ERLANGEN REGIONAL
COMPUTING CENTER

What Role Does Software Play in Energy Efficiency?

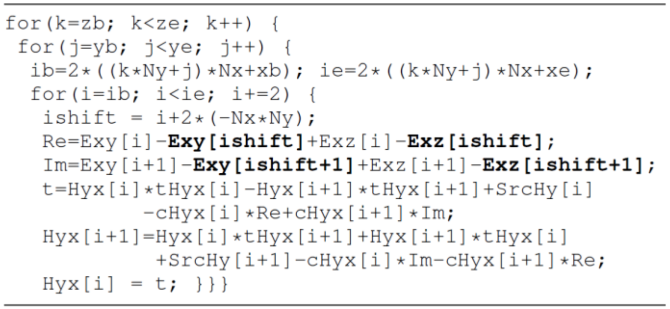
Georg Hager
Erlangen Regional Computing Center (RRZE)

SC15 Workshop Panel on Energy-Efficient Supercomputing
November 15, 2015
Austin, TX

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

(My) Definitions

Software



```
for(k=zb; k<ze; k++) {
 for(j=yb; j<ye; j++) {
  ib=2*((k*Ny+j)*Nx+xb); ie=2*((k*Ny+j)*Nx+xe);
  for(i=ib; i<ie; i+=2) {
   ishift = i+2*(-Nx*Ny);
   Re=Exy[i]-Exy[ishift]+Exz[i]-Exz[ishift];
   Im=Exy[i+1]-Exy[ishift+1]+Exz[i+1]-Exz[ishift+1];
   t=Hyx[i]*tHyx[i]-Hyx[i+1]*tHyx[i+1]+SrcHy[i]
           -cHyx[i]*Re+cHyx[i+1]*Im;
   Hyx[i+1]=Hyx[i]*tHyx[i+1]+Hyx[i+1]*tHyx[i]
            +SrcHy[i+1]-cHyx[i]*Im-cHyx[i+1]*Re;
   Hyx[i] = t; }}}
```

- Applications running on the system
- Auxiliary software controlling the execution of applications
- Tools that generate “advice” on or “automate” how to run apps

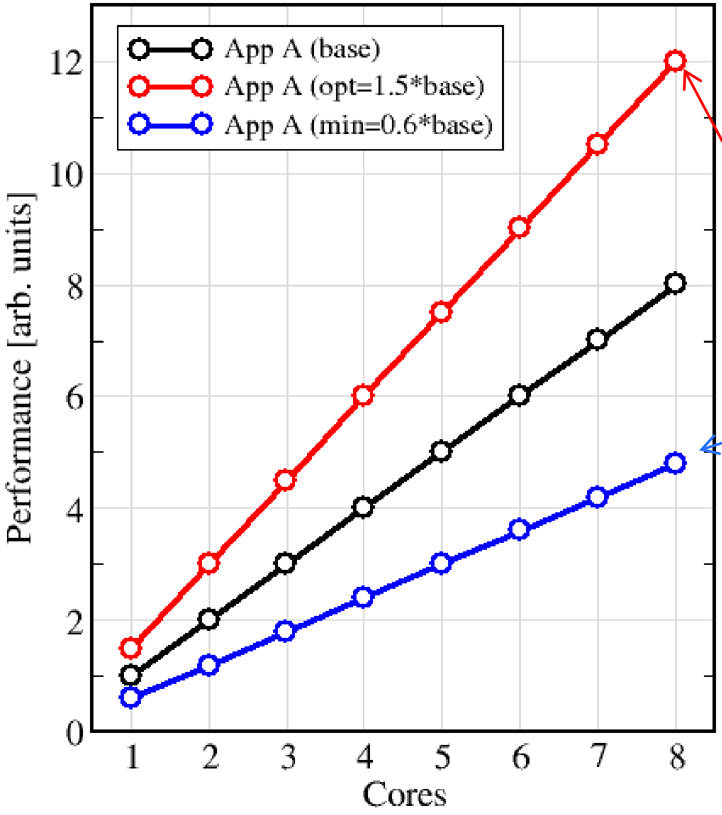
Energy Efficiency

- Energy to solution
- Energy-delay-whatever product
- Running under a power cap

Two types of application phases

"LINPACK type"

