SPIRAL
www.spiral.net

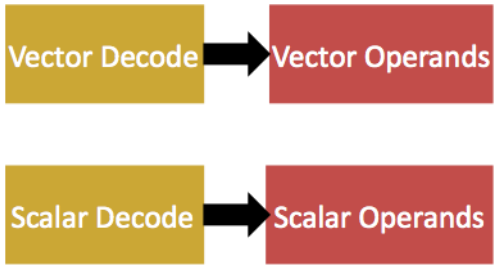
High Performance SAR Image Formation On Commodity Multicore Architectures

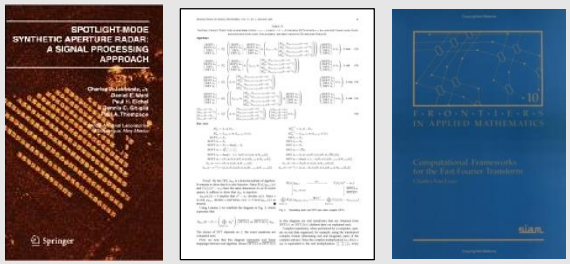
Daniel S. McFarlin
Franz Franchetti
Markus Püschel
José M. F. Moura

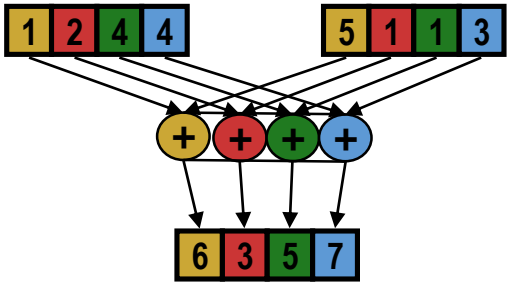
Carnegie Mellon

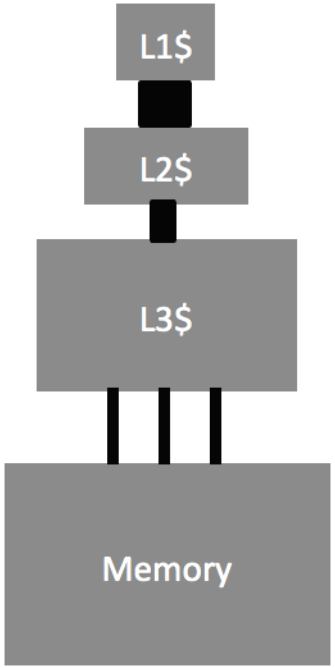
Challenge: High Performance Mapping of Algorithms to Highly Parallel Hardware

