

Optimizing a 3D Multigrid Algorithm for the IA-64 Architecture

Markus Stürmer, Jan Treibig, Ulrich Rüdè
System Simulation (Computer Science 10)
University Erlangen-Nuremberg

Outline

- Motivation and Introduction
 - The Itanium 2 Processor
 - Model Problem
 - Simple Multigrid V-Cycle
- Optimization
 - Reduction of Floating Point Operations
 - Reduction of Memory Throughput
 - Prefetching Techniques
- Results
- Conclusions



IA-64 - Itanium 2

key concept:

- simpler control logic...
 - in-order execution
 - Explicitly Parallel Instruction Computing
 - no hardware prefetchers
- ... but more resources
 - large register file
 - big and fast caches
 - multiple execution units
 - up to 6 instructions per cycle (2 fused multiply-add)
- looks suitable for scientific computing
- more predictable behavior enables better study of optimization techniques



Poisson's Equation...

- elliptic PDE

$$\Delta \phi = f \quad \xrightarrow{\text{3D}} \quad \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} = f$$

- discretized by finite differences to

$$\frac{U_{x+1,y,z} + U_{x-1,y,z} + U_{x,y+1,z} + U_{x,y-1,z} + U_{x,y,z+1} + U_{x,y,z-1} - 6 \cdot U_{x,y,z}}{h^2} = F_{x,y,z}$$

- 7-point stencil update:

$$U(x,y,z) = \left(\frac{1}{6} \right) \cdot \left(\begin{aligned} &U(x+1,y,z) + U(x-1,y,z) + U(x,y+1,z) \\ &+ U(x,y-1,z) + U(x,y,z+1) + U(x,y,z-1) \\ &- h^2 \cdot F(x,y,z) \end{aligned} \right)$$

- here: Dirichlet boundary conditions only

Relaxation Methods

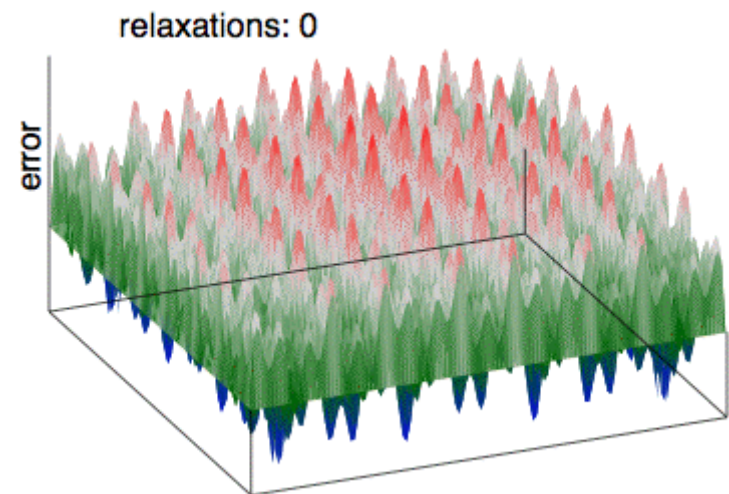
linear system of equations from descretized PDE

$$A u = f$$

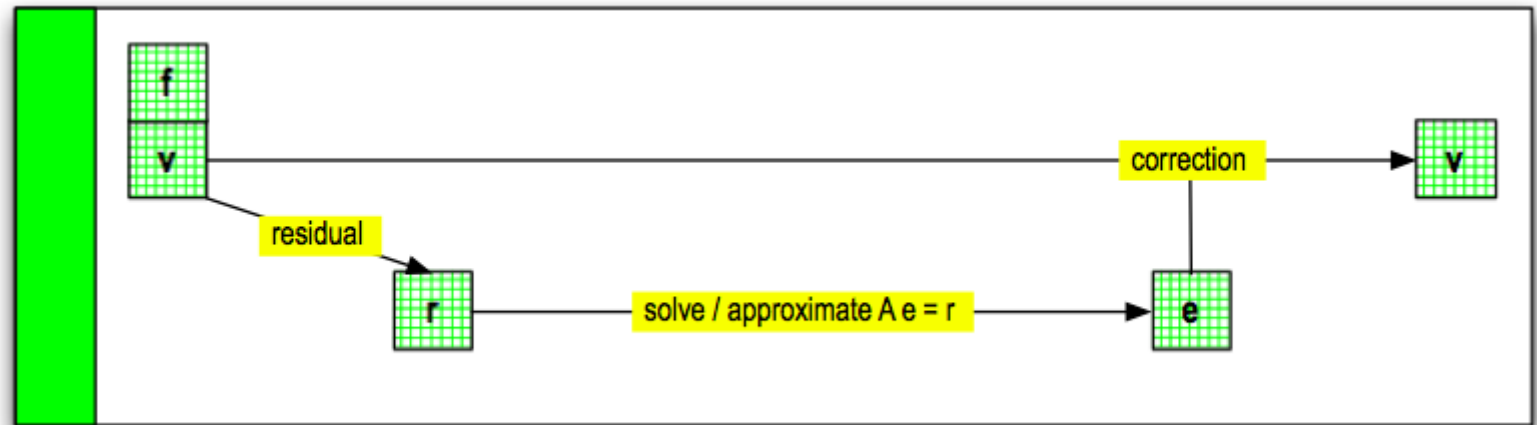
we usually only have an approximation with unknown error

