Automatic Performance Tuning
Spiral, FFTW, ATLAS & Friends

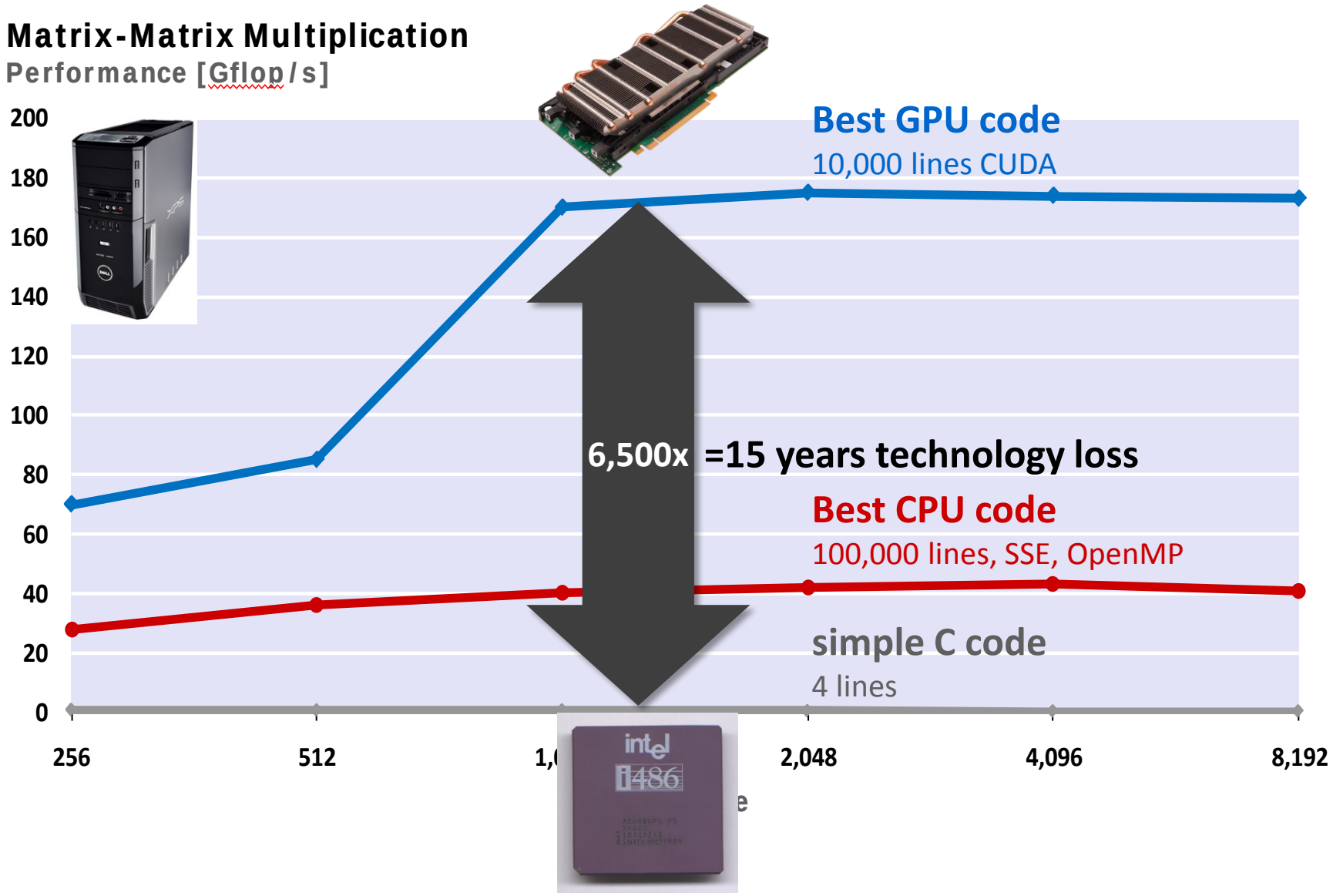
Franz Franchetti

ECE, Carnegie Mellon University
www.ece.cmu.edu/~franzf

Co-Founder, SpiralGen
www.spiralgen.com

The work was sponsored by Defense Advanced Research Projects Agency (DARPA) under agreement No. HR0011-13-2-0007. The content, views and conclusions presented in this document do not necessarily reflect the position or the policy of DARPA or the U.S. Government. No official endorsement should be inferred.

