

# Programming Techniques for Supercomputers Tutorial

Erlangen National High Performance Computing Center

Department of Computer Science

FAU Erlangen-Nürnberg

Sommersemester 2026



# Assignment 5 – Task 1

## Multicore power envelope

### a) Overclocking

- Assumptions
  - No static power consumption  $W(0) = 0$
  - $W_d$  Dynamic power consumption at  $f = f_0$
  - $\Delta f$  Clock frequency change ( $\Delta f = f - f_0$ )
  - $W$  proportional to  $f^3$
  - Power consumption of one chip:  $W = W_d(1 + \Delta f/f_0)^3$
- What is  $W$  when  $\Delta f/f_0 = 0.3$ ?

$$W = W_d(1 + 0.3)^3 = W_d \cdot 2.197$$

# Assignment 5 – Task 1

## Multicore power envelope

### b) Multicore

- Denote  $\Delta v = \Delta f / f_0$

$$W(m) = m \cdot W_d (1 + \Delta v)^3 = W_d \text{ for } f_0$$

$$\Rightarrow m = (1 + \Delta v)^{-3}$$

$$m = (1 - 0.5)^{-3} = 8$$

We want to keep  $W_d$  fixed!

### c) Performance extrema

- Assume  $m = 8$  cores
- Minimum performance gain?
  - Speedup: 1, Memory bound codes** would not benefit from additional cores
- Maximum performance gain?
  - Speedup:  $m \cdot \Delta v = 8 \cdot 0.5 = 4$ . Compute bound codes** benefit here

# Assignment 5 – Task 1

## Multicore power envelope

### d) Energy to solution

$$W \propto (1 + \Delta f/f_0)^3$$

$$E \propto W \times t \propto W \times (1 + \Delta v)^{-1} \propto (1 + \Delta v)^2$$

- smallest  $E$  when  $\Delta v$  is small
- smallest  $\Delta v$  when  $\Delta f$  is small
- smallest  $\Delta f$  when  $f = 0$

# Assignment 5 – Task 2

## Optimal energy to solution

- $W_d$  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):  $W(n = 0) = W_0$