$$A (v + e) = f$$

- most iterative solvers have error smoothing property



Residual Equation



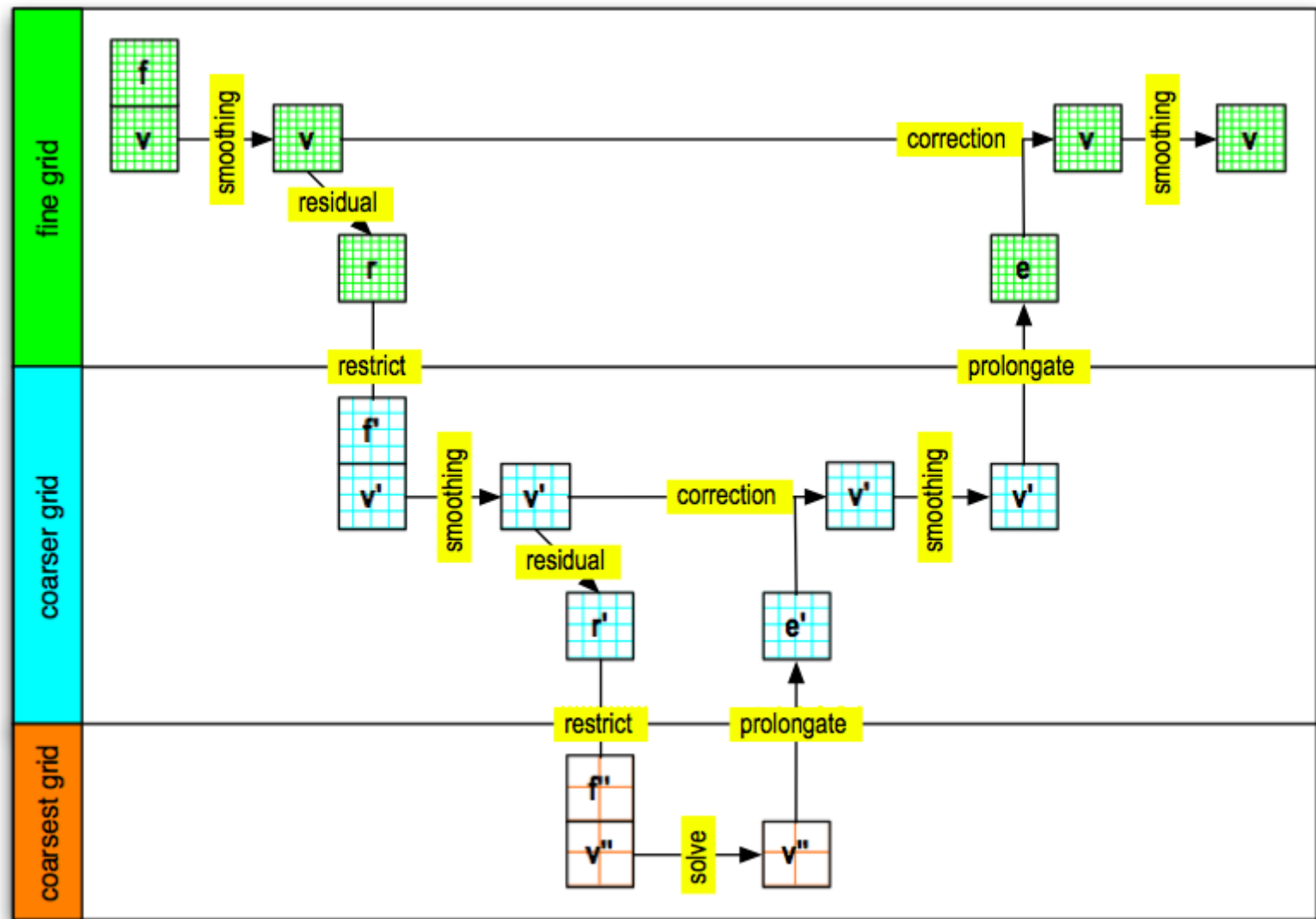
The residual

$$r = f - A v$$

and the error satisfy the original relationship

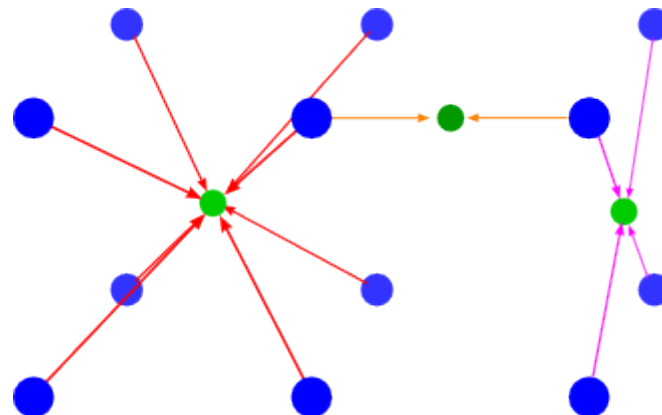
$$A e = r$$

The Multigrid V-Cycle



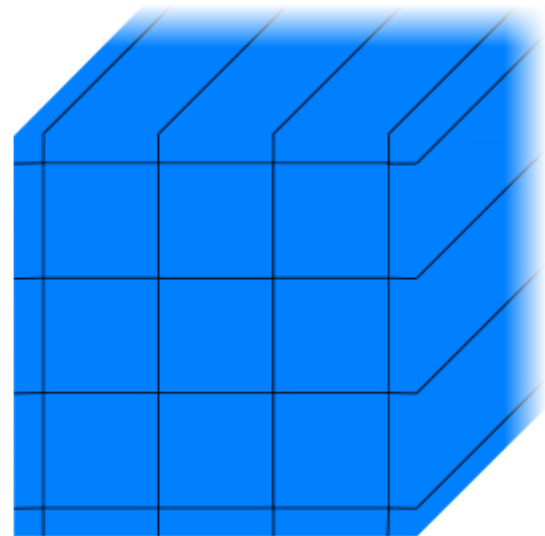
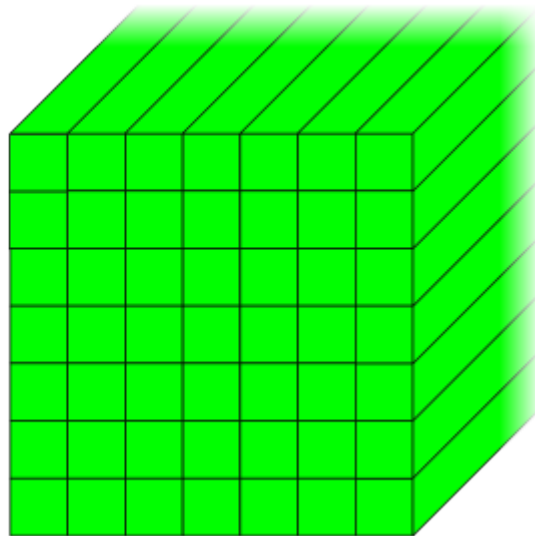
Components Poisson 3D MG V-Cycle

- smoother
 - Red-Black Gauss-Seidel
- calculate residual
- restriction
 - 27 point stencil (full weighting)
- prolongation and correction
 - tri-linear interpolation



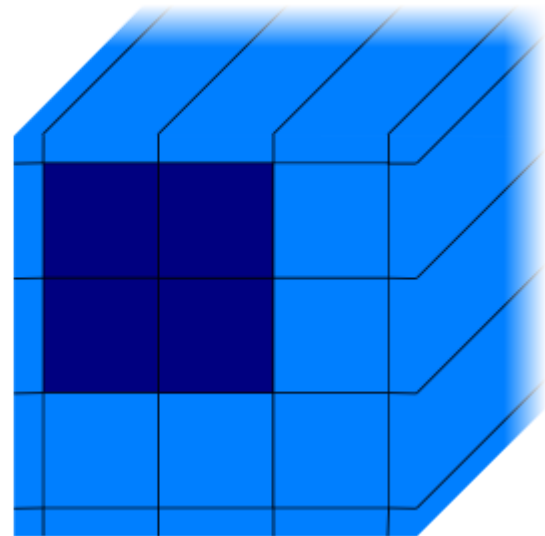
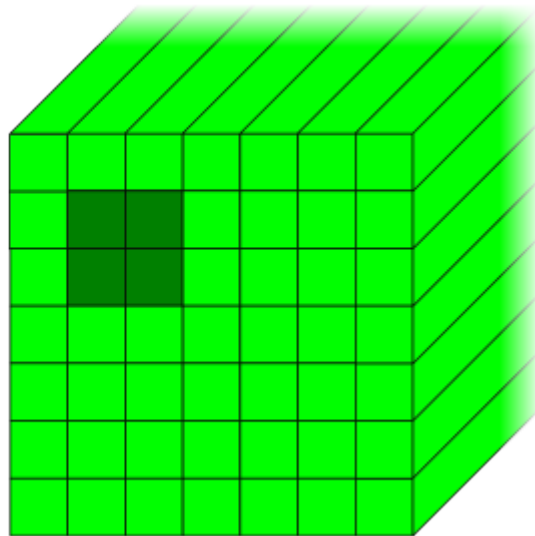
Optimization I

- prolongation / interpolation
 - can neglect “red” points if at least one post-smoothing step
 - local blocking
 - fine grid points visited only once



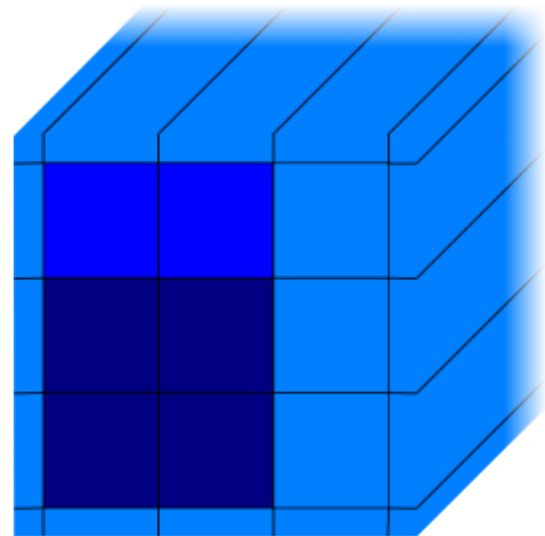
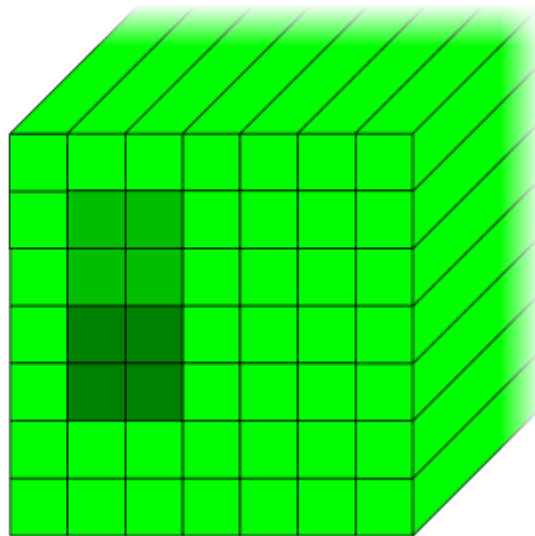
Optimization I

- prolongation / interpolation
 - can neglect “red” points if at least one post-smoothing step
 - local blocking
 - fine grid points visited only once



Optimization I

- prolongation / interpolation
 - can neglect “red” points if at least one post-smoothing step
 - local blocking
 - fine grid points visited only once



Optimization II

- prolongation / interpolation
 - can neglect “red” points if at least one post-smoothing step
 - local blocking
 - fine grid points visited only once
- restriction
 - local blocking
 - residual at “black” points equals zero if at least one pre-smoothing step



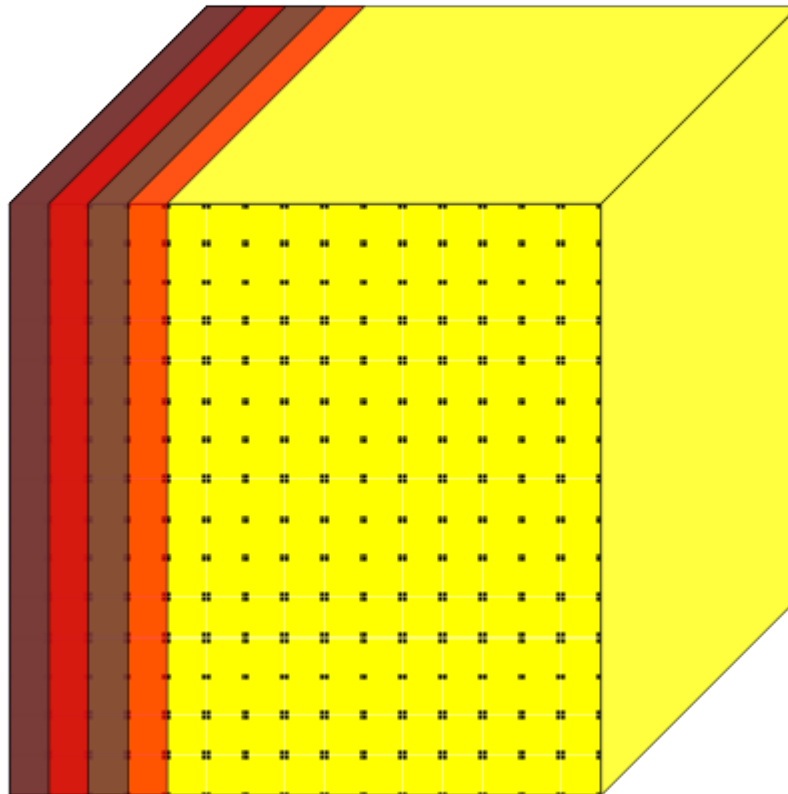
Optimization III

- prolongation / interpolation
 - can neglect red points if at least one post-smoothing step
 - local blocking
 - fine grid points visited only once
- restriction
 - local blocking
 - residual at “black” points equals zero if at least one pre-smoothing step
- calculation of residual
 - only necessary for “red” points
 - can be written over “red” unknowns