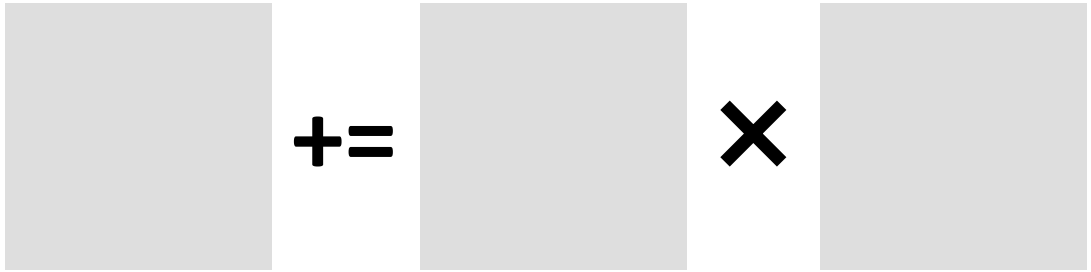
The Cost Of Portability and Maintainability



Compilers: The Fundamental Problem

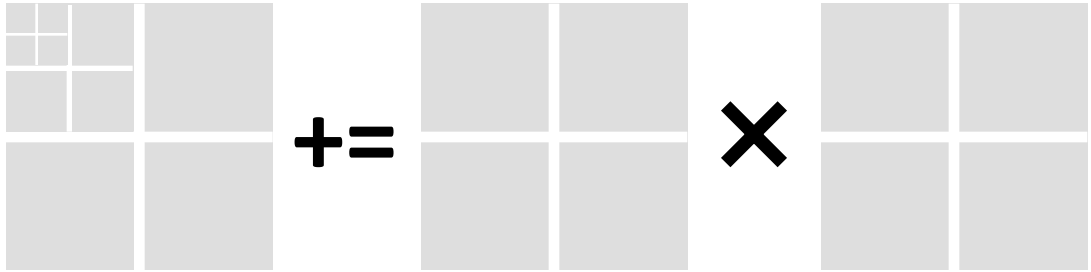
Matrix-matrix multiplication

```
for i=1:N
  for j=1:N
    for k=1:N
      C[i,j] += A[i,k]*B[k,j]
```



Tiled matrix-matrix multiplication

```
for i=1:NB:N
  for j=1:NB:N
    for k=1:NB:N
      for i0=i:NU:i+NB
        for j0=j:NU:j+NB
          for k0=k:NU:k+NB
            for k00=k0:1:k0+NU
              for j00=j0:1:j0+NU
                for i00=i0:1:i0+NU
                  C[i00,j00] +=
                    A[i00,k00]*B[k00,j00]
```

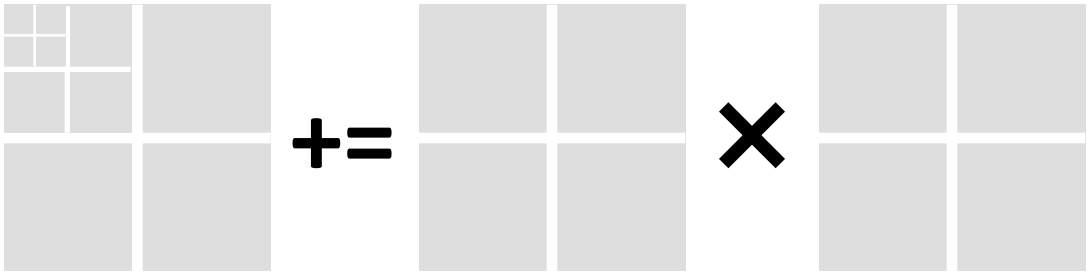


Problem: which transformation order? What parameter values?

How to Overcome Compiler Limitations?

- **Autotuning**

  Write parameterized program and empirically find good parameters



- **Program generation**

  Automatically generate big basic blocks and special case code

  Use special instructions (SSE/AVX/NEON/AltiVec/QPX,…)

```
for (i = 0; i < 4; i++)
   A[i] = B[i] * C[i];
```

```
A[0] = B[0] * C[0];
A[1] = B[1] * C[1];
A[2] = B[2] * C[2];
A[3] = B[3] * C[3];
```

When done: 30MB of source code for DGEMM

Autotuning: From Simple to "Really Hard"

- **Level 0: simple C program**
  implements the algorithm cleanly
- **Level 1: C macros plus search script**
  use C preprocessor for meta-programming
- **Level 2: scripting for code specialization**
  text-based program generation, e.g., ATLAS
- **Level 3: add compiler technology**
  internal code representation, e.g., FFTW's genfft
- **Level 4: synthesize the program from scratch**
  high level representation, e.g., TCE and Spiral

Performance Portability Landscape

Synthesis from Domain Math

- **Spiral**
  Signal and image processing, SDR
- **Tensor Contraction Engine**
  Quantum Chemistry Code Synthesizer
- **FLAME**
  Numerical linear algebra (LAPACK)