$$\text{Power : } W(n, f) = W_0 + nW_d$$

- $P(1)$  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

$$\text{Time to solution : } T(n) = 1/(\min(nP(1), P_{limit}))$$

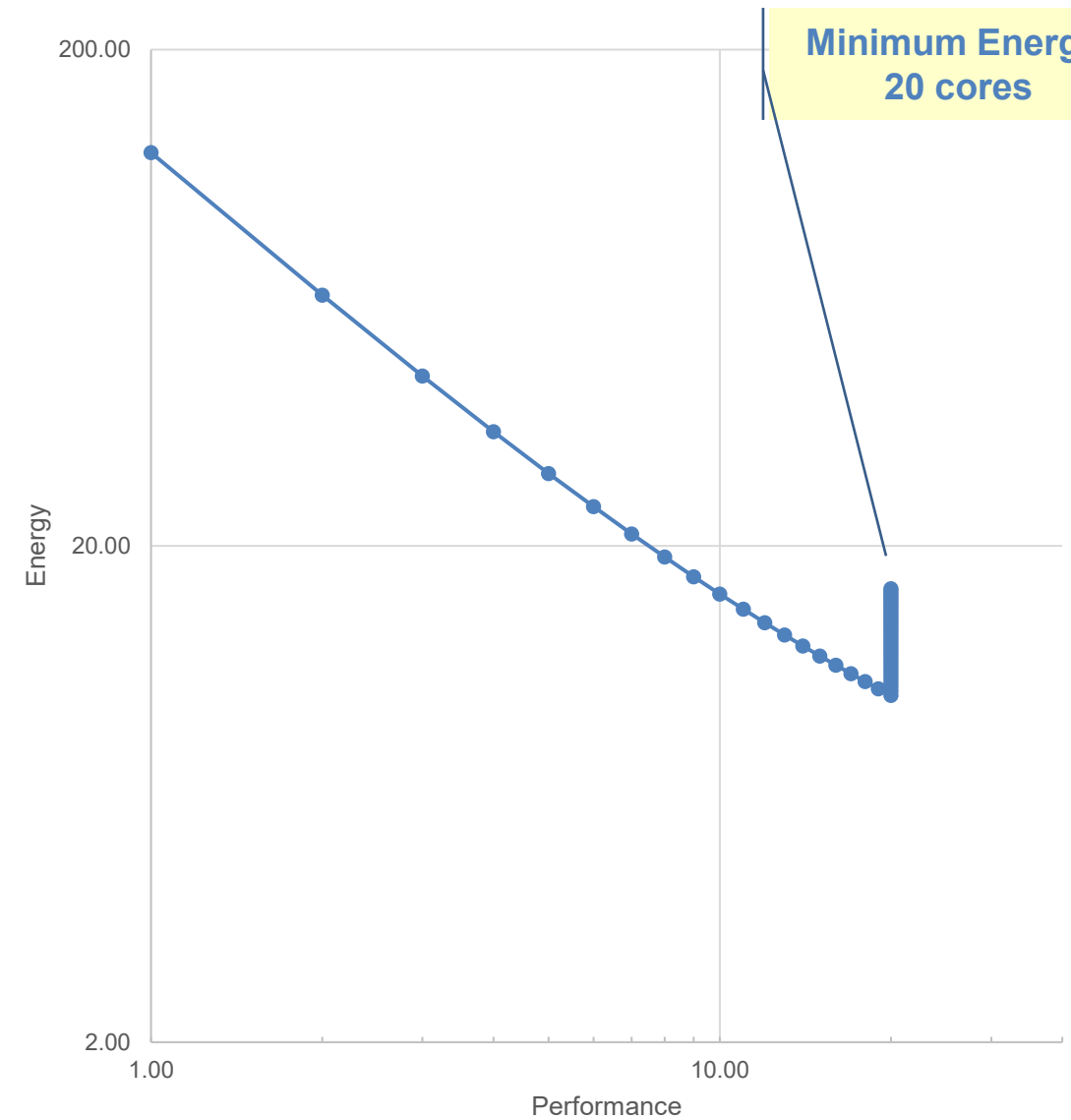
$$\text{Energy to solution : } E(n) = W(n, f) \times T(n) = (W_0 + nW_d)/(\min(nP(1), P_{limit}))$$

# Assignment 5 – Task 2

$$E(n) = (W_0 + nW_d) / (\min(nP(1), P_{limit}))$$

## Optimal energy to solution

- a)  $W_0 = 120[W]$ ,  $W_d = 4[W]$ ,  
 $P(1) = 1[s^{-1}]$ ,  $P_{limit} = 20[s^{-1}]$



