragma omp parallel for schedule(static)
for(j=0; j<N-2; ++j)
  for(i=1; i<M-1; ++i)
    y[j][i] = c * (x[j][i-1]
                 + x[j][i+1]
                 + x[j-1][i]
                 + x[j+1][i]
                 + x[j+2][i]);
```




a) Layer condition: 4 rows of length M have to fit in the cache → **n_threads** $\times 4 \times M \times 8$ bytes $< C_t/2$, where $C_t$ is the cache size per thread

**(static,1)** → 4+(**n_threads**-1), as opposed to 4***n_threads**



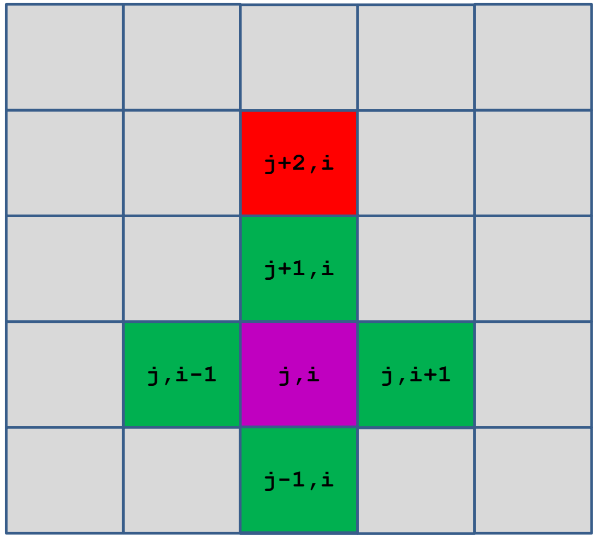
# Assignment 8 – Task 1

b) Code balance (assuming standard stores)

Case 1: LC is satisfied (best case)

→ per LUP: one load miss on `x[][]`
one store miss on `y[][]`

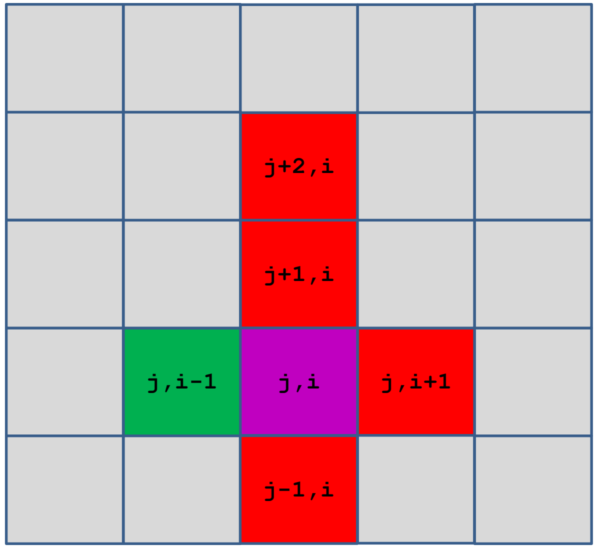
→ $B_c = (8 + 16)\ \mathrm{B/LUP} = 24\ \mathrm{B/LUP}$




Case 2: LC broken (worst case)

→ per LUP: four load misses on `x[][]`
one store miss on `y[][]`

→ $B_c = (4 \times 8 + 16)\ \mathrm{B/LUP} = 48\ \mathrm{B/LUP}$



# Assignment 8 – Task 1

c) Optimistic performance limit on one Fritz socket:

$$\mathrm{P} = \frac{b_s}{B_c} = \frac{150\ \mathrm{GB/s}}{24\ \mathrm{B/LUP}} = 6.25\ \mathrm{GLUP/s}$$

With the cache size of Fritz (1.5 MiB L3 + 1.25 MiB L2 per core) and static scheduling, we get:

$$\mathrm{M} < \frac{2.75\times10^{6}\ \mathrm{byte}}{2\times4\times8\ \mathrm{byte}} \approx 4\ \times 10^{4}$$