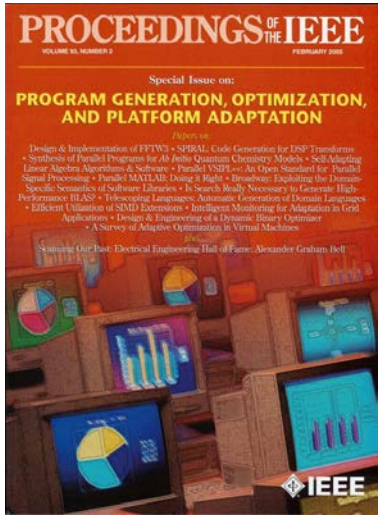
Compiler Synthesis/Autotuning

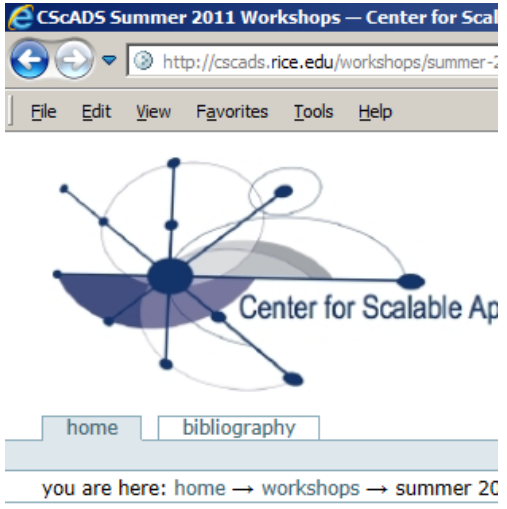
- **Polyhedral framework**
  XL C, Pluto, CHiLL
- **Transformation prescription**
  CHiLL, POET
- **Profile guided optimization**
  Intel C, IBM XL

Numerical Libraries

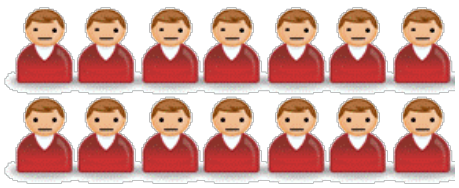
- **ATLAS**
  BLAS generator
- **FFTW**
  kernel generator
- **Vendor math libraries**
  Code generation scripts

Autotuning Primer

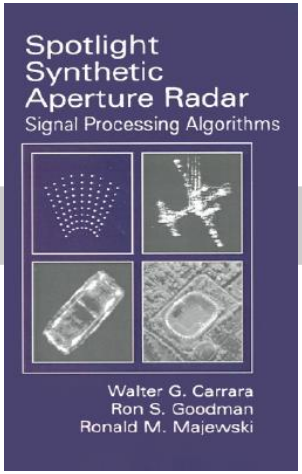




Spiral: Computer Writes Best SAR Code



Programming



Synthesis

Result

Same performance, 1/10th human effort, non-expert user

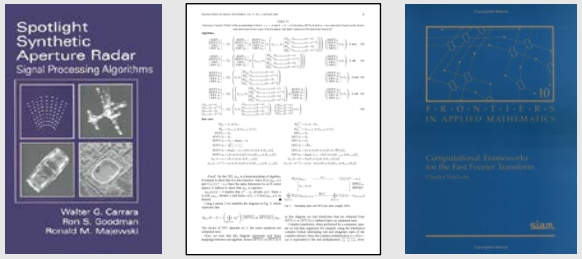
Key ideas

restrict domain, use mathematics, performance portability

D. McFarlin, F. Franchetti, M. Püschel, and J. M. F. Moura: **High Performance Synthetic Aperture Radar Image Formation On Commodity Multicore Architectures.** in Proceedings SPIE, 2009.

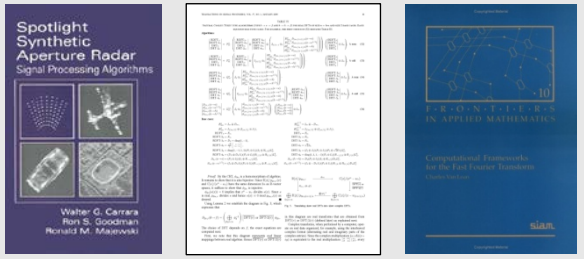
What is Spiral?

