

# Multiscale Dataflow Computing

## The Vertical Perspective



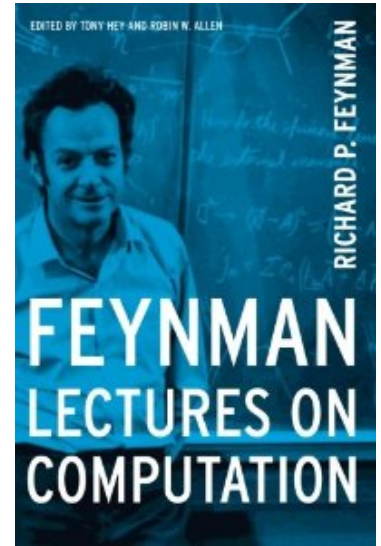
# Richard Feynman on Computation

In theory, a computer system  
can be constructed which uses no energy.

Energy is only needed when **information** is lost.

Reordering of **information** does not require energy  
from a pure physics perspective.

Of course, moving **information** takes Energy...



# The Next Generation

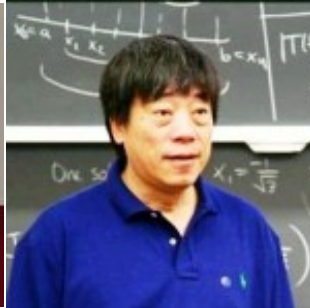
## Systolic Arrays



Martin Morf



S.Y. Kung



H.T. Kung

## Reversible Computing



Tommaso Tofoli  
Tofoli Gate



Ed Fredkin  
Fredkin Gate



Jack Dennis  
Static Dataflow

## Reconfigurable Computing



Arvind  
Dynamic Dataflow



Mike Flynn  
SIMD, SISD  
MIMD, MISD



Dan Slotnick  
ILLIAC IV  
SOLOMON



Norman Margolus  
Cellular Automata  
Machine CAM-8



Jean Vuillemin  
Programmable Active  
Memories



Wayne  
Luk

# Multiscale dataflow computing

Definiton: “Multiscale”

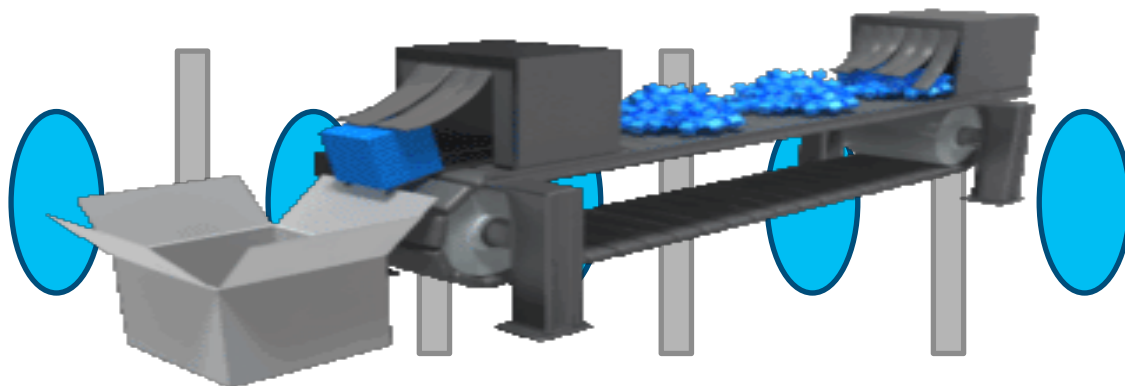
Problems which have important features at multiple scales

Multiple scales of computing	Important features for optimization
complete system level	⇒ balance compute, storage and IO
parallel node level	⇒ maximize utilization of compute and interconnect
microarchitecture level	⇒ minimize data movement
arithmetic level	⇒ tradeoff range, precision and accuracy = discretize in time, space and value
bit level	⇒ encode and add redundancy
transistor level	=> create the illusion of ‘0’ and ‘1’

# Assembly-line Computing

Static ultradeep (>1000 stage) computing pipelines

One result  
per clock cycle



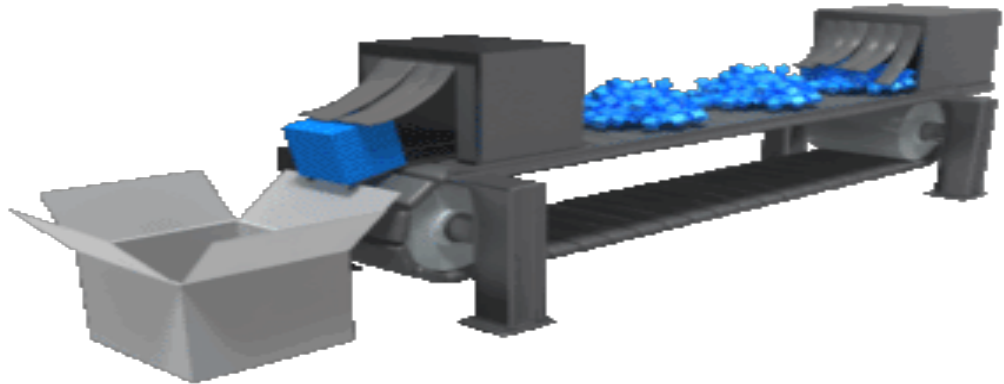
Dynamic (switching) Power:

$$P_{avg} = C_{load} \cdot V_{DD}^2 \cdot f$$

# Assembly-line Computing

Static ultradeep (>1000 stage) computing pipelines

One result  
per clock cycle



Dynamic (switching) Power:

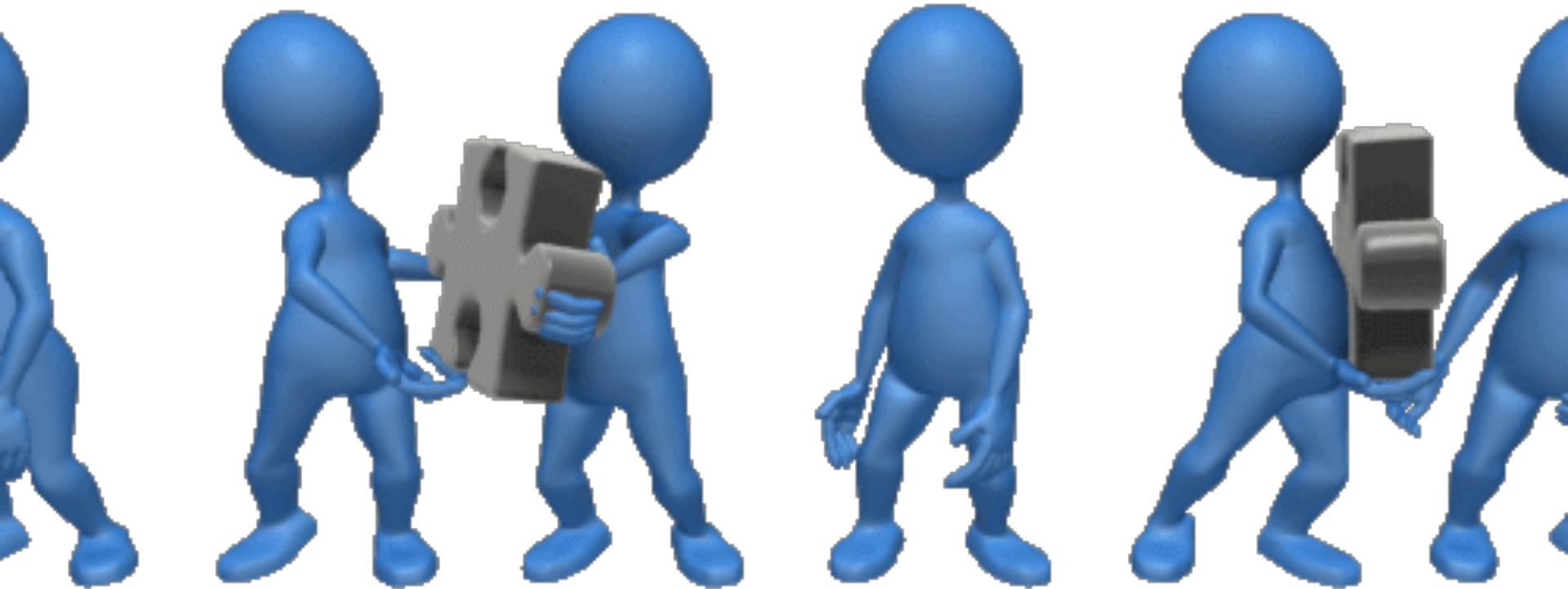
$$P_{avg} = C_{load} \cdot V_{DD}^2 \cdot f$$

Maximum Performance Computing:

Generate the maximal number of results per clock cycle

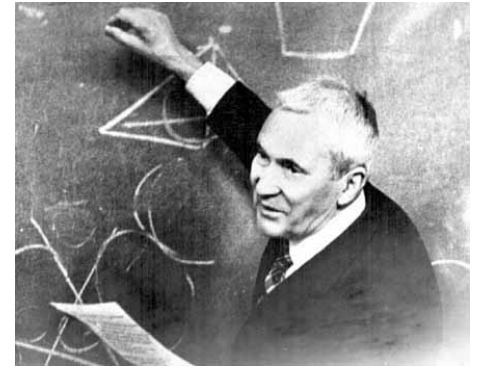
=> Maximum performance per Watt, and per cubic foot.

# What is the most efficient way to move lot's of stuff?



Courtesy of Bob Clapp, Stanford Geophysics

# Kolmogorov Complexity (K)



## **Definition** (Kolmogorov):