Limiting factor: core execution



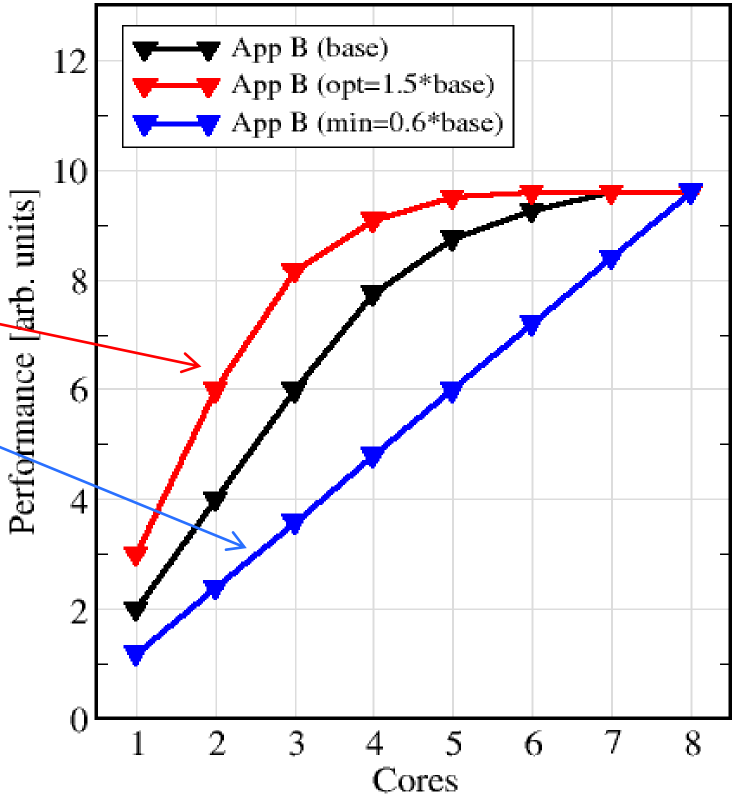
Change clock speed:

1.5x

0.6x

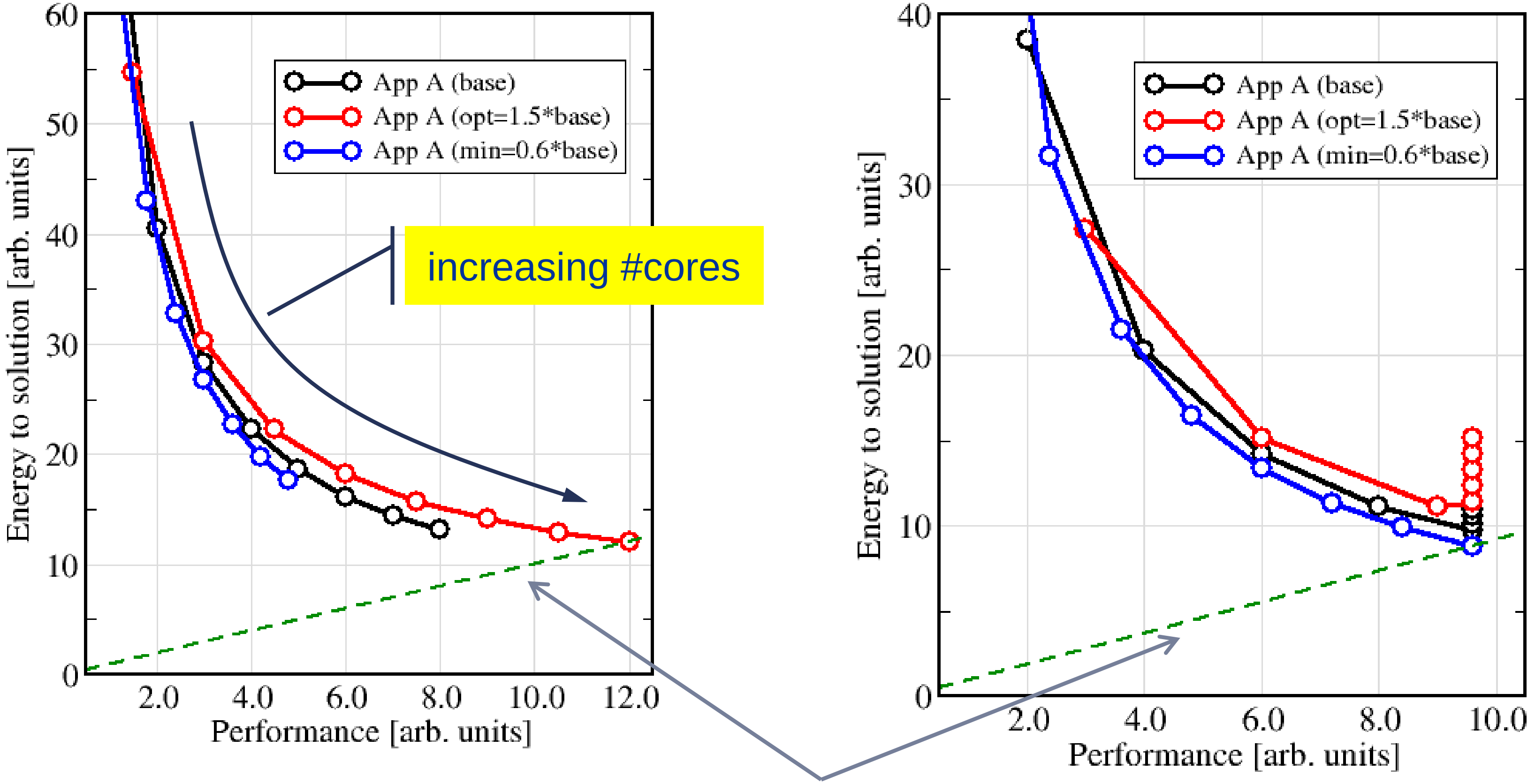
"STREAM type"