Traditionally



High performance library optimized for given platform

Spiral Approach



Spiral

High performance library optimized for given platform

Comparable performance

Algorithms: Rules in Domain Specific Language

Linear Transforms

$$
\begin{aligned}
\mathbf{DFT}_n &\rightarrow (\mathbf{DFT}_k \otimes \mathrm{I}_m)\, \mathrm{T}^n_m (\mathrm{I}_k \otimes \mathbf{DFT}_m)\, \mathrm{L}^n_k, \quad n = km \\
\mathbf{DFT}_n &\rightarrow P_n (\mathbf{DFT}_k \otimes \mathbf{DFT}_m) Q_n, \quad n = km,\ \gcd(k,m) = 1 \\
\mathbf{DFT}_p &\rightarrow R_p^T (\mathrm{I}_1 \oplus \mathbf{DFT}_{p-1}) D_p (\mathrm{I}_1 \oplus \mathbf{DFT}_{p-1}) R_p, \quad p \text{ prime} \\
\mathbf{DCT\text{-}3}_n &\rightarrow (\mathrm{I}_m \oplus \mathrm{J}_m)\, \mathrm{L}^n_m (\mathbf{DCT\text{-}3}_m(1/4) \oplus \mathbf{DCT\text{-}3}_m(3/4)) \\
&\quad \cdot (\mathrm{F}_2 \otimes \mathrm{I}_m) \begin{bmatrix} \mathrm{I}_m & 0 \oplus -\mathrm{J}_{m-1} \\ & \frac{1}{\sqrt{2}}(\mathrm{I}_1 \oplus 2\,\mathrm{I}_m) \end{bmatrix}, \quad n = 2m \\
\mathbf{DCT\text{-}4}_n &\rightarrow S_n \mathbf{DCT\text{-}2}_n\, \mathrm{diag}_{0 \le k < n}(1/(2\cos((2k+1)\pi/4n))) \\
\mathbf{IMDCT}_{2m} &\rightarrow (\mathrm{J}_m \oplus \mathrm{I}_m \oplus \mathrm{I}_m \oplus \mathrm{J}_m) \left(\begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \mathrm{I}_m \right) \oplus \left(\begin{bmatrix} -1 \\ -1 \end{bmatrix} \otimes \mathrm{I}_m \right) \mathrm{J}_{2m}\, \mathbf{DCT\text{-}4}_{2m} \\
\mathbf{WHT}_{2^k} &\rightarrow \prod_{i=1}^{t} (\mathrm{I}_{2^{k_1+\cdots+k_{i-1}}} \otimes \mathbf{WHT}_{2^{k_i}} \otimes \mathrm{I}_{2^{k_{i+1}+\cdots+k_t}}), \quad k = k_1 + \cdots + k_t \\
\mathbf{DFT}_2 &\rightarrow \mathrm{F}_2 \\
\mathbf{DCT\text{-}2}_2 &\rightarrow \mathrm{diag}(1, 1/\sqrt{2})\, \mathrm{F}_2 \\
\mathbf{DCT\text{-}4}_2 &\rightarrow \mathrm{J}_2\, \mathrm{R}_{13\pi/8}
\end{aligned}
$$

Viterbi Decoding

$$
\begin{aligned}
\underbrace{\mathbf{Vit}}_{\mathrm{vec}(v)} &\rightarrow \underbrace{\left(\prod (L \times I) \circ (I \otimes C)\right) \circ Id}_{\mathrm{vec}(v)} \\
&\rightarrow \left(\prod \underbrace{(L \times I) \circ (I \otimes C)}_{\mathrm{vec}(v)} \right) \circ Id \\
\times \quad &\rightarrow \left(\prod (L \otimes I_v \times I) \circ (I \otimes C \otimes I_v) \circ (\vec{L} \times I)\right) \circ Id \\
&\rightarrow \prod (L \otimes I_v \times I) \circ (I \otimes (B \otimes I_v)) \circ (\vec{L} \times I)
\end{aligned}
$$