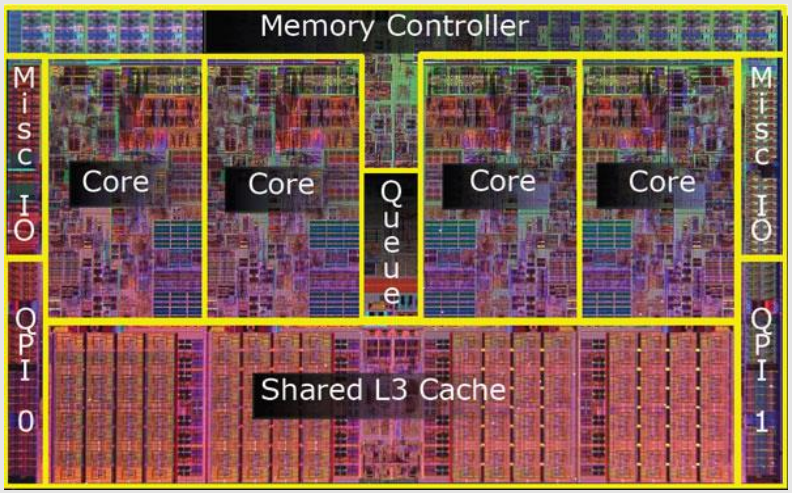
SPIRAL
www.spiral.net

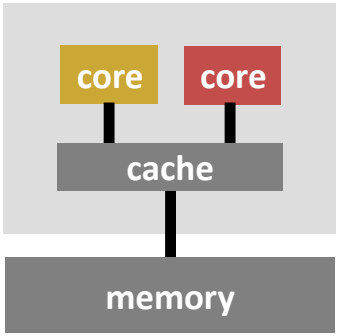












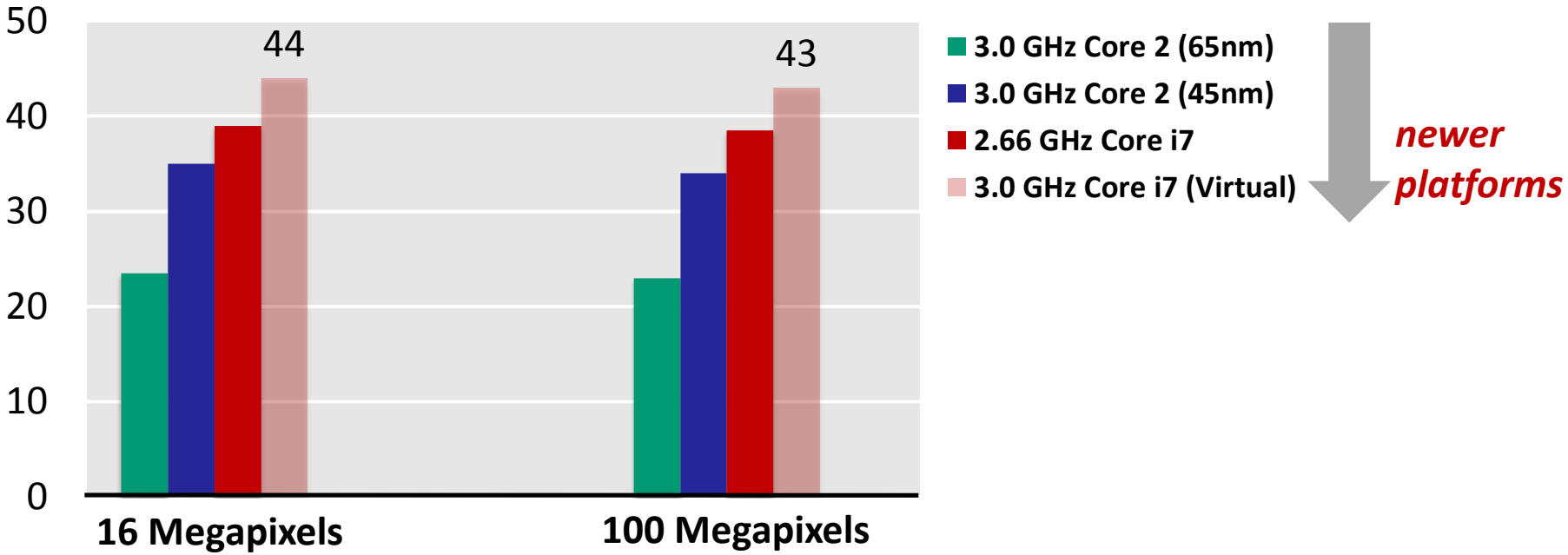
Difficult for General Purpose C Toolchain

Spiral's Automatically Generated PFA SAR Image Formation Code

SPIRAL
www.spiral.net

SAR Image Formation on Intel platforms

performance [Gflop/s]



- Algorithm by J. Rudin (best paper award, HPEC 2007): 30 Gflop/s on Cell
- Each implementation: vectorized, threaded, cache tuned, ~13 MB of code
- *Code was not written by a human*

Spiral: A Domain Specific Program Generator

SPIRAL www.spiral.net

Transform
user specified

$$\mathrm{DFT}_8$$

Fast algorithm in SPL
many choices

$$(\mathrm{DFT}_2 \otimes I_4)\, T^8_4\, (I_2 \otimes ((\mathrm{DFT}_2 \otimes I_2) \cdot T^4_2\, (I_2 \otimes \mathrm{DFT}_2)\, L^4_2))\, L^8_2$$