Limiting factor: saturation (bandwidth)



An interesting way of plotting energy: The Z-plot

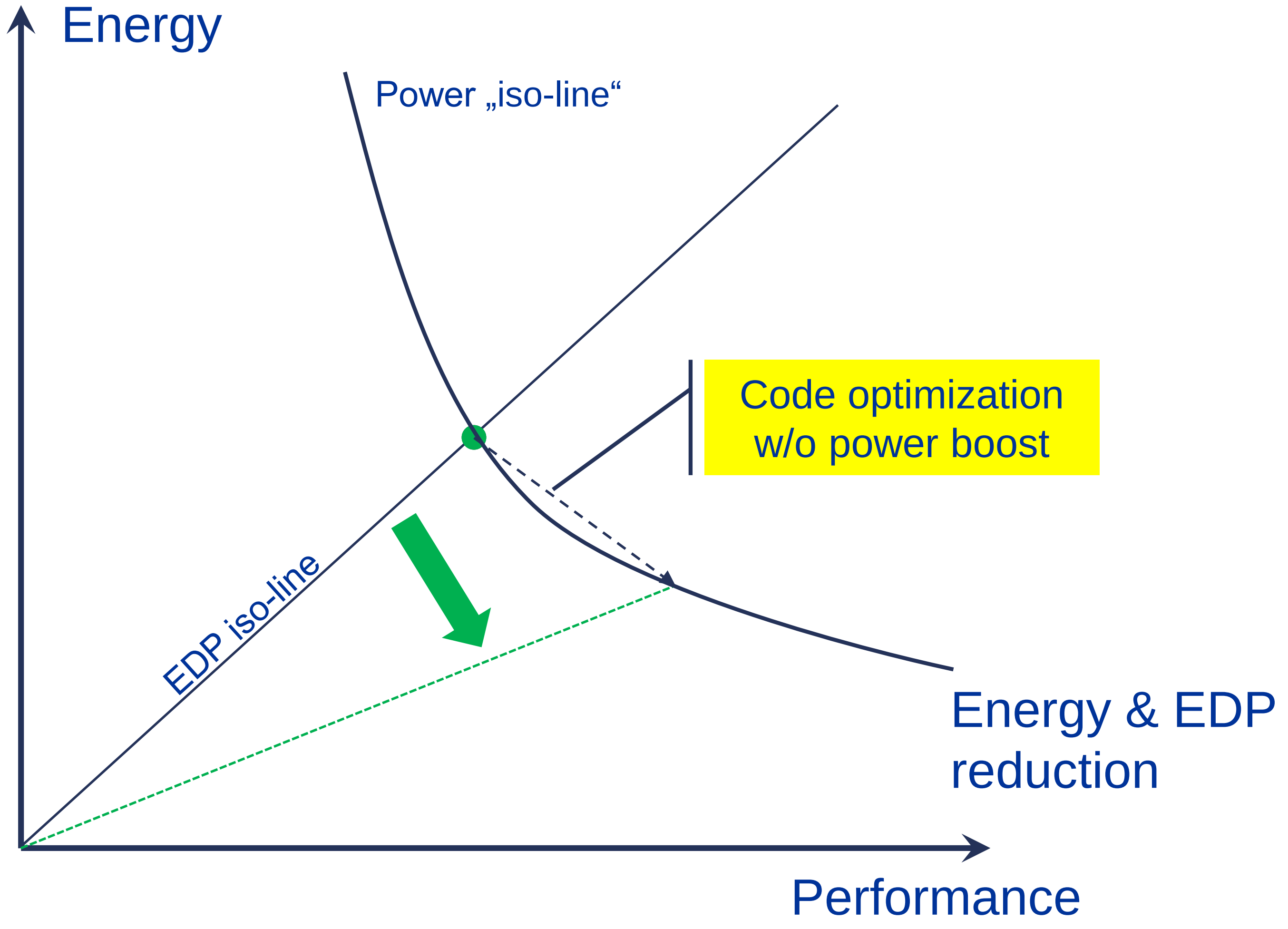
(a.k.a. “Zeiser Plot”)