# Assignment 5 – Task 2

$$E(n) = (W_0 + nW_d) / (\min(nP(1), P_{limit}))$$

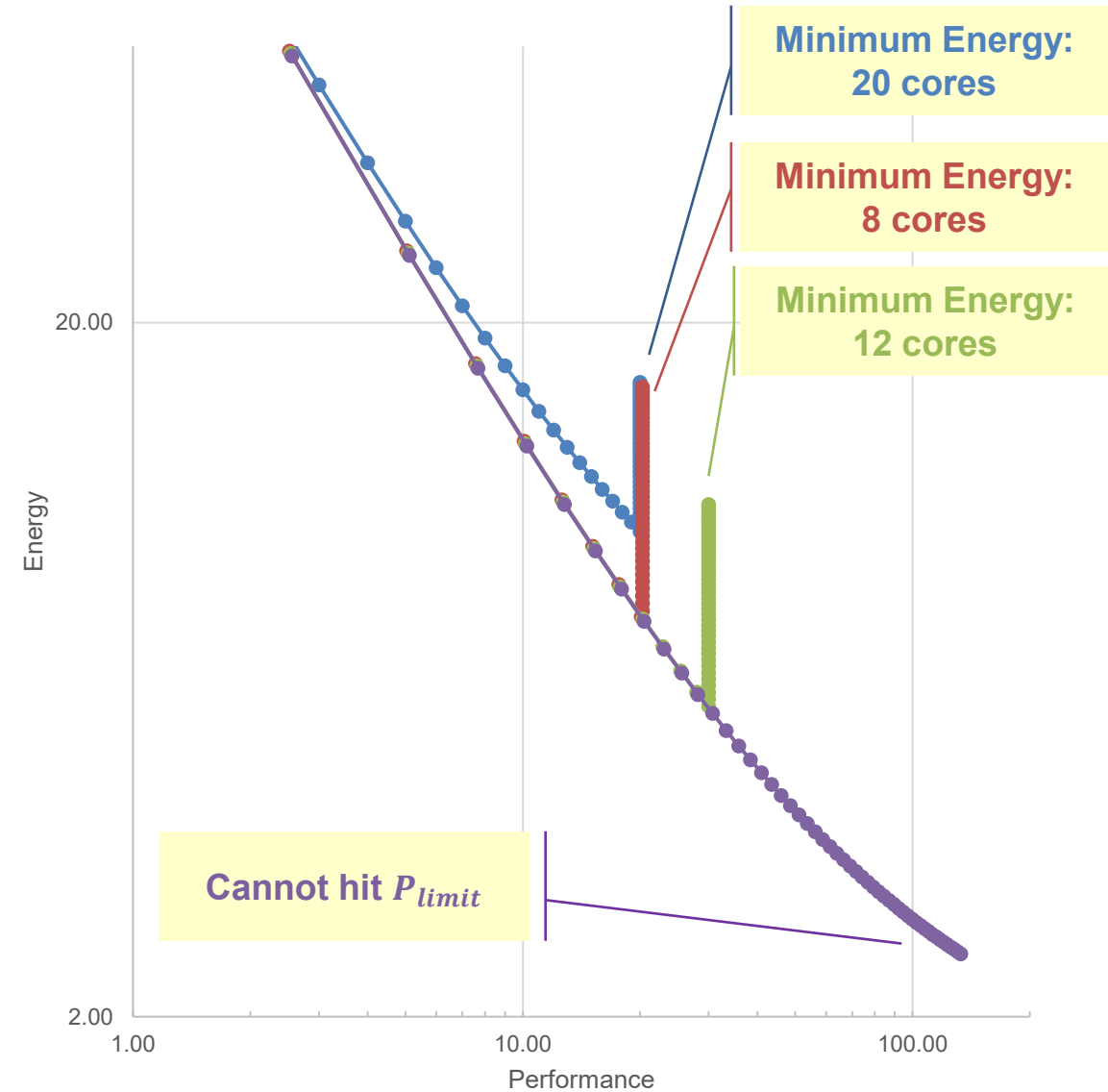
## Optimal energy to solution

a)  $W_0 = 120[W]$ ,  $W_d = 4[W]$ ,  
 $P(1) = 1[s^{-1}]$ ,  $P_{limit} = 20[s^{-1}]$

b)  $W_0 = 120[W]$ ,  $W_d = 4[W]$ ,  
 $P(1) = 2.5[s^{-1}]$ ,  $P_{limit} = 20[s^{-1}]$

c)  $W_0 = 120[W]$ ,  $W_d = 4[W]$ ,  
 $P(1) = 2.5[s^{-1}]$ ,  $P_{limit} = 30[s^{-1}]$

$W_0 = 120[W]$ ,  $W_d = 4[W]$ ,  
 $P(1) = 2.5[s^{-1}]$ ,  $P_{limit} = 200[s^{-1}]$



# Assignment 5 – Task 2

## Optimal energy to solution

d)  $W = W_0 + nW_d$  with  $W_d = W_2 f^2$

$$P = 1/T$$

$$\Rightarrow T = 1/nP(1) = \frac{1}{ns^{-1}} = \frac{1}{ncf}$$

$$E(f) = W \times T = \frac{W_0 + nW_2 f^2}{ncf}$$
$$= \frac{W_0}{nc} f^{-1} + \frac{W_2}{c} f$$