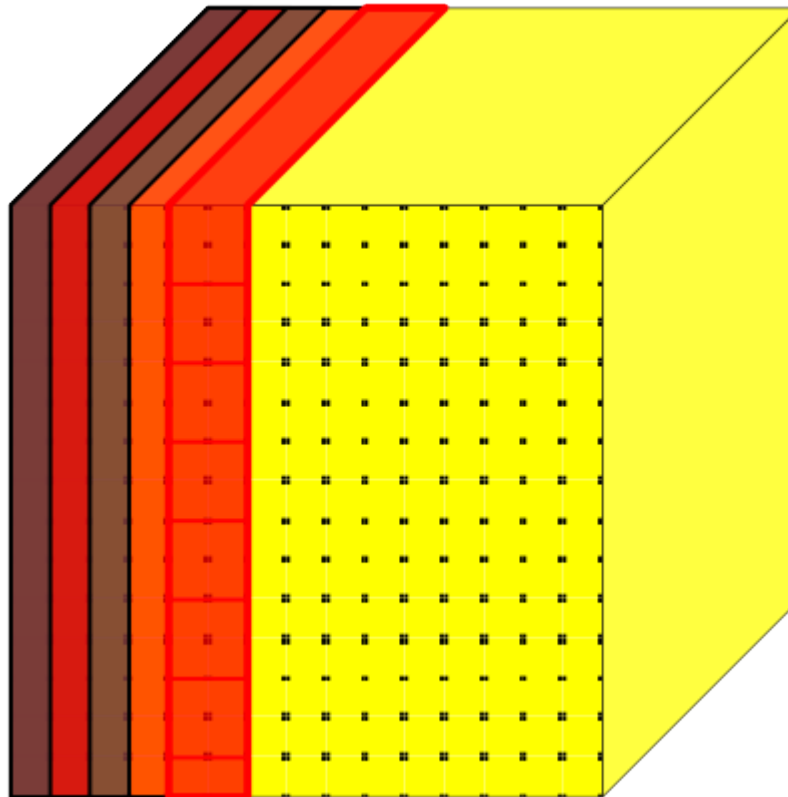
Temporal Blocking

- smoother
calculate residual } temporal blocking



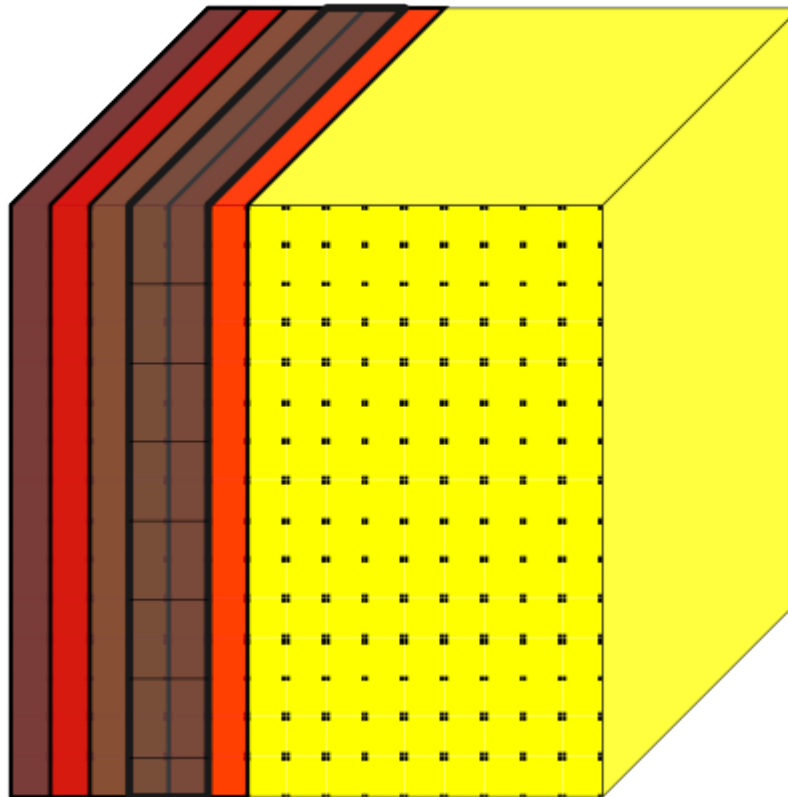
Temporal Blocking

- smoother
calculate residual } temporal blocking



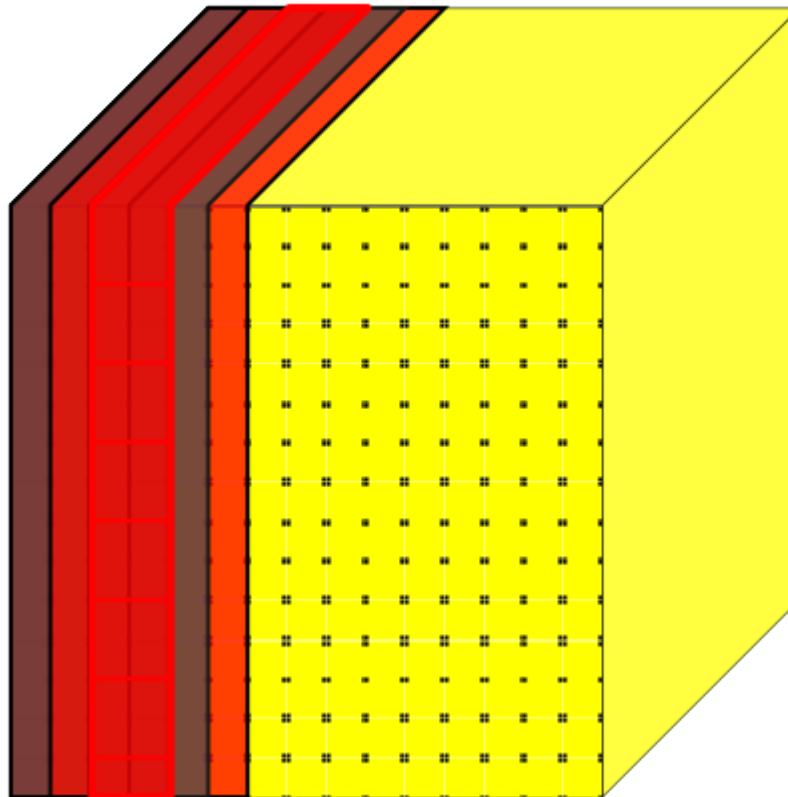
Temporal Blocking

- smoother
calculate residual } temporal blocking



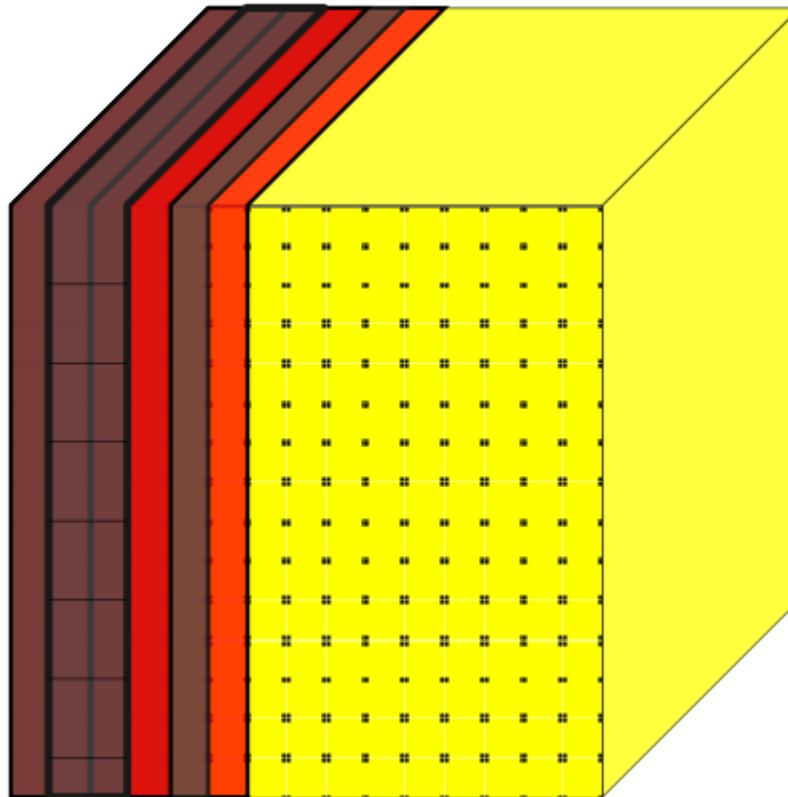
Temporal Blocking

- smoother
calculate residual } temporal blocking



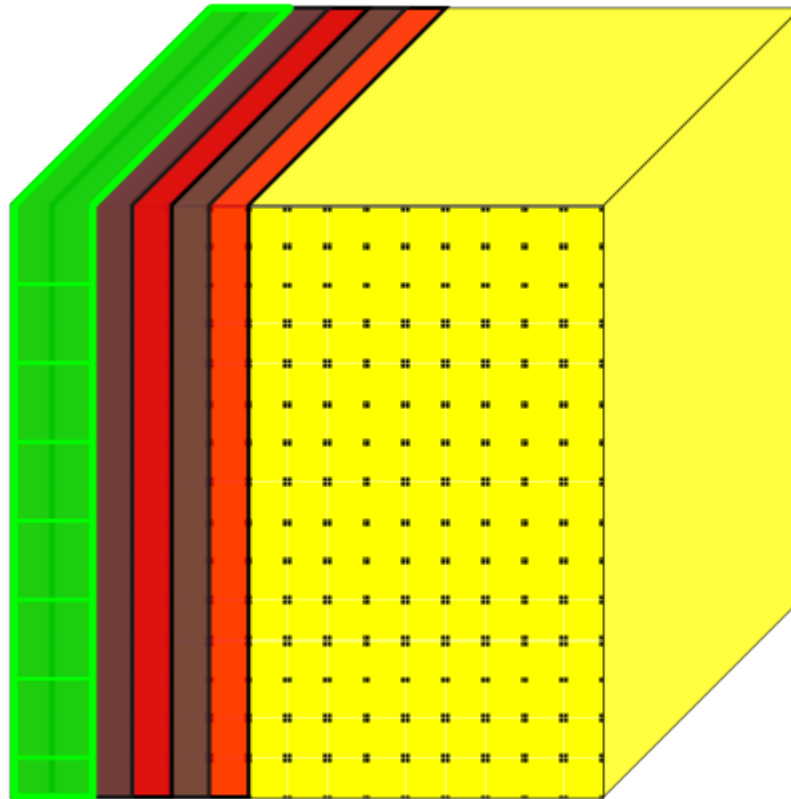
Temporal Blocking

- smoother