Matrix-Matrix Multiplication

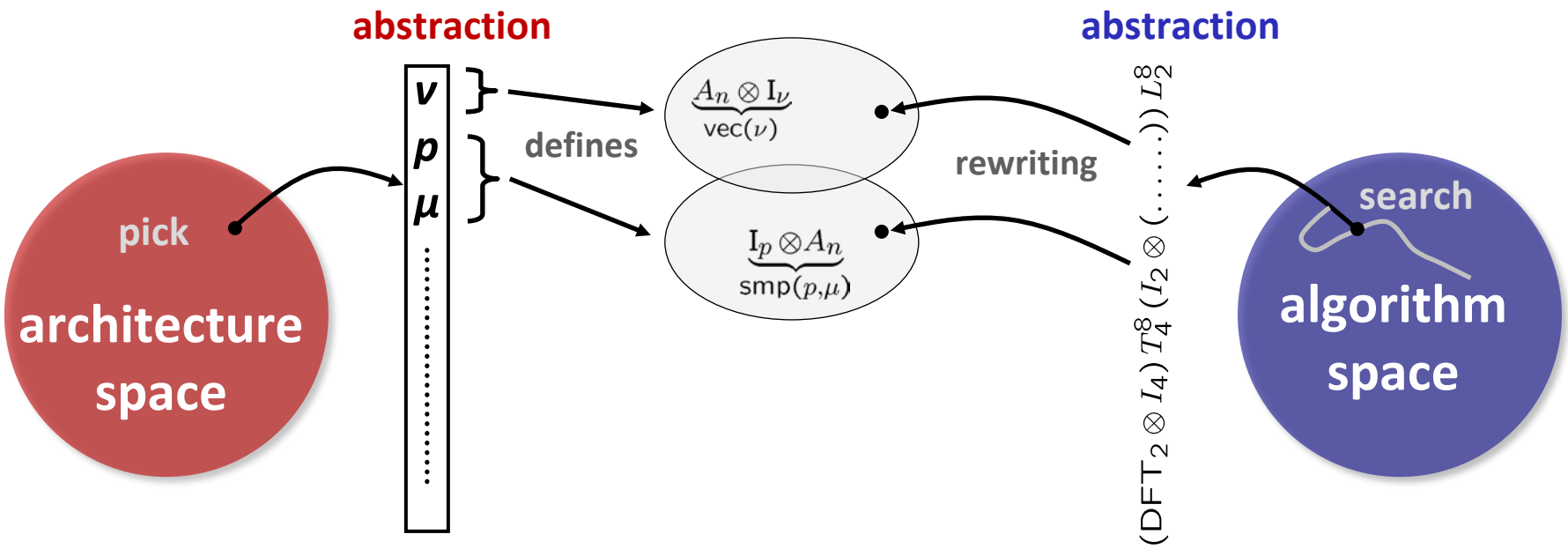
$$
\begin{aligned}
&\mathrm{MMM}_{1,1,1} \rightarrow (\cdot)_1 \\
&\mathrm{MMM}_{m,n,k} \rightarrow (\otimes)_{m/m_b \times 1} \otimes \mathrm{MMM}_{m_b,n,k} \\
&\mathrm{MMM}_{m,n,k} \rightarrow \mathrm{MMM}_{m,nb,k} \otimes (\otimes)_{1 \times n/nb} \\
&\mathrm{MMM}_{m,n,k} \rightarrow ((\Sigma_{k/k_b} \circ (\cdot)_{k/k_b}) \otimes \mathrm{MMM}_{m,n,k_b}) \circ \\
&\qquad\qquad ((L^{mk/k_b}_{k/k_b} \otimes I_{k_b}) \times I_{kn}) \\
&\mathrm{MMM}_{m,n,k} \rightarrow (L^{mn/n_b}_m \otimes I_{n_b}) \circ \\
&\qquad ((\otimes)_{1 \times n/n_b} \otimes \mathrm{MMM}_{m,n_b,k}) \circ \\
&\qquad\qquad (I_{km} \times (L^{kn/n_b}_{n/n_b} \otimes I_{n_b}))
\end{aligned}
$$

Synthetic Aperture Radar (SAR)

$$
\begin{aligned}
\mathrm{SAR}_{k \times m \rightarrow n \times n} &\rightarrow \mathbf{DFT}_{n \times n} \circ \mathrm{Interp}_{k \times m \rightarrow n \times n} \\
\mathbf{DFT}_{n \times n} &\rightarrow (\mathbf{DFT}_n \otimes \mathrm{I}_n) \circ (\mathrm{I}_n \otimes \mathbf{DFT}_n) \\
\mathrm{Interp}_{k \times m \rightarrow n \times n} &\rightarrow (\mathrm{Interp}_{k \rightarrow n} \otimes_i \mathrm{I}_n) \circ (\mathrm{I}_k \otimes_i \mathrm{Interp}_{m \rightarrow n}) \\
\mathrm{Interp}_{r \rightarrow s} &\rightarrow \left(\bigoplus_{i=0}^{n-2} \mathrm{InterpSeg}_k \right) \oplus \mathrm{InterpSegPruned}_{k,\ell} \\
\mathrm{InterpSeg}_k &\rightarrow \mathrm{G}^{u \cdot n \rightarrow k}_f \circ \mathrm{iPrunedDFT}_{n \rightarrow u \cdot n} \circ \left(\frac{1}{n}\right) \circ \mathbf{DFT}_n
\end{aligned}
$$