∑-SPL:

$$\sum \left(S_j\, \mathrm{DFT}_2\, G_j\right) \sum \left(\sum \left(S_{k,l}\, \mathrm{diag}\left(\mathrm{t}_{k,l}\right) \mathrm{DFT}_2\, G_l\right) \sum \left(S_m\, \mathrm{diag}\left(\mathrm{t}_m\right) \mathrm{DFT}_2\, G_{k,m}\right)\right)$$

C Code:

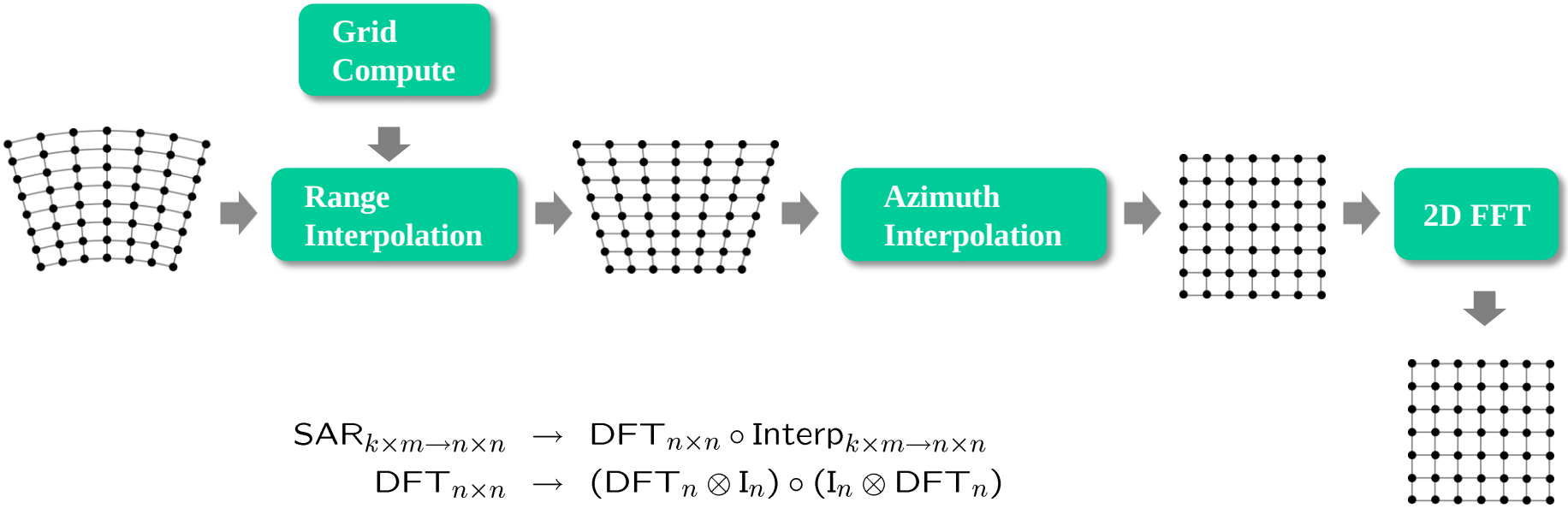
```
void sub(double *y, double *x) {
  double f0, f1, f2, f3, f4, f7, f8, f10, f11;
  f0 = x[0] - x[3];
  f1 = x[0] + x[3];
  f2 = x[1] - x[2];
  f3 = x[1] + x[2];
  f4 = f1 - f3;
  y[0] = f1 + f3;
  y[2] = 0.7071067811865476 * f4;
  f7 = 0.9238795325112867 * f0;
  f8 = 0.3826834323650898 * f2;
  y[1] = f7 + f8;
  f10 = 0.3826834323650898 * f0;
  f11 = (-0.9238795325112867) * f2;
  y[3] = f10 + f11;
}
```

Optimization at all abstraction levels

- **parallelization vectorization**
- **loop optimizations**
- **constant folding scheduling ……**