calculate residual } temporal blocking



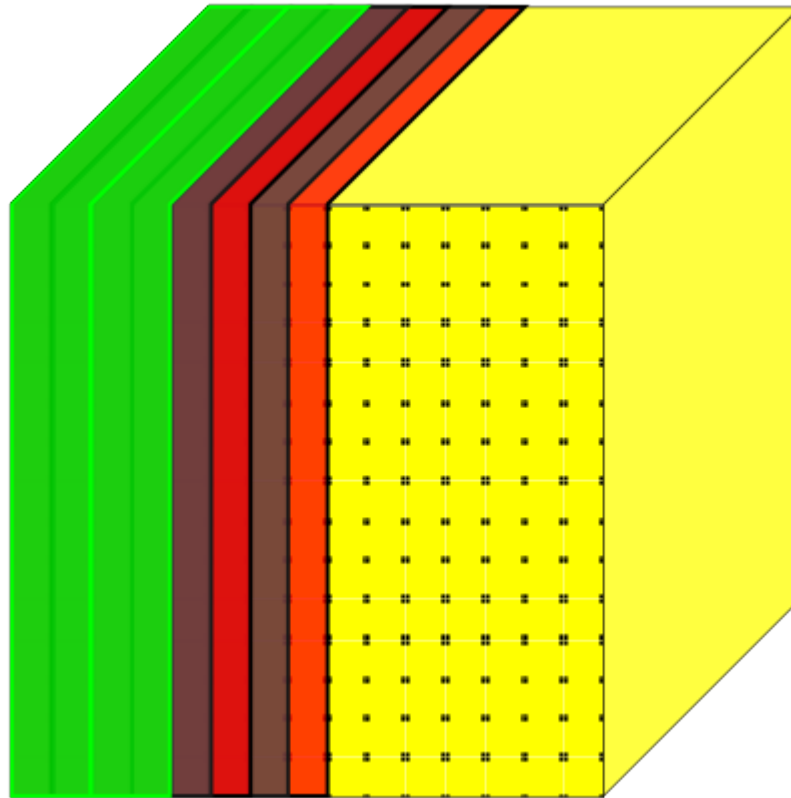
Temporal Blocking

- smoother
calculate residual } temporal blocking



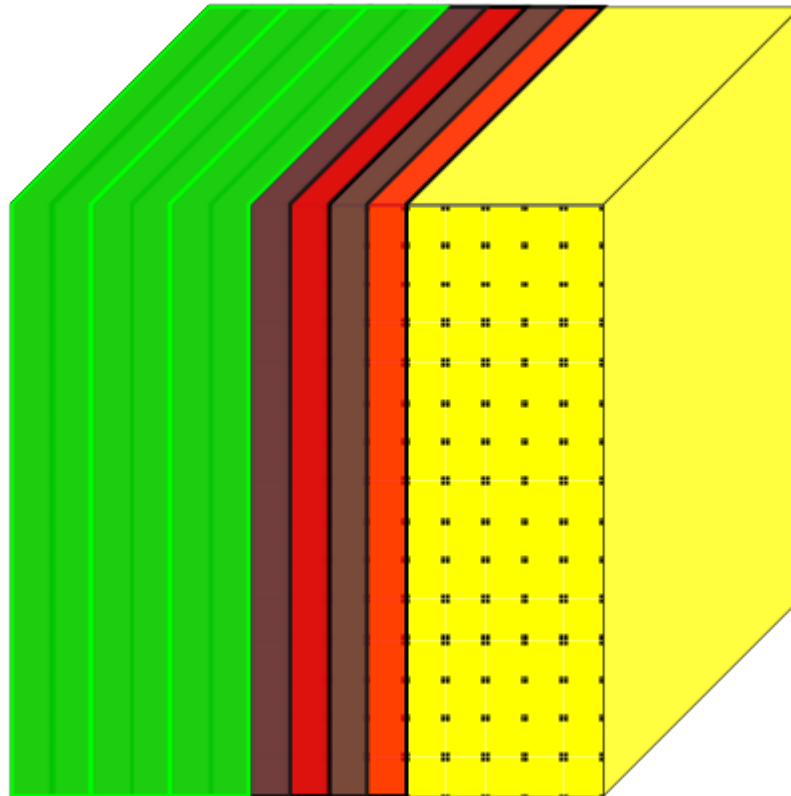
Temporal Blocking

- smoother
calculate residual } temporal blocking



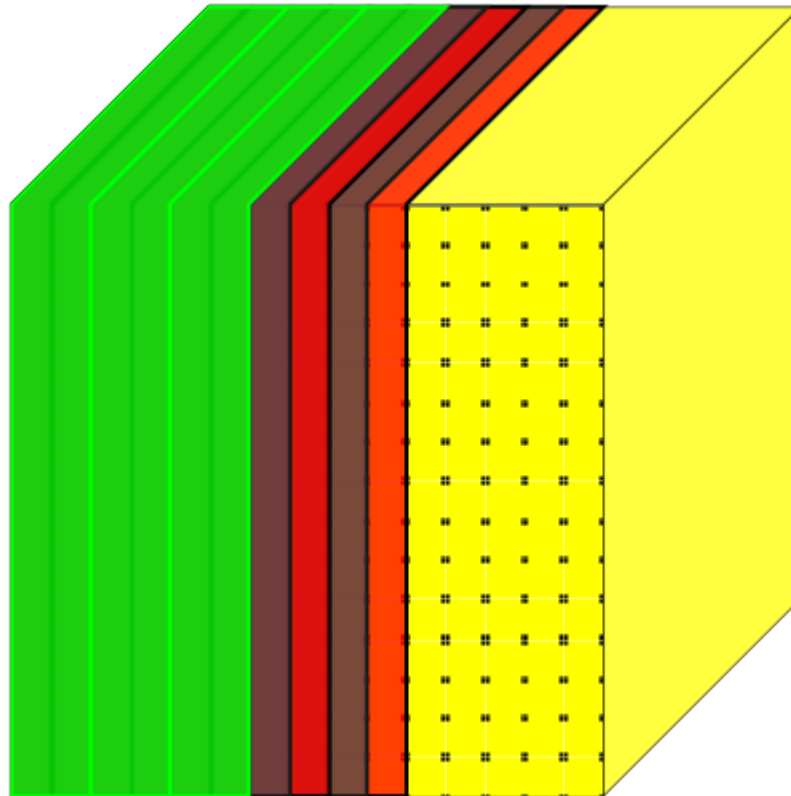
Temporal Blocking

- smoother
calculate residual } temporal blocking



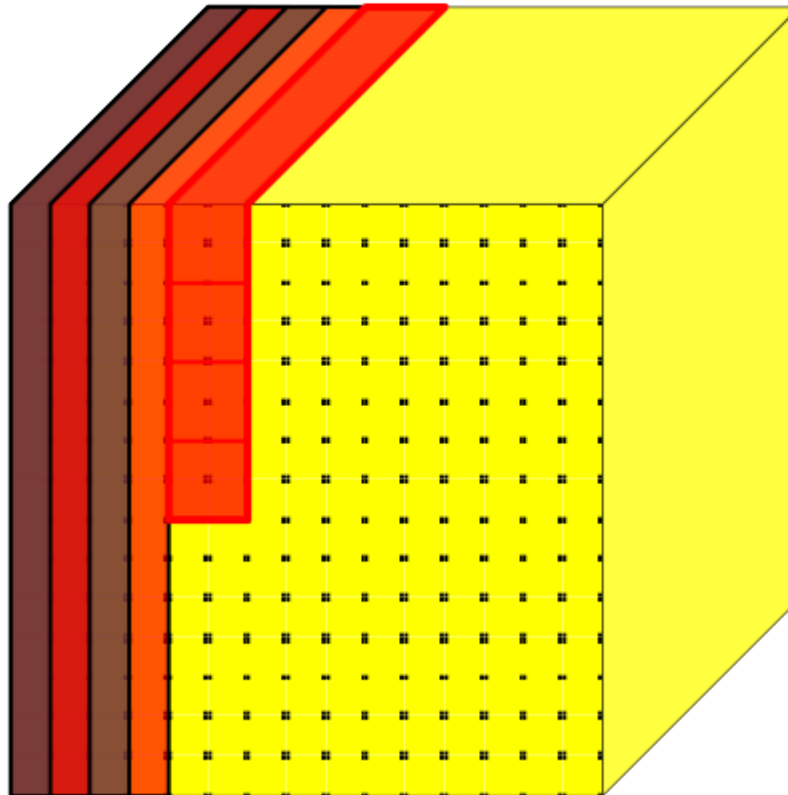
Temporal Blocking

- smoother
calculate residual } temporal blocking
- What to do for larger planes?