Spiral's Domain-Specific Program Synthesis

Model: common abstraction
= spaces of matching formulas



Architectural parameter:
Vector length,
#processors, …

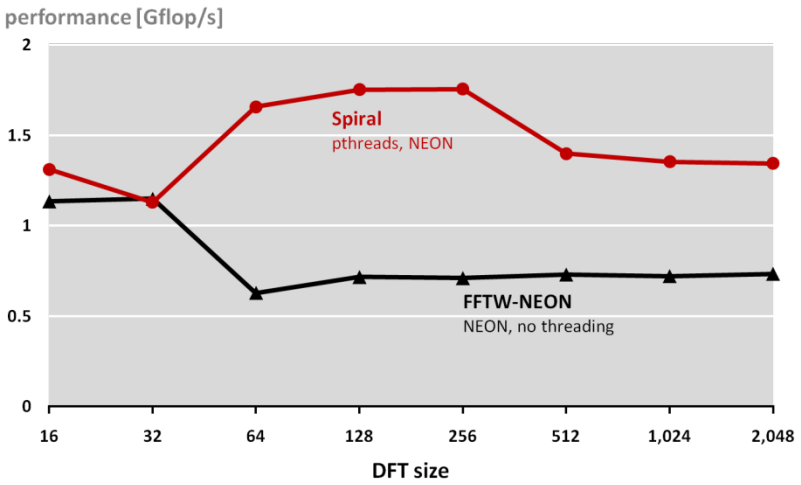
optimization

Kernel:
problem size,
algorithm choice

From Cell Phone To Supercomputer

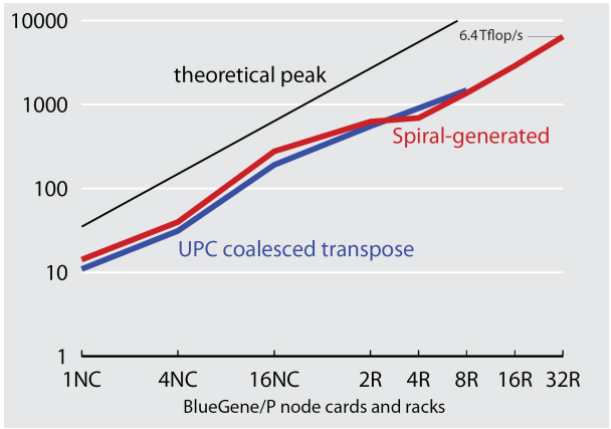
DFT on Samsung Galaxy S II

Dual-core 1.2 GHz Cortex-A9 with NEON ISA



Global FFT (1D FFT, HPC Challenge)

performance [Gflop/s]



6.4 Tflop/s on BlueGene/P

Samsung i9100 Galaxy S II

Dual-core ARM at 1.2GHz with NEON ISA

SIMD vectorization + multi-threading

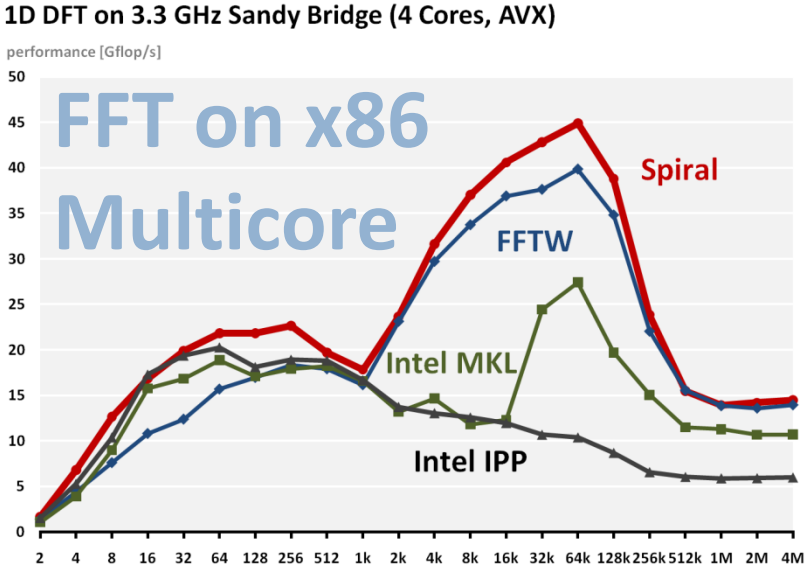
BlueGene/P at Argonne National Laboratory