Iteration of this process to search for the fastest

Spiral Formula Representation of SAR



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

Parallelization through Rewriting

Threading:

$$\underbrace{\mathbf{DFT}_{mn}}_{\mathrm{smp}(p,\mu)} \rightarrow \underbrace{\left((\mathbf{DFT}_m \otimes \mathrm{I}_n)\mathsf{T}^{mn}_n(\mathrm{I}_m \otimes \mathbf{DFT}_n)\,\mathsf{L}^{mn}_m\right)}_{\mathrm{smp}(p,\mu)}$$

$$\cdots$$

$$\rightarrow \underbrace{\left(\mathbf{DFT}_m \otimes \mathrm{I}_n\right)}_{\mathrm{smp}(p,\mu)}\ \underbrace{\mathsf{T}^{mn}_n}_{\mathrm{smp}(p,\mu)}\ \underbrace{\left(\mathrm{I}_m \otimes \mathbf{DFT}_n\right)}_{\mathrm{smp}(p,\mu)}\ \underbrace{\mathsf{L}^{nm}_m}_{\mathrm{smp}(p,\mu)}$$

$$\cdots$$

$$\rightarrow \left((\mathsf{L}^{mp}_m \otimes \mathrm{I}_{n/p\mu}) \otimes_\mu \mathrm{I}_\mu\right)\left(\mathrm{I}_p \otimes_\parallel (\mathbf{DFT}_m \otimes \mathrm{I}_{n/p})\right)\left((\mathsf{L}^{mp}_p \otimes \mathrm{I}_{n/p\mu}) \otimes_\mu \mathrm{I}_\mu\right)$$
$$\left(\bigoplus_{i=0}^{p-1}{}_\parallel\, \mathsf{T}^{mn,i}_n\right)\left(\mathrm{I}_p \otimes_\parallel (\mathrm{I}_{m/p} \otimes \mathbf{DFT}_n)\right)\left(\mathrm{I}_p \otimes_\parallel \mathsf{L}^{mn/p}_{m/p}\right)\left((\mathsf{L}^{pn}_p \otimes \mathrm{I}_{m/p\mu}) \otimes_\mu \mathrm{I}_\mu\right)$$

Vectorization:

$$\underbrace{\overline{\mathbf{DFT}_{mn}}}_{\mathrm{vec}(\nu)} \rightarrow \underbrace{\overline{\left((\mathbf{DFT}_m \otimes \mathrm{I}_n)\mathsf{T}^{mn}_n(\mathrm{I}_m \otimes \mathbf{DFT}_n)\,\mathsf{L}^{mn}_m\right)}}_{\mathrm{vec}(\nu)}$$

$$\cdots$$

$$\rightarrow \underbrace{\overleftarrow{\left(\mathbf{DFT}_m \otimes \mathrm{I}_n\right)}^{\nu}}_{\mathrm{vec}(\nu)}\ \underbrace{\overleftarrow{\left(\mathsf{T}^{mn}_n\right)}^{\nu}}_{\mathrm{vec}(\nu)}\ \underbrace{\overrightarrow{\left(\mathrm{I}_m \otimes \mathbf{DFT}_n\right)\mathsf{L}^{mn}_m}^{\nu}}_{\mathrm{vec}(\nu)}$$