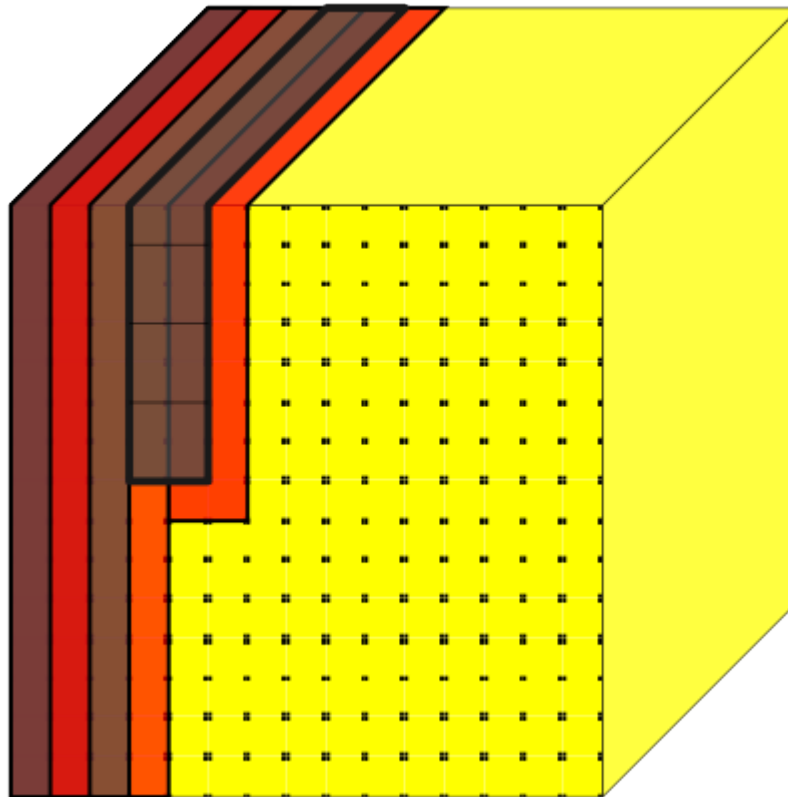
Temporal Blocking with super-blocks

- smoother
calculate residual } temporal blocking
- additional blocking level



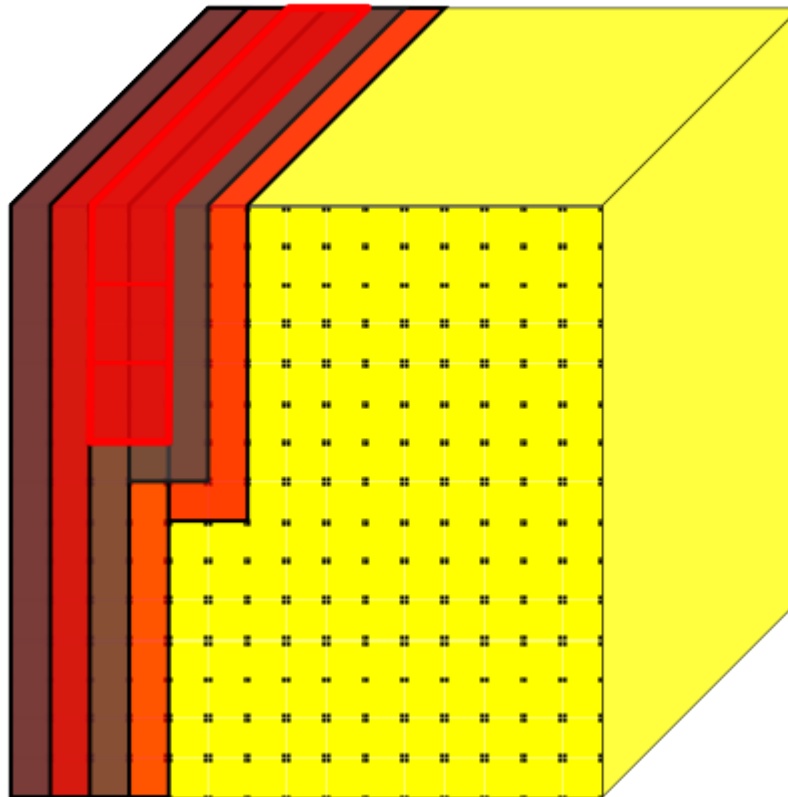
Temporal Blocking with super-blocks

- smoother
calculate residual } temporal blocking
- additional blocking level



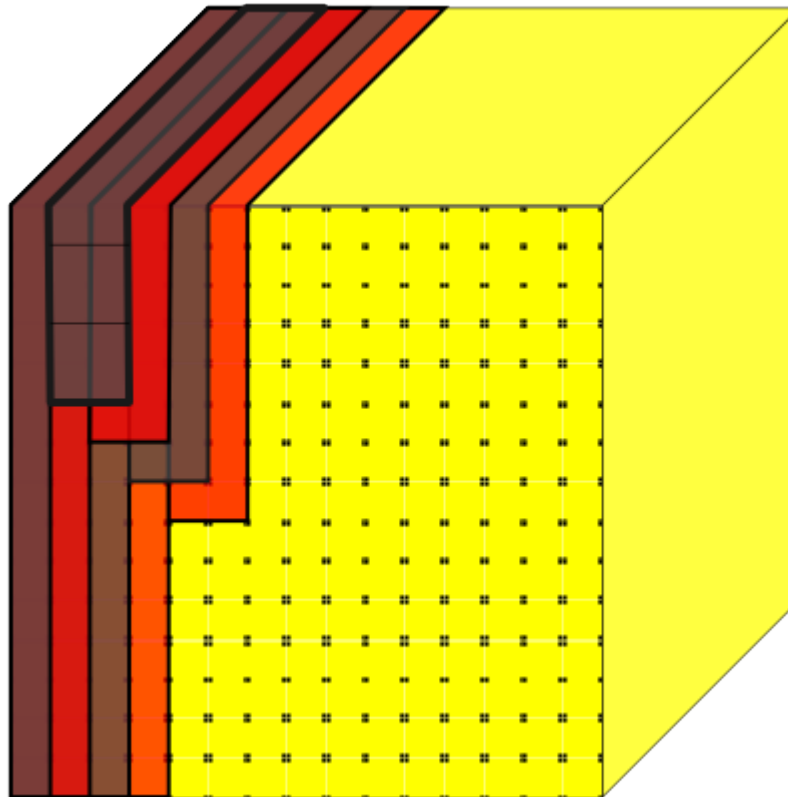
Temporal Blocking with super-blocks

- smoother
calculate residual } temporal blocking
- additional blocking level



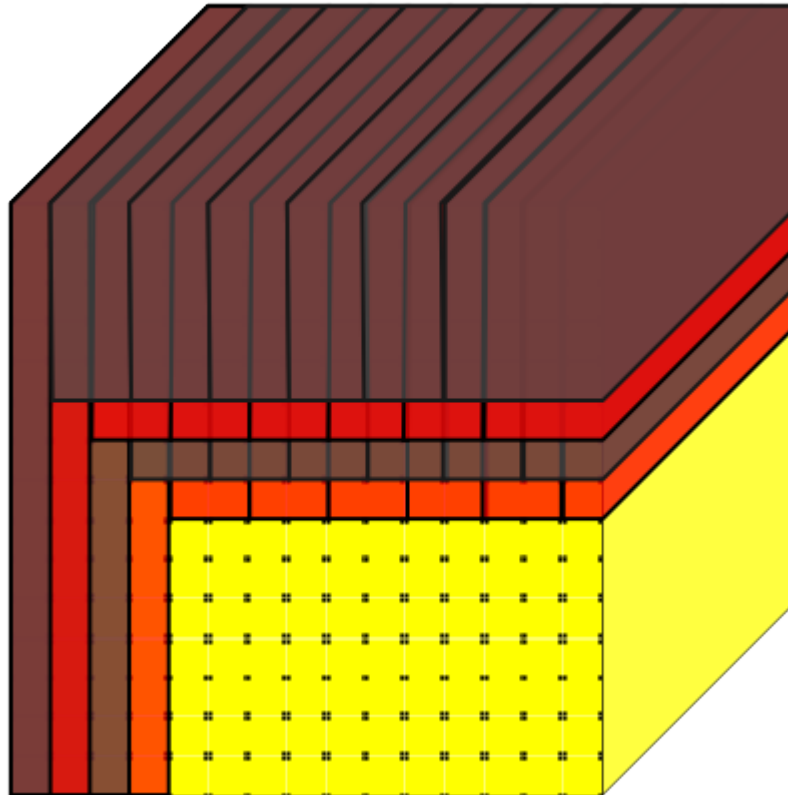
Temporal Blocking with super-blocks

- smoother