increasing #cores

“Isoline” of constant energy-delay product ($\boldsymbol{E} \times \Delta \boldsymbol{t}$)

Finding your way in the Z-plot



Energy

Power „iso-line“

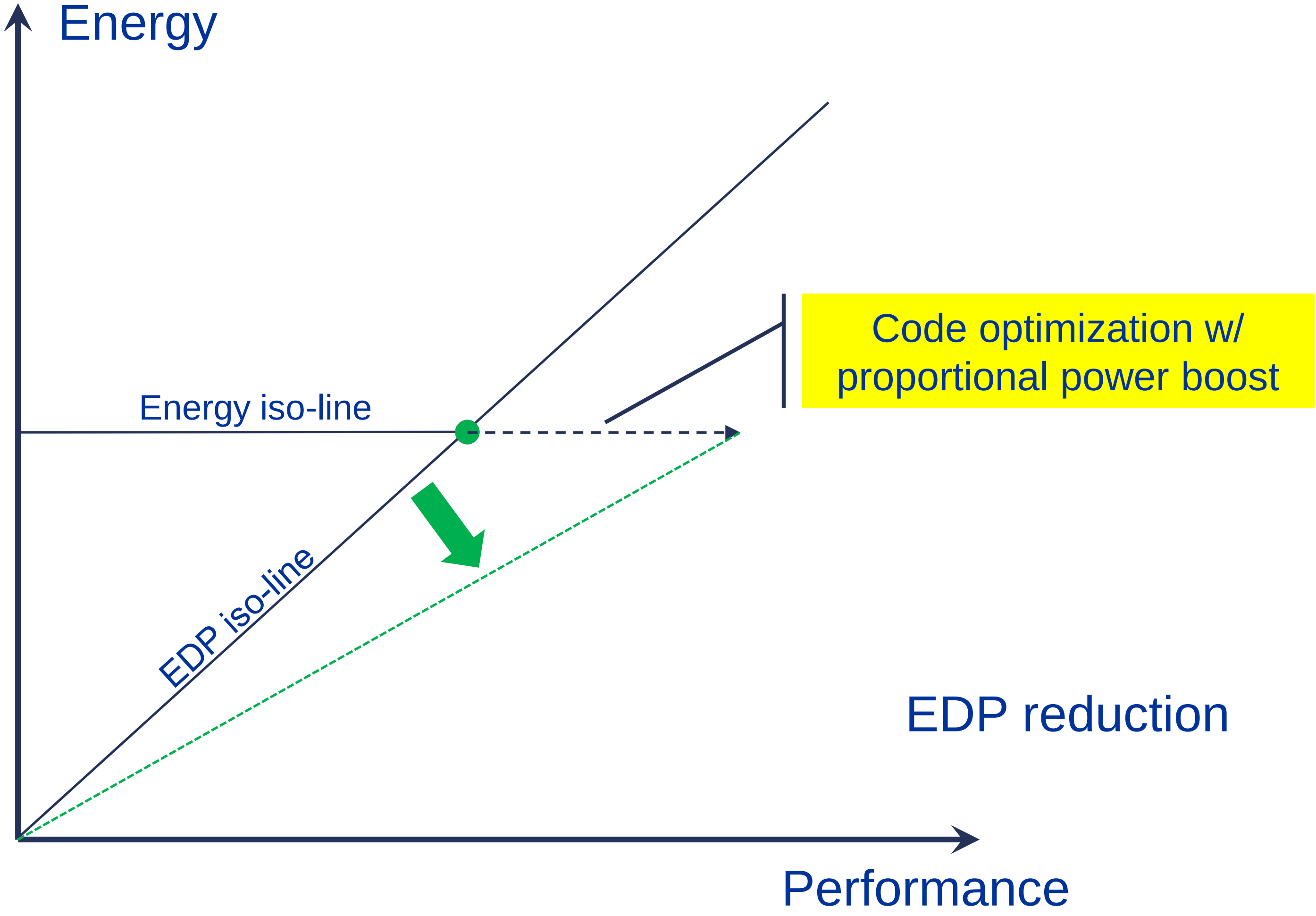
Code optimization
w/o power boost

EDP iso-line

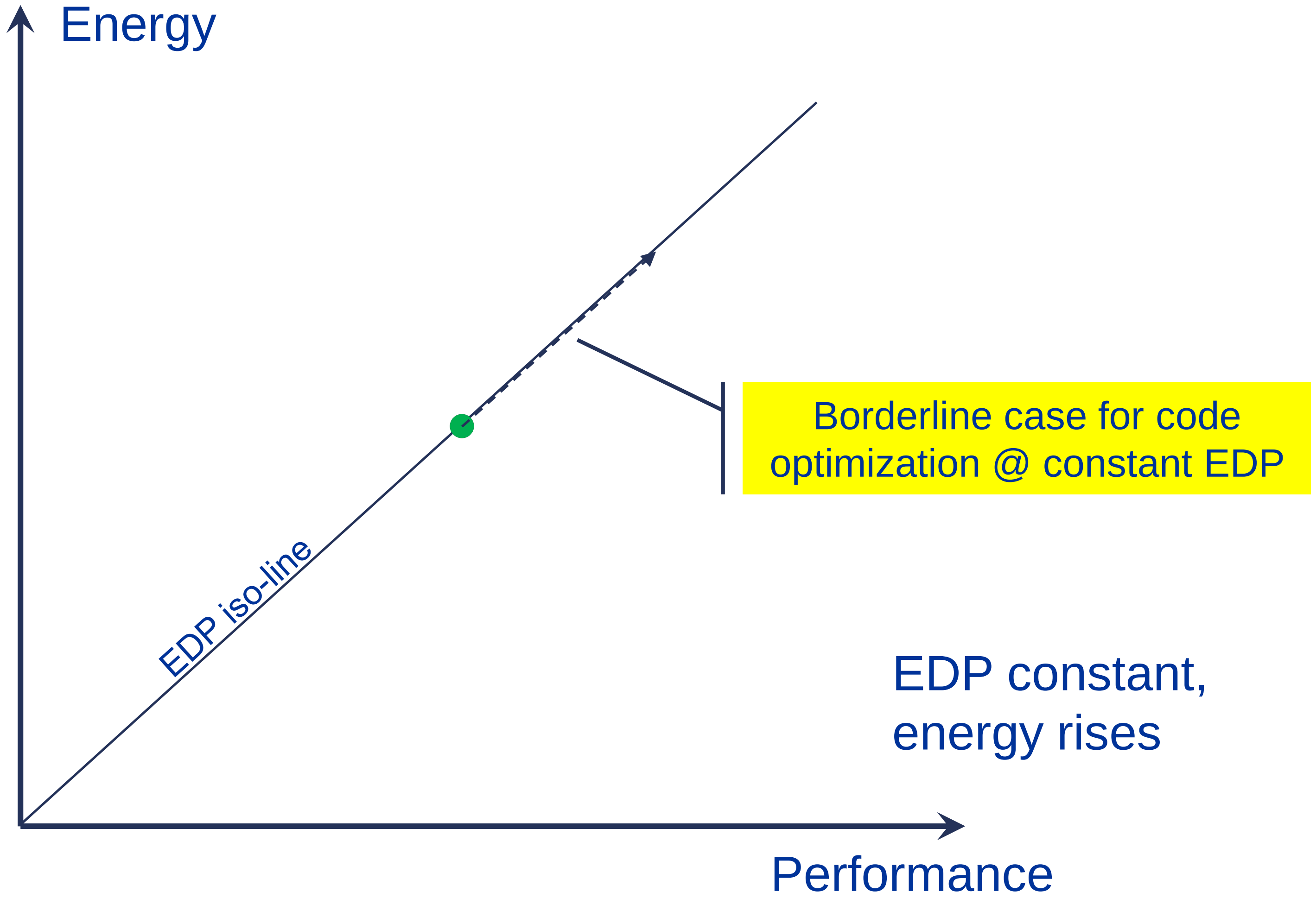
Energy & EDP
reduction

Performance

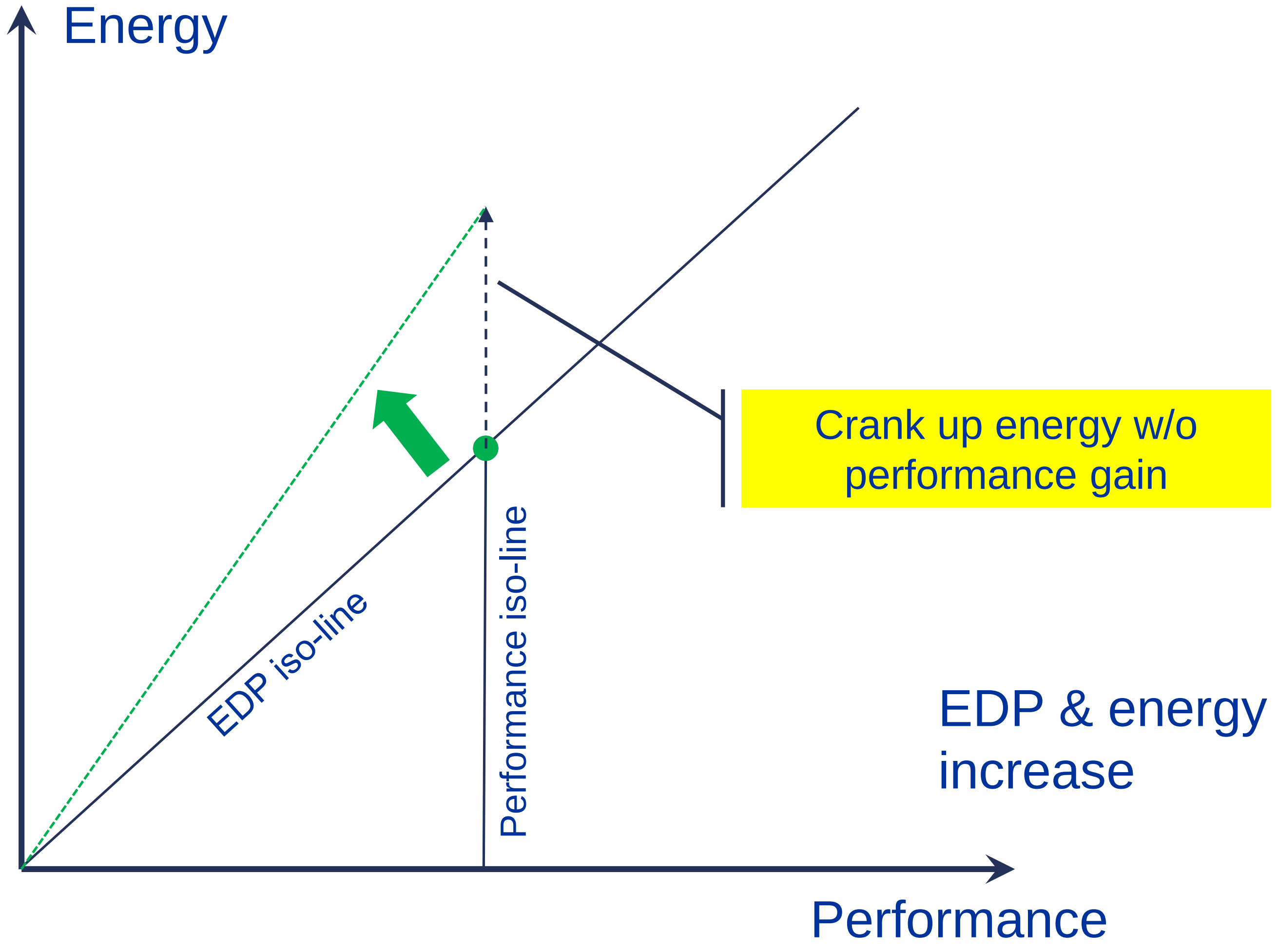