$$\cdots$$

$$\rightarrow (\mathrm{I}_{mn/\nu} \otimes \underbrace{\mathsf{L}^{2\nu}_\nu}_{\mathrm{sse}})(\overline{\mathbf{DFT}_m \otimes \mathrm{I}_{n/\nu}} \vec{\otimes} \mathrm{I}_\nu)\underbrace{\overrightarrow{\left(\mathsf{T}^{mn}_n\right)}^{\nu}}_{\mathrm{sse}}$$
$$\left(\mathrm{I}_{m/\nu} \otimes (\overline{\mathsf{L}^{n}_\nu} \vec{\otimes} \mathrm{I}_\nu)(\mathrm{I}_{n/\nu} \otimes (\mathsf{L}^{2\nu}_\nu \vec{\otimes} \mathrm{I}_\nu)(\mathrm{I}_2 \otimes \underbrace{\mathsf{L}^{\nu^2}_\nu}_{\mathrm{sse}})(\mathsf{L}^{2\nu}_2 \vec{\otimes} \mathrm{I}_\nu))(\overline{\mathbf{DFT}_n} \vec{\otimes} \mathrm{I}_\nu)\right)$$
$$\left((\mathsf{L}^{mn}_m \otimes \mathrm{I}_2) \vec{\otimes} \mathrm{I}_\nu\right)(\mathrm{I}_{mn/\nu} \otimes \underbrace{\mathsf{L}^{2\nu}_2}_{\mathrm{sse}})$$

GPUs:

$$\underbrace{\left(\mathbf{DFT}_{r^k}\right)}_{\mathrm{gpu}(t,c)} \rightarrow \underbrace{\left(\prod_{i=0}^{k-1} \mathsf{L}^{r^k}_r \left(\mathrm{I}_{r^{k-1}} \otimes \mathbf{DFT}_r\right)\left(\mathsf{L}^{r^k}_{r^{k-i-1}}(\mathrm{I}_{r^i} \otimes \mathsf{T}^{r^{k-i}}_{r^{k-i-1}})\underbrace{\mathsf{L}^{r^k}_{r^i+1}}_{\mathrm{vec}(c)}\right)\right)\mathsf{R}^{r^k}_r}_{\mathrm{gpu}(t,c)}$$

$$\cdots$$

$$\rightarrow \left(\prod_{i=0}^{k-1} (\mathsf{L}^{r^n/2}_r \vec{\otimes} \mathrm{I}_2)\left(\mathrm{I}_{r^{n-1}/2} \otimes_\times \underbrace{(\mathbf{DFT}_r \vec{\otimes} \mathrm{I}_2)\,\mathsf{L}^{2r}_r}_{\mathrm{shd}(t,c)}\right)\mathsf{T}_i\right)$$
$$(\mathsf{L}^{r^n/2}_r \vec{\otimes} \mathrm{I}_2)(\mathrm{I}_{r^{n-1}/2} \otimes_\times \underbrace{\mathsf{L}^{2r}_r}_{\mathrm{shd}(t,c)})(\mathsf{R}^{r^{n-1}}_r \vec{\otimes} \mathrm{I}_r)$$

Verilog for FPGAs:

$$\underbrace{\left(\mathbf{DFT}_{r^k}\right)}_{\mathrm{stream}(r^s)} \rightarrow \underbrace{\left[\prod_{i=0}^{k-1} \mathsf{L}^{r^k}_r \left(\mathrm{I}_{r^{k-1}} \otimes \mathbf{DFT}_r\right)\left(\mathsf{L}^{r^k}_{r^{k-i-1}}(\mathrm{I}_{r^i} \otimes \mathsf{T}^{r^{k-i}}_{r^{k-i-1}})\,\mathsf{L}^{r^k}_{r^i+1}\right)\right]\mathsf{R}^{r^k}_r}_{\mathrm{stream}(r^s)}$$