calculate residual } temporal blocking
- additional blocking level



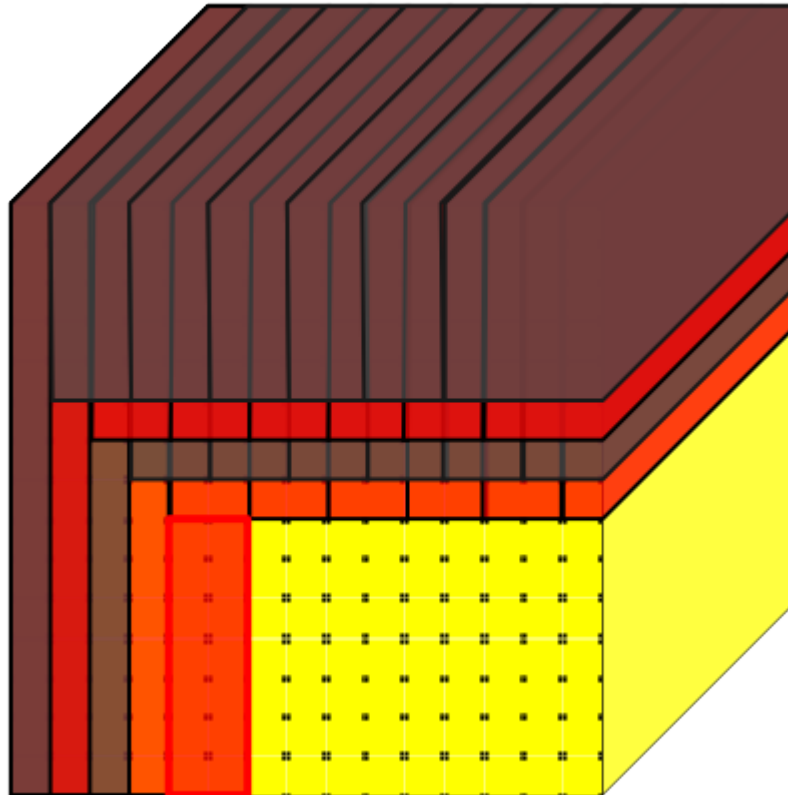
Temporal Blocking with super-blocks

- smoother
calculate residual } temporal blocking
- additional blocking level



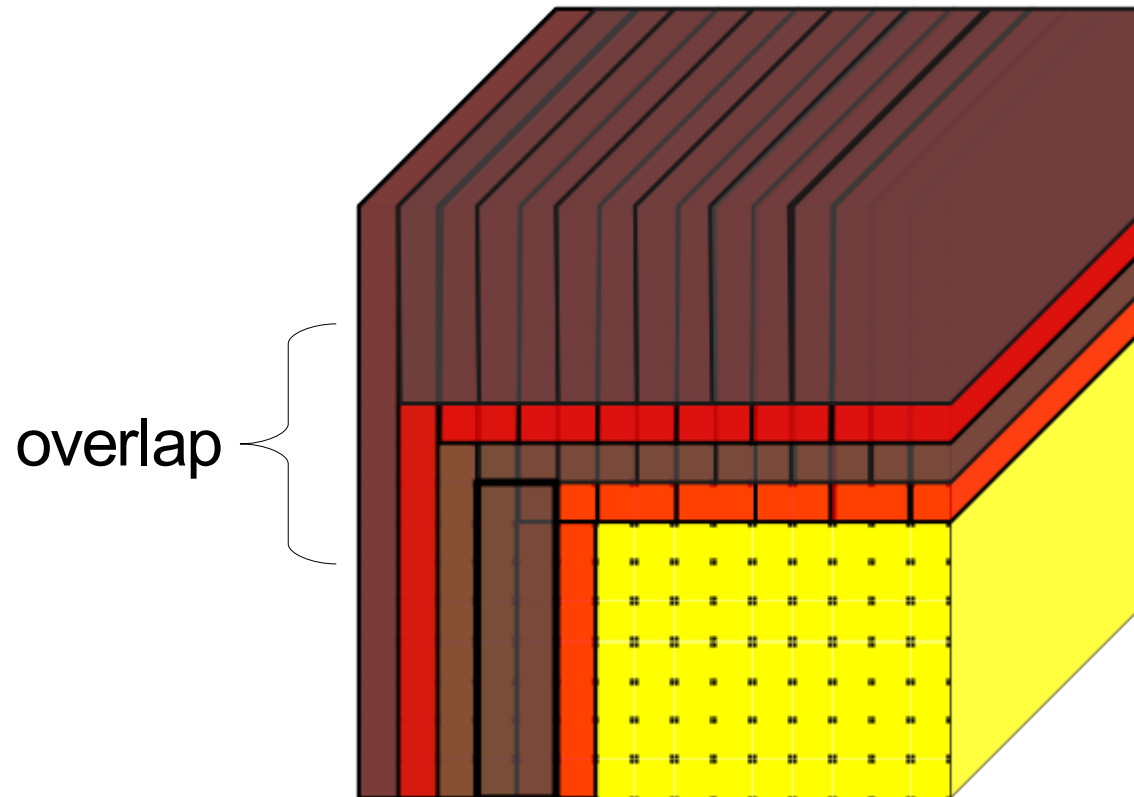
Temporal Blocking with super-blocks

- smoother
calculate residual } temporal blocking
- additional blocking level



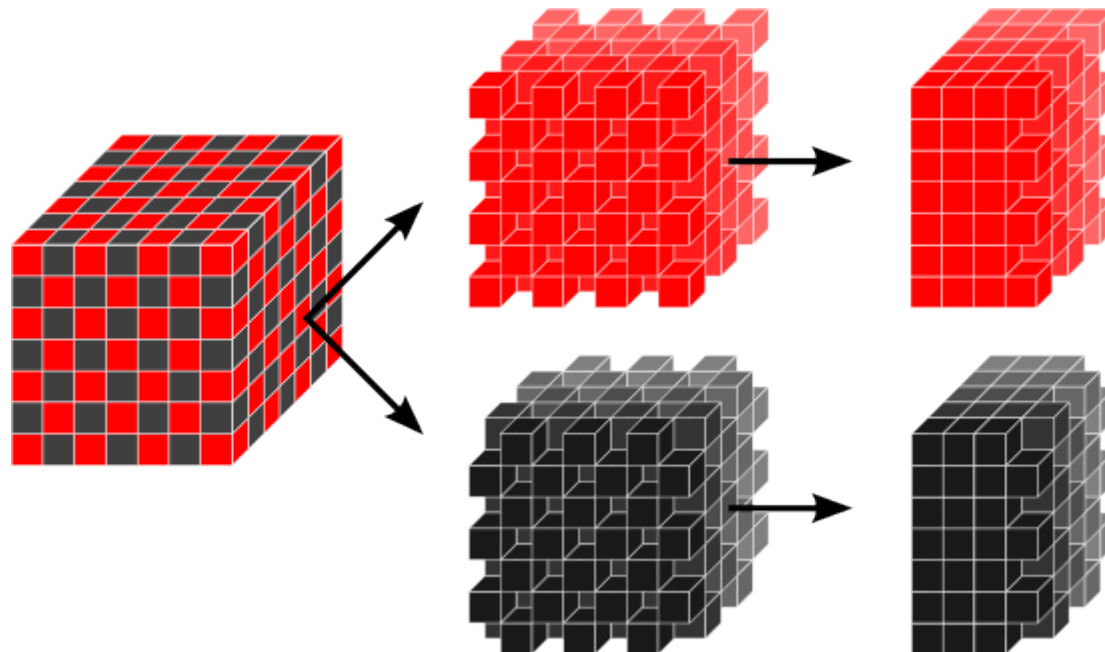
Temporal Blocking with super-blocks

- smoother
calculate residual } temporal blocking
- additional blocking level
- can be used for thread level parallelization