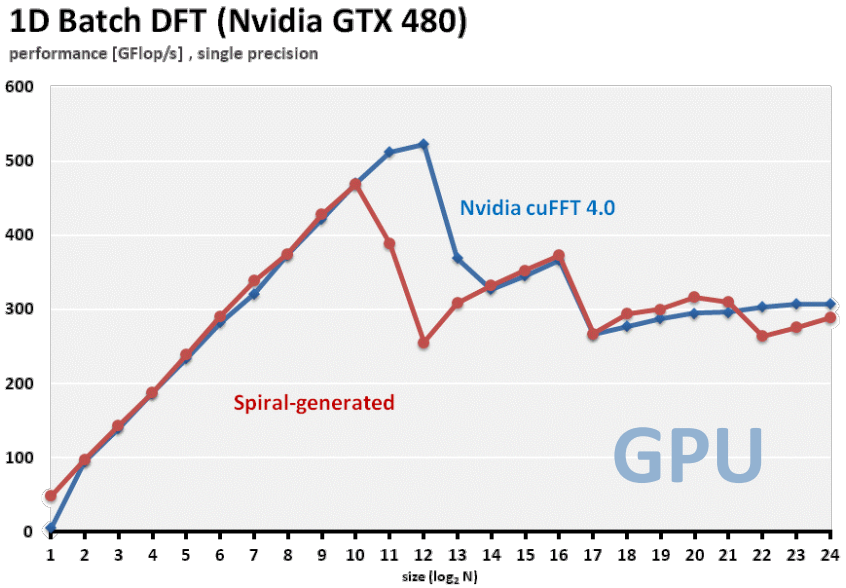
128k cores (quad-core CPUs) at 850 MHz

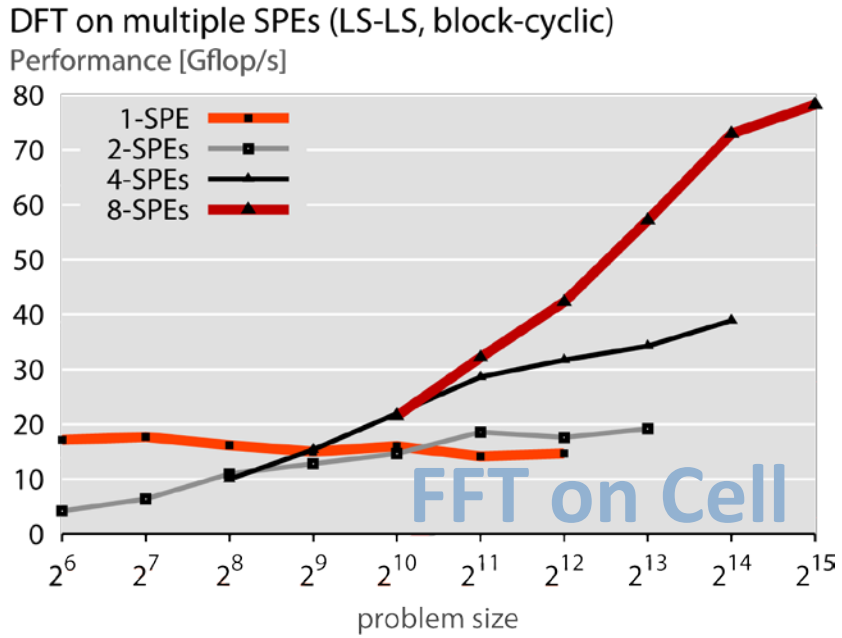
SIMD vectorization + multi-threading + MPI

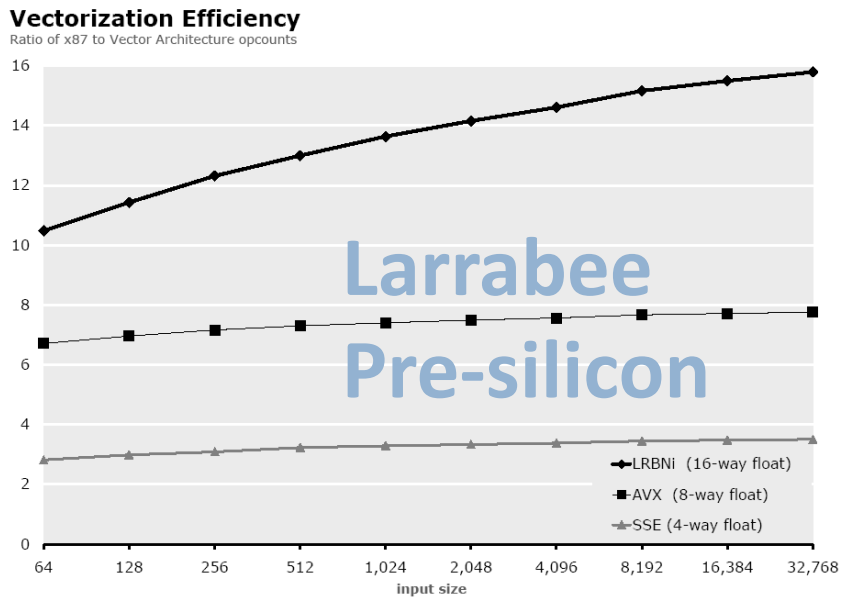
G. Almási, B. Dalton, L. L. Hu, F. Franchetti, Y. Liu, A. Sidelnik, T. Spelce, I. G. Tānase, E. Tiotto, Y. Voronenko, X. Xue:
2010 IBM HPC Challenge Class II Submission. Winner of the 2010 HPC Challenge Class II Award (Most Productive System).

