What are structured grids?

- Data is arranged in regular multidimensional grids (usually 2-4 dimensions)
- Computation is a series of grid update steps
- Neighbor addressing is implicit on each point’s coordinates
- For a given point, a stencil is a pre-determined set of nearest neighbors (possibly including itself)
- A stencil code updates every point in a regular grid with a common stencil

What is Autotuning?

- There are too many complex architectures with too many possible code transformations to optimize over
- An optimization on one machine may slow another machine down
- Need a general, automatic solution

Code Generators
- Stencil-specific
- Pert script generates 1000s of code variations
- Autotuner searches over all possible implementations (sometimes guided by a performance model to prune the space) to find the optimal configuration
- Optimizations included in this work:
  - Array Padding: avoids conflicts in the L1,L2
  - Vectorization: avoids rolling the TLB
  - Unrolling/DLP: compensates for poor compilers
  - SW Prefetching: attempts to hide L2 and DRAM latency
  - SIMDization: compensates for poor compilers, and streaming stores minimize memory traffic

Architectures Evaluated
- 2.33GHz Intel Xeon (Clovertown)
- 2.2GHz AMD Opteron
- 1.4GHz Sun Niagara2 (Huron)
- 3.2GHz 8M Cell Blade (Q520)

Paper Reference

Introduction
Investigation of stencil optimizations for a simple 7-point heat equation
- Constructed memory models for single-timestep cache blocking and time skewed blocking

Single-Timestep Cache Blocking
- 2D Cache Blocking algorithm in 3D with reuse only in space
- Largest-stride dimension is unblocked
- Early work on Pentium III-class machines by Rivera et al. showed performance improvements

Multiple-Iteration Time Skewing
- Extends cache blocking to reuse points over multiple sweeps in the grid
- Diagram above left shows the shape of each block and the order they are executed in 3D. Above right shows a 3D version of the block execution order.
- Block shape takes into account the inter-point dependencies of the stencil

Cache Oblivious
- Recursive algorithm that does not use cache size as a parameter
- Cuts a spacetime trapezoid such as the 1D example in Figure (a) in either time (c) or space (b), preserving point dependencies
- Extensive effort by our group to optimize this algorithm

Circular Queue
- Designed to pipeline planes of a stencil into a local store or cache and perform stencil operations
- Originally for Cell local stores using DMA operations

Overall Results

Stencill Performance (GFlops)

Opteron

Clovertown

Niagara2

Cell Blade

Scalability and Performance Comparison
- Clovertown has problems with both multicore and multisocket scaling
- Niagara2 delivered performance between Opteron and Clovertown
- Despite being heavily bound by double precision, Cell is far by the fastest

System Power Efficiency
- Used a digital power meter to measure sustained system power
- Niagara2 system required 50% more power than other systems