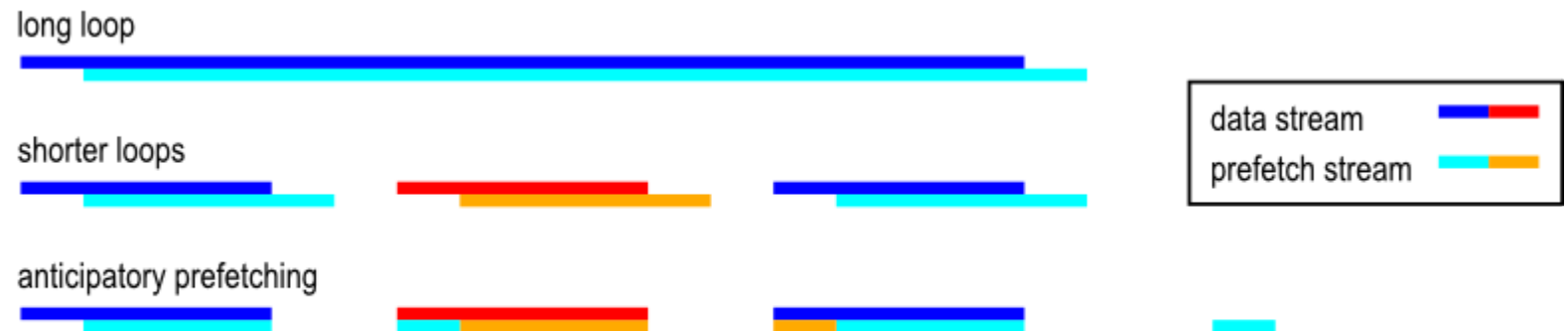
Memory Layout

- separates “red” and “black” points
- reduces memory transfer
 - interpolation/correction and restriction operate on half as much fine grid data
 - lower inter-cache transfer for blocked smoother
- more complicated address calculation



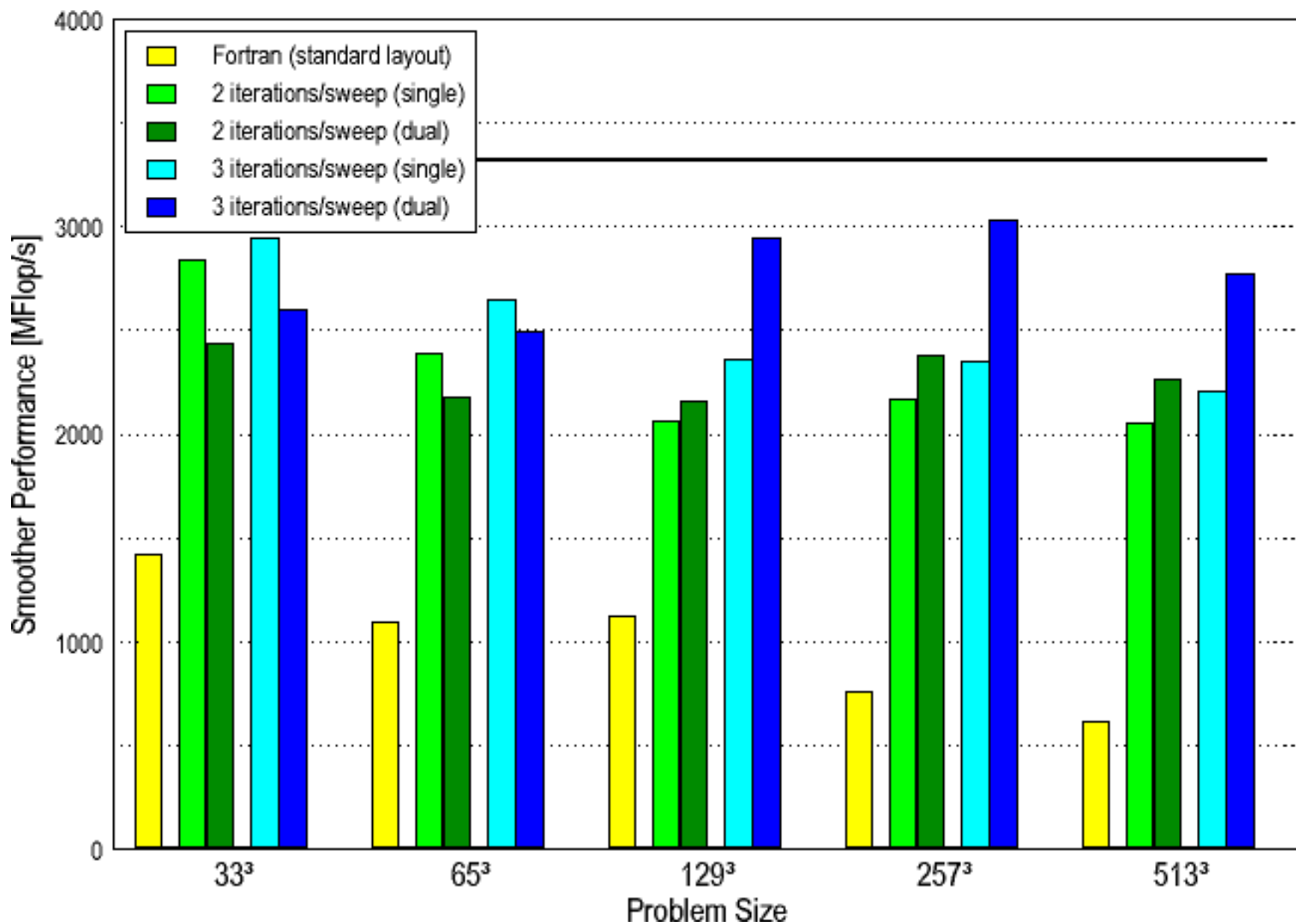
Prefetching on IA-64

- anticipatory prefetching

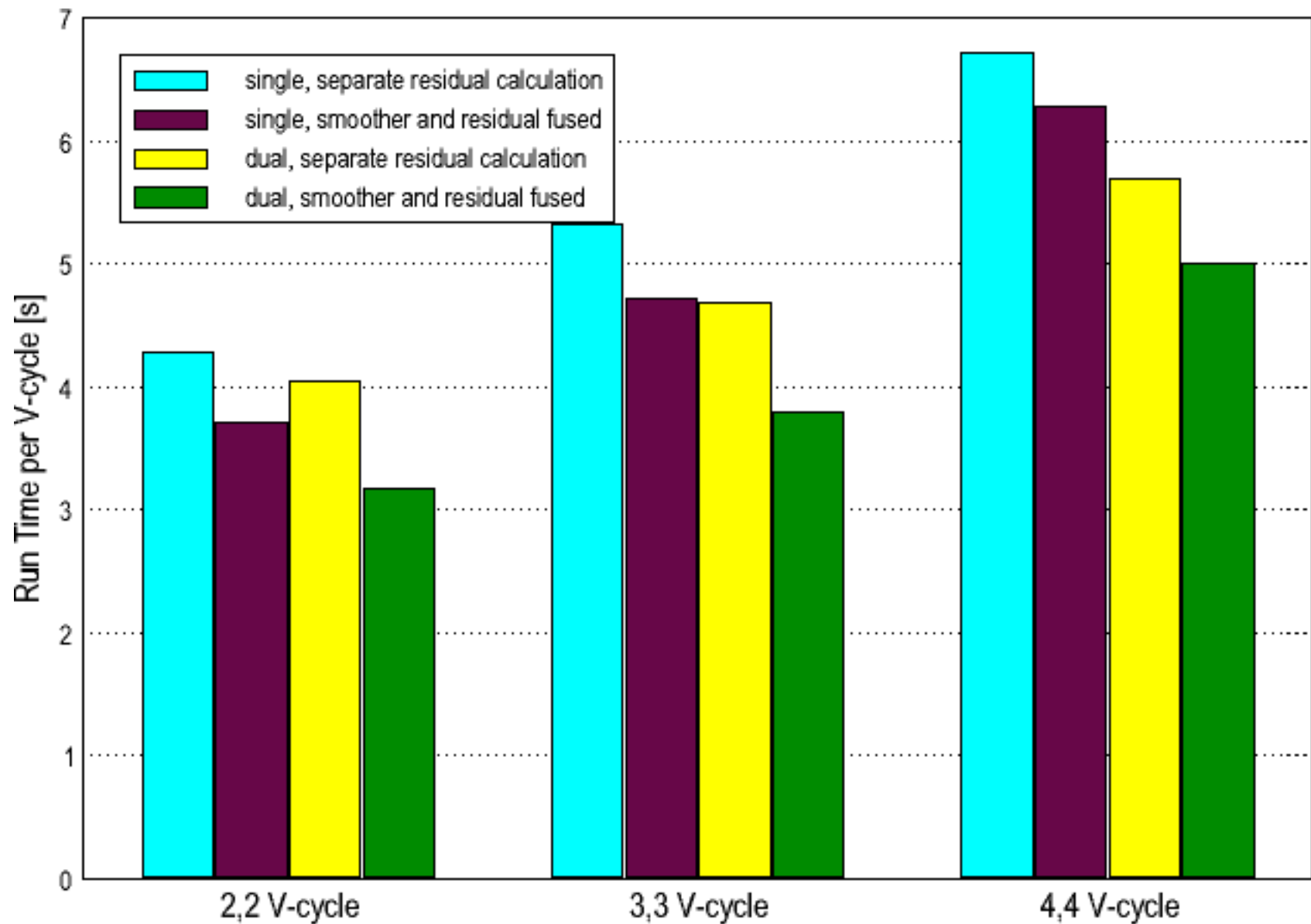


- extended prefetching for RBGS-smoother
 - blocking methods suffers from memory bursts
 - prefetching of data for next block is distributed over current blocked calculation
 - results in smoother memory throughput