For the minimal  $E$ , get derivative:

$$\frac{dE}{df} = -\frac{W_0}{nc} f^{-2} + \frac{W_2}{c} \stackrel{!}{=} 0$$

$$\rightarrow f_{opt}^{-2} = \frac{W_2 nc}{W_0 c} = \frac{W_2 n}{W_0}$$

$$f_{opt} = \sqrt{\frac{W_0}{W_2 n}}$$

# Assignment 5 – Task 3

## Peak Performance and Bandwidth

```
for(int i=0; i < 1000000000; ++i)
    a[i] = b[i] + s * c[i];
```

$$P_{peak} = 48 \times 8 \text{ Flops/instr} \times 2 \text{ instr/cy} \times 2 \times 2.0 \text{ Gcy/s} = \mathbf{3.072 \text{ TFlop/s}}$$

$$b_s = 12 \times 3200 \text{ MT/s} \times 8 \text{ Byte/T} = \mathbf{307.2 \text{ GB/s}}$$

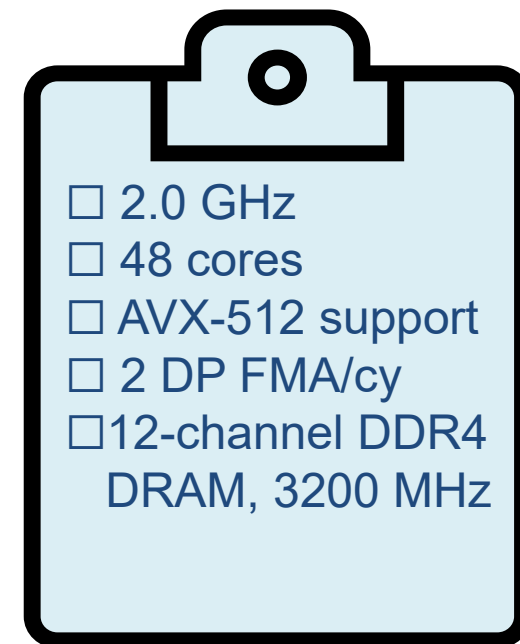
$$T_{Flop} = \frac{2 \times 10^9 \text{ Flop}}{3.072 \times 10^{12} \text{ Flop/s}} = \mathbf{651 \mu s}$$

$$T_{BW} = \frac{3 \times 8 \times 10^9 \text{ Byte}}{307.2 \times 10^9 \text{ Byte/s}} = \mathbf{78,125 \mu s}$$

**100x more, memory bandwidth is clearly the bottleneck!**

$P_{peak}$  is **not** realistic (**LD/ST limit**),  $P_{max}$  would be a better choice for a refined model.

$$b_{s,GPU} = 1400 \text{ GB/s} \approx \mathbf{4.6 \times b_s} \rightarrow \mathbf{\text{expected speedup } 4.6x}$$



# Assignment 5 – Task 4

## Energy of $\pi$

```
$ srun --cpu-freq=1200000-1200000:performance likwid-perfctr -g ENERGY -C 0 ./pi
```

```
-----  
Result=3.141592653579389 in 5.031e+02 s -> 12.07 cy/it (N = 100000000000)  
-----
```

Metric	HWThread 0
Runtime (RDTSC) [s]	<b>501.9134</b>
Runtime unhaltd [s]	250.1449
Clock [MHz]	1200.0195
Uncore Clock [MHz]	800.0039
CPI	0.3531
Temperature [C]	41
Energy [J]	<b>25585.8011</b>
Power [W]	<b>50.9765</b>
Energy DRAM [J]	2093.9931
Power DRAM [W]	4.1720

# Assignment 5 – Task 4

## Energy of $\pi$

Baseline power **too large** compared to dynamic power of one core  
→ **faster is better**