Finding your way in the Z-plot



Finding your way in the Z-plot


Energy
EDP iso-line
Borderline case for code optimization @ constant EDP
EDP constant, energy rises
Performance

Finding your way in the Z-plot



Applying it

Take control on a phase-by-phase basis.

Parameters for taking control

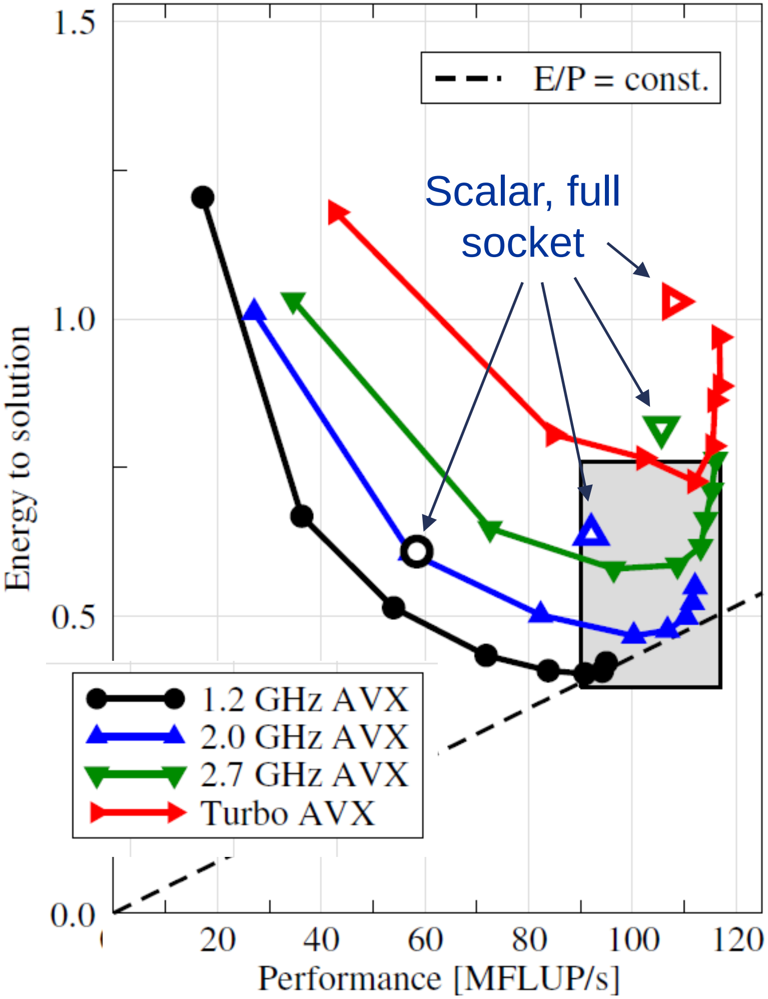
- Serial code performance
- Number of active cores
- Clock speed

Automated tools will never understand your code as well as you can.

Given enough knowledge about the code, energy can be modeled.

DOI: 10.1002/cpe.3489

ILBDC Lattice-Boltzmann @ SNB



Wish list: Give me…

- A fast library for changing P-states (clock speed)

- A well-defined way of dealing with affinity with changing #threads per socket
- Tools that help in figuring out the energy cost of basic operations
  - Flops (SIMD, scalar,….)
  - Data transfers (inter-cache, memory)
  - Synchronization (barriers, MPI wait)

- Tools that help establish performance and energy models
  - Promising examples: PerfExpert, kerncraft, RL Toolkit

- Zero idle power, to make all other efforts beyond faster code more effective!