$$\cdots$$

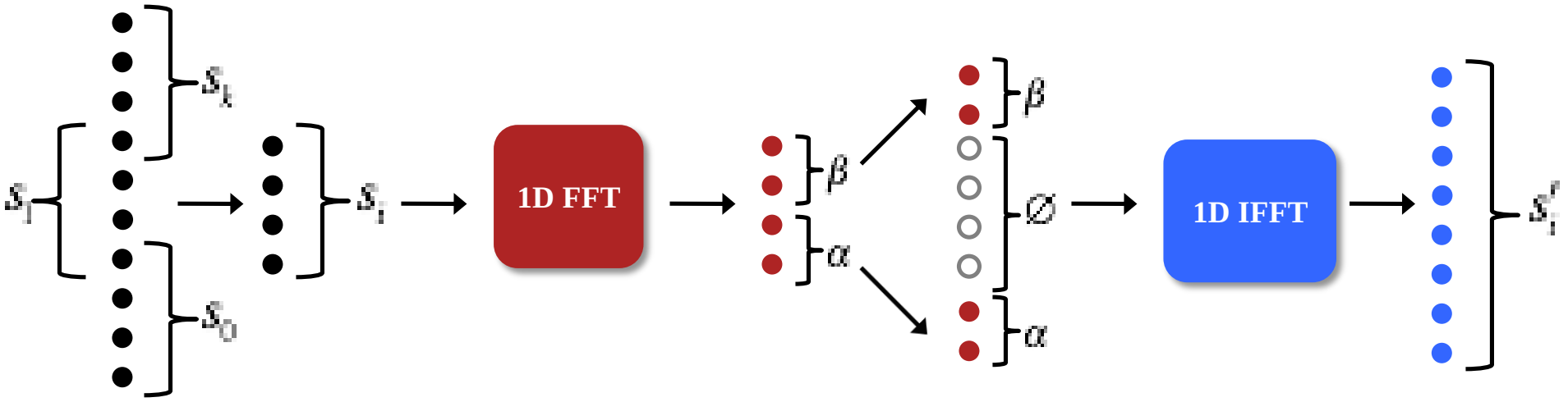
$$\rightarrow \left[\prod_{i=0}^{k-1} \underbrace{\mathsf{L}^{r^k}_r}_{\mathrm{stream}(r^s)} \underbrace{\left(\mathrm{I}_{r^{k-1}} \otimes \mathbf{DFT}_r\right)}_{\mathrm{stream}(r^s)} \underbrace{\left(\mathsf{L}^{r^k}_{r^{k-i-1}}(\mathrm{I}_{r^i} \otimes \mathsf{T}^{r^{k-i}}_{r^{k-i-1}})\,\mathsf{L}^{r^k}_{r^i+1}\right)}_{\mathrm{stream}(r^s)}\right] \underbrace{\mathsf{R}^{r^k}_r}_{\mathrm{stream}(r^s)}$$

$$\cdots$$

$$\rightarrow \left[\prod_{i=0}^{k-1} \underbrace{\mathsf{L}^{r^k}_r}_{\mathrm{stream}(r^s)} \left(\mathrm{I}_{r^{k-s-1}} \otimes_s (\mathrm{I}_{r^{s-1}} \otimes \mathbf{DFT}_r)\right) \underbrace{\mathsf{T}'_i}_{\mathrm{stream}(r^s)}\right] \underbrace{\mathsf{R}^{r^k}_r}_{\mathrm{stream}(r^s)}$$