# Assignment 8 – Task 2

- $W_d$ Dynamic power consumption of a running core
- $n$ cores used (of $n_{max}$ available)
- $W_0$ Baseline power consumption of the chip (all cores idle): $W(n=0) = W_0$

→ Power $W(n) = W_0 + nW_d$

- $P(1)$ Performance of serial program
- $P_{limit}$ Maximum performance of parallel program
- $P(n)$ Performance of parallel program on $n$ cores
- $T(n)$ Time to solution on $n$ cores

- → Time to solution $T(n) = 1/(\min(nP(1), P_{limit}))$
- → Energy to solution $E(n) = W(n)T(n)$

# Assignment 8 – Task 2

a) $W_0 = 80W, W_d = 4W$

$P(1) = \frac{1}{s}, P_{limit} = \frac{15}{s}$

Minimum Energy:
15 cores

b) $P(1) = \frac{2.5}{s}$

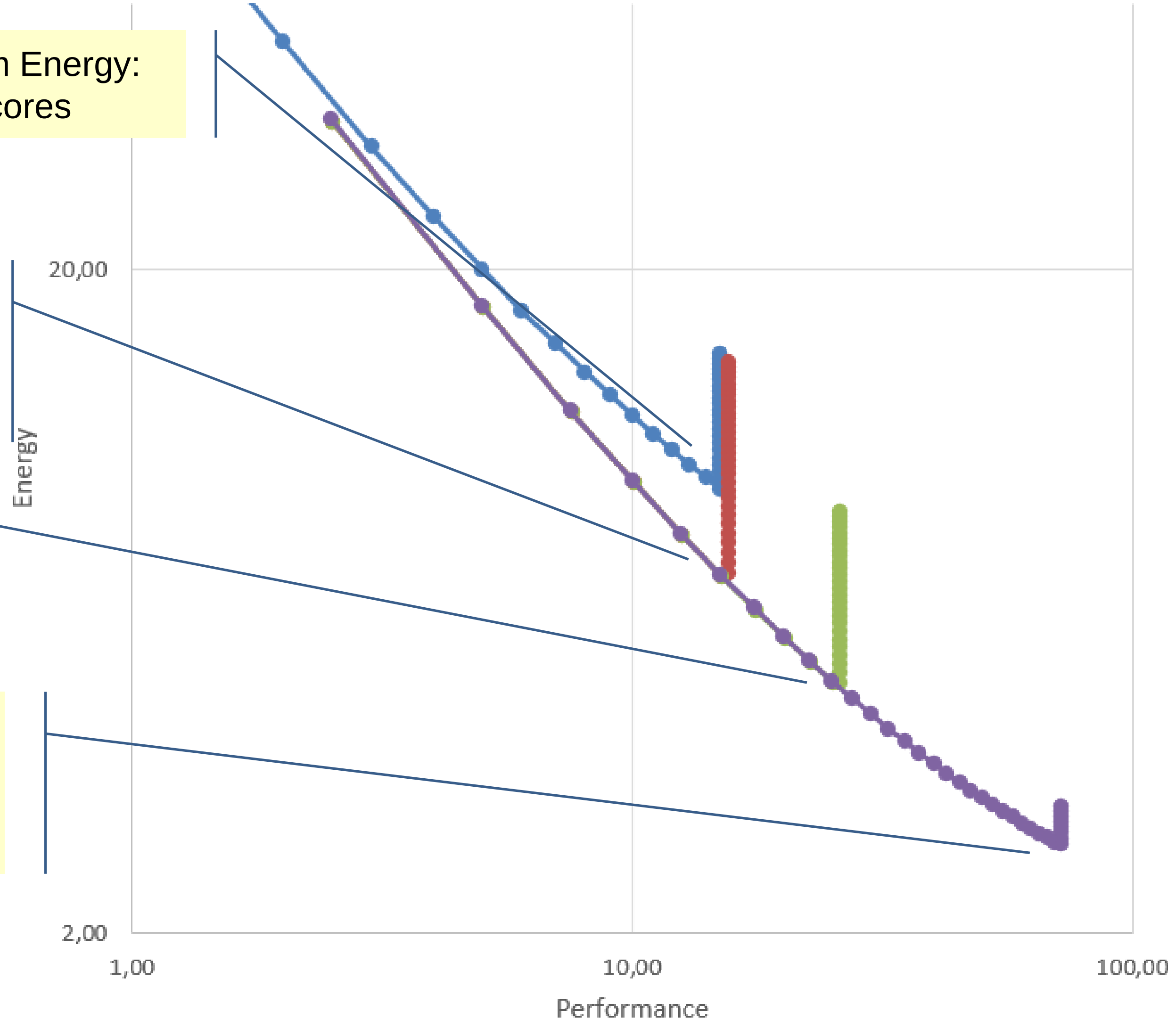
Speed up single core =>
Minimum Energy:
6 cores
Less cores to “saturate”.

c) $P_{limit} = \frac{25}{s}$

Optimize bottleneck
(i.e. cache blocking)=>
Minimum Energy:
10 cores

$P_{limit} = \frac{72}{s}$

Optimize bottleneck again =>
Minimum Energy:
30 cores



# Assignment 8 – Task 2

d) Remarks one could make:

- which frequency is "best"
- Minimum at each frequency
- How many cores to reach that minimum? (All of them)

Discuss the tradeoff of energy vs. performance:

"Going from 2.4 to 1.6 GHz we lose ⅓ of the performance but only about 16% of energy."