Results: Spiral Outperforms Humans





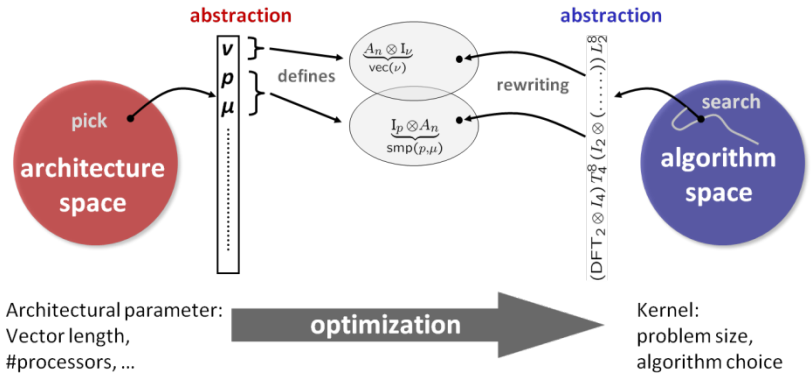




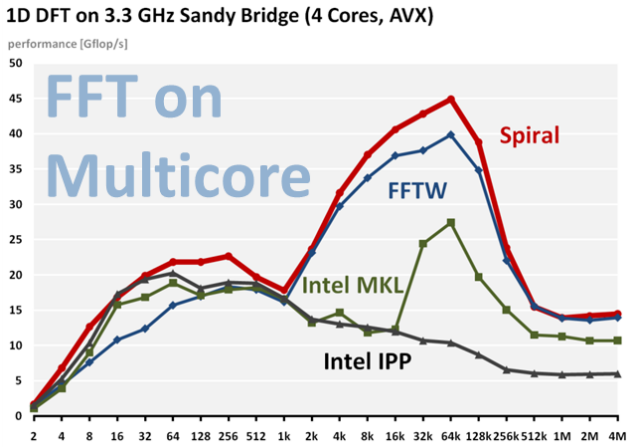
Summary: Spiral in a Nutshell

Joint Abstraction

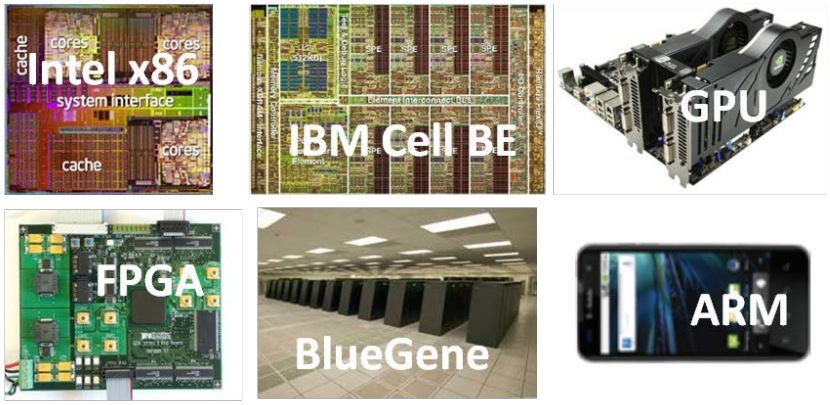
Model: common abstraction
= spaces of matching formulas



Outperforms Human Experts



Performance Portability



Application Domains

Signal Processing

$$\mathrm{DFT}_n \;\rightarrow\; (\mathrm{DFT}_k \otimes \mathrm{I}_m)\, \mathrm{T}^n_m (\mathrm{I}_k \otimes \mathrm{DFT}_m)\, \mathrm{L}^n_k,$$

Matrix Algorithms

Software Defined Radio

Image Formation (SAR)

Academic @ CMU: www.spiral.net
Commercial: www.spiralgen.com