- **Rigorous, correct by construction**
- **Overcomes compiler limitations**

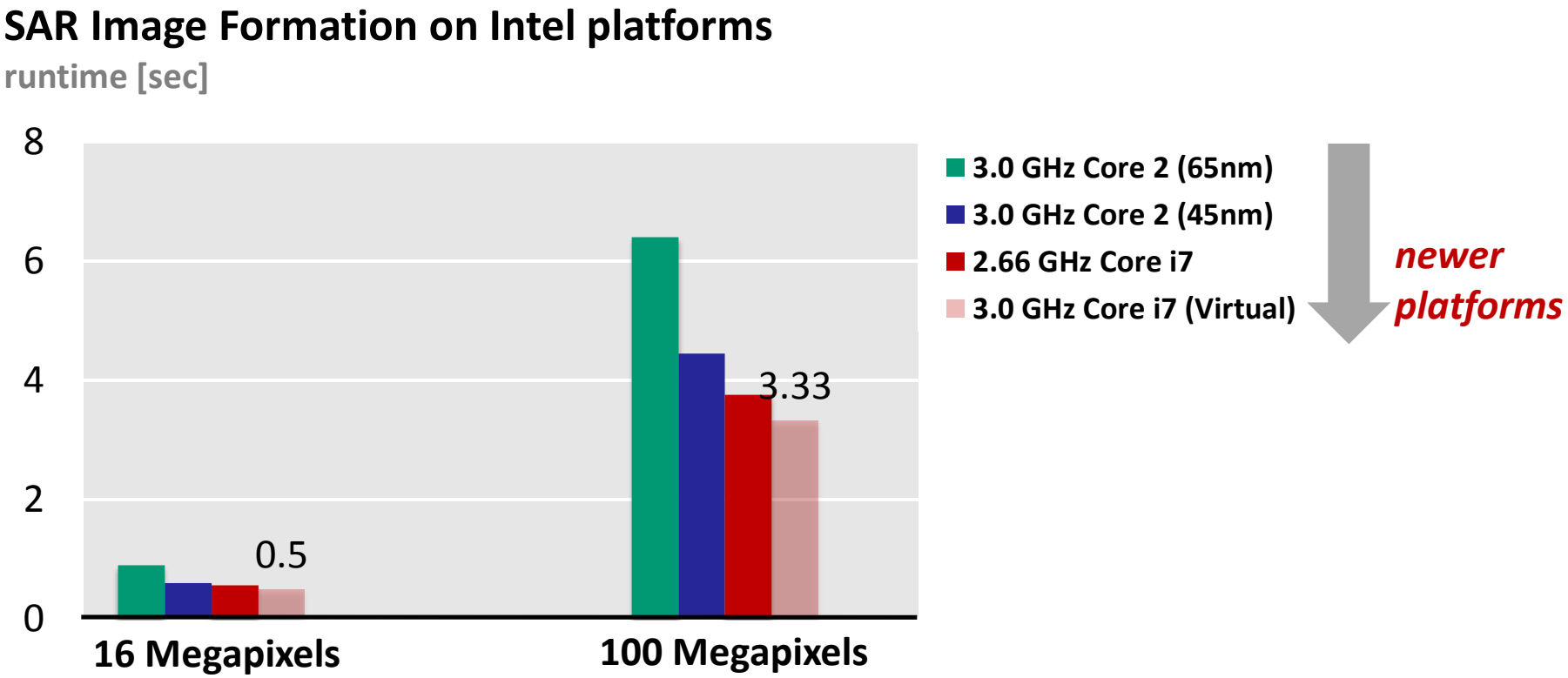
Domain Specific FFT



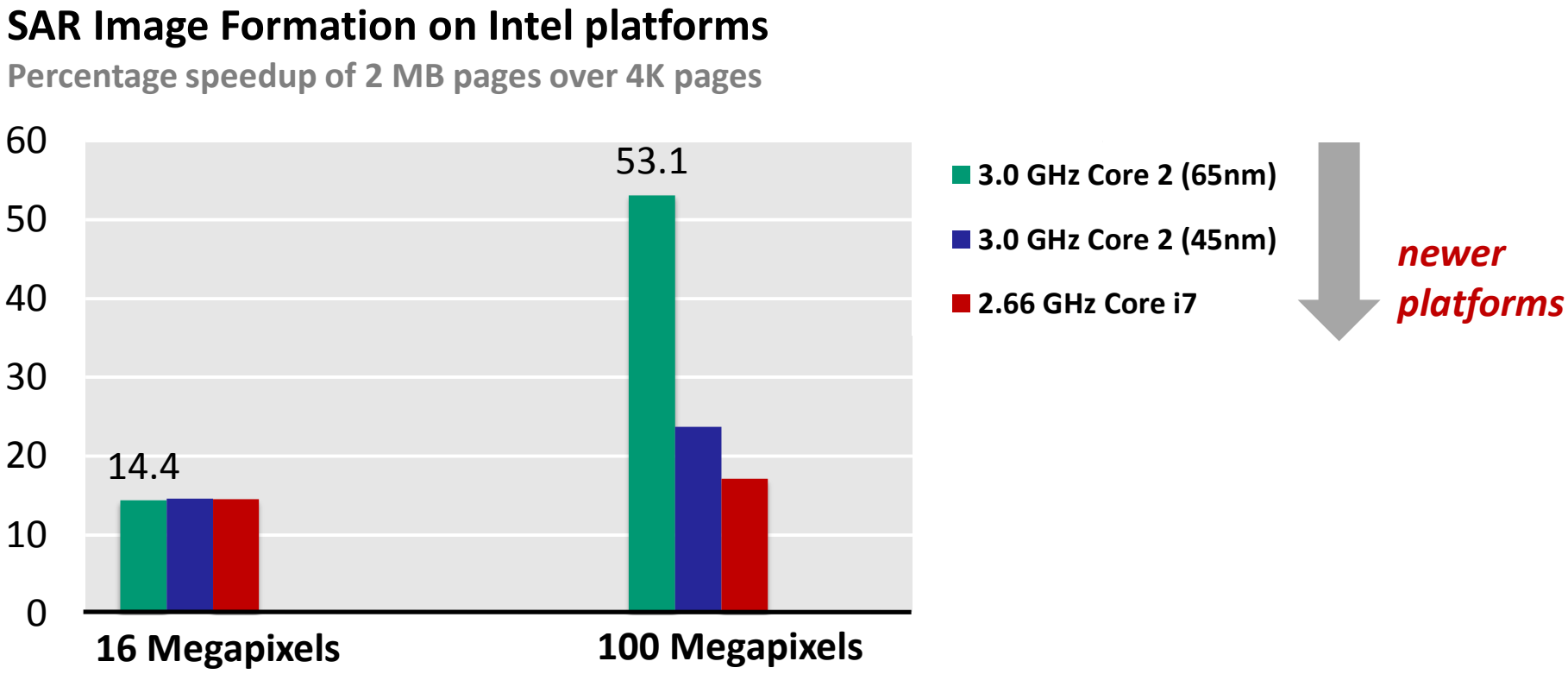
- **Segmented Interpolation**
  k segments of length *r*, with *u*–fold upsamping

Pruned FFT can reduce dominant Interpolation opcount by up to 15%