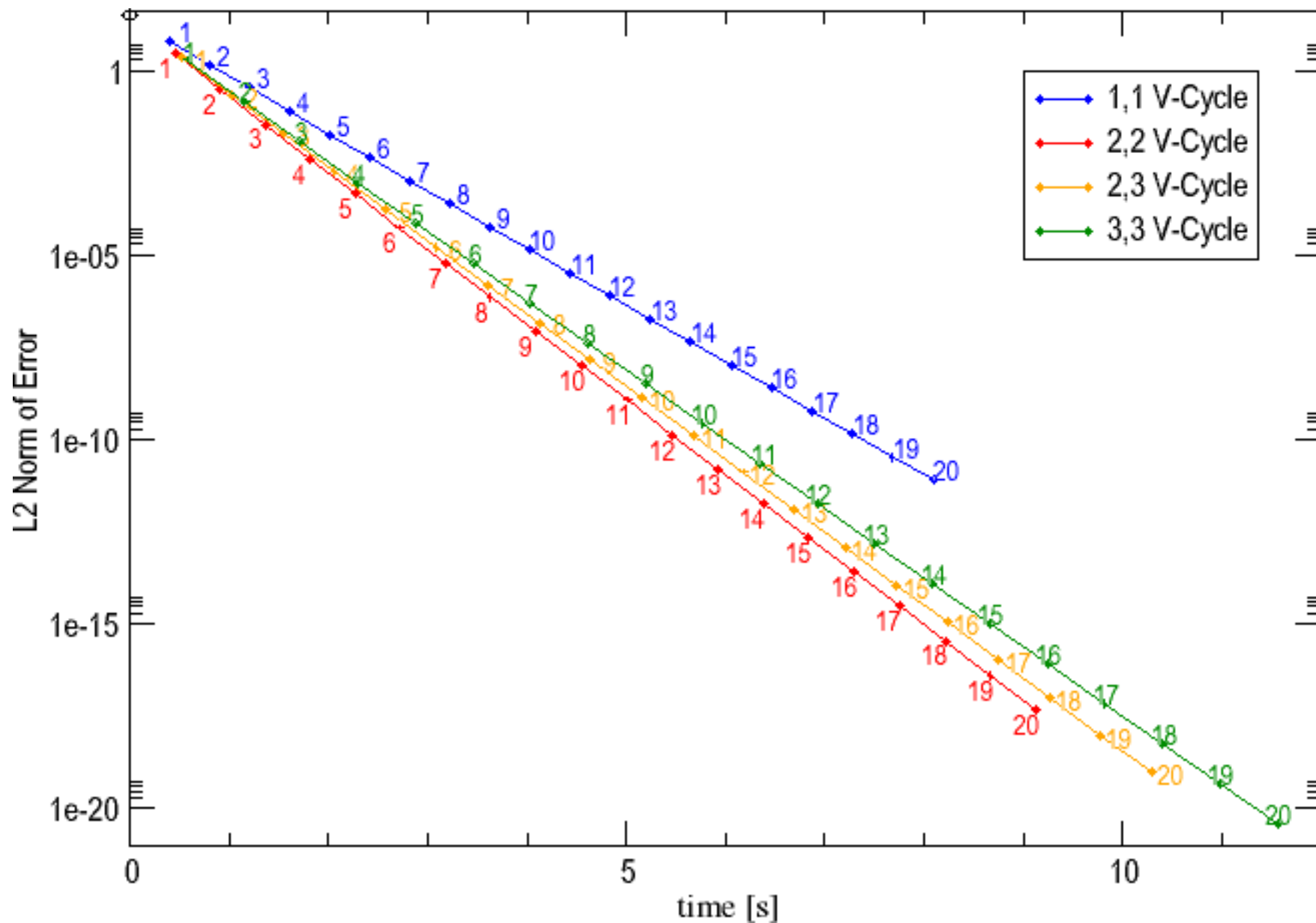
Smoother Performance



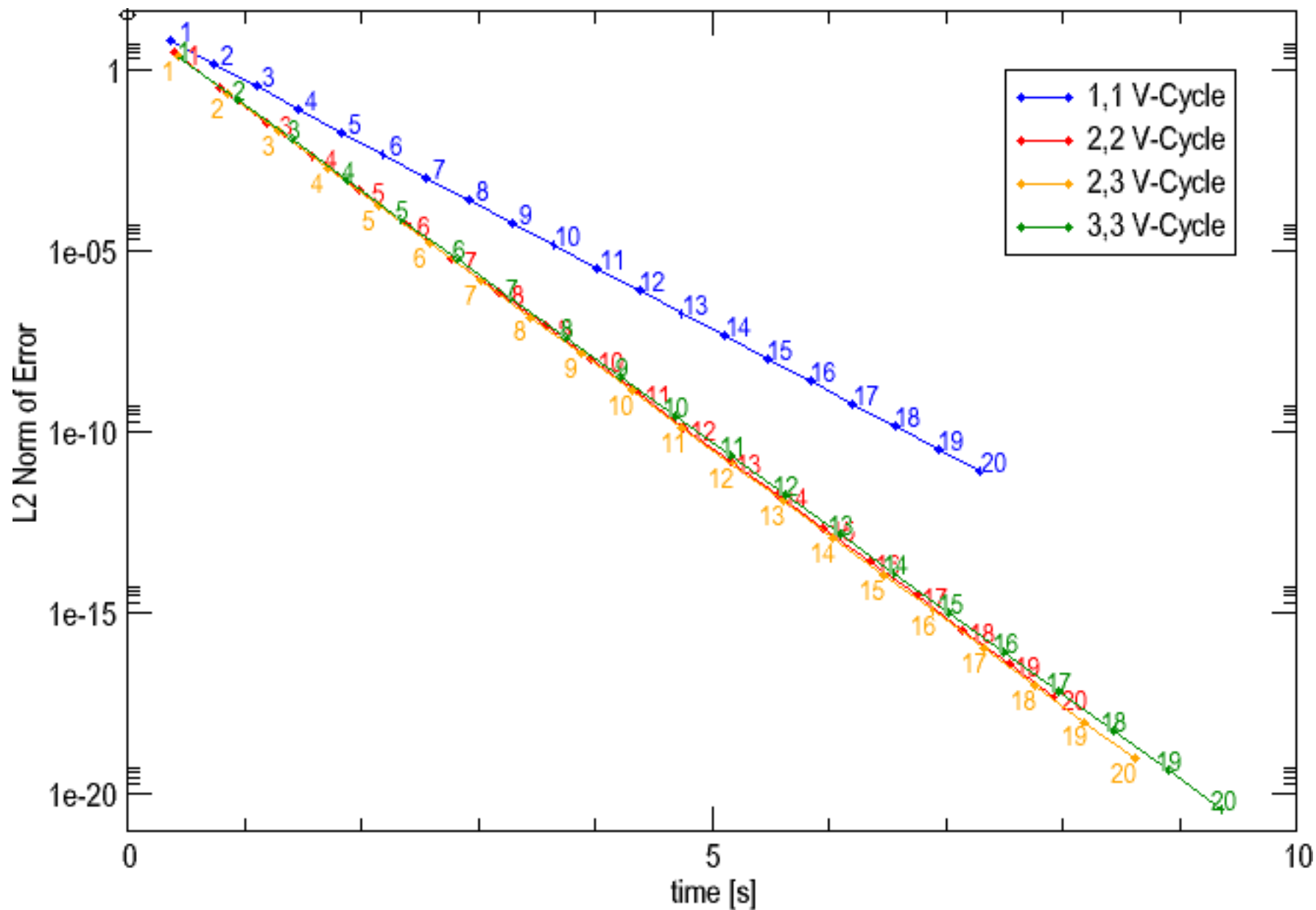
V-Cycle Performance



Convergence - 257^3 on single CPU



Convergence - 257^3 on two CPUs



Conclusions

- Itanium 2 has huge potential but is hard to program
- optimization techniques must be tuned for specific platform
- speed-up of factor 3 to 5 possible
- optimal parameters for algorithm can depend on machine configuration
- elaborate prefetching can dramatically improve performance of blocked code