Performance Results

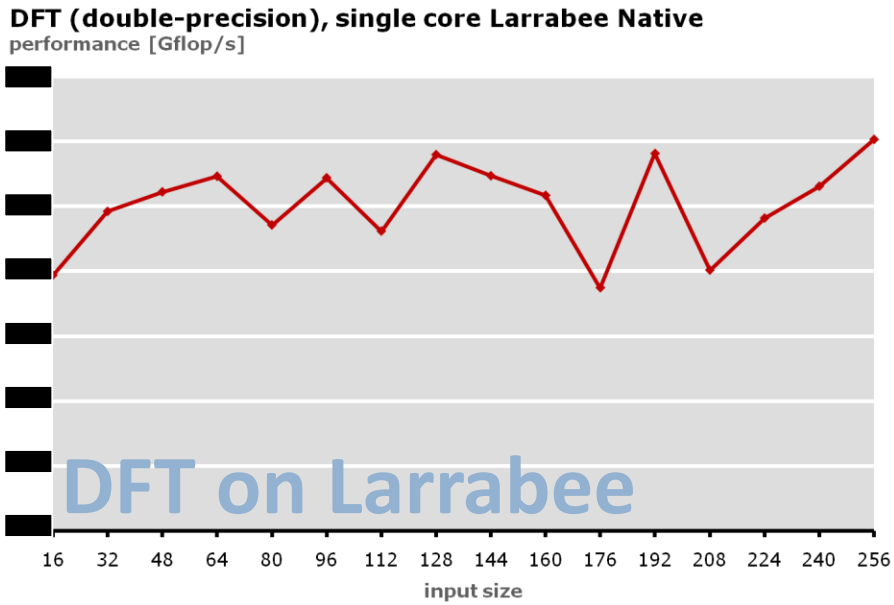


Performance Results



Conclusions and Future Work

- Spiral generated SAR Image Formation performance comparable to hand-tuned code on the Cell
- SAR generation for non-released platforms
  - AVX
  - Larrabee



Not actual data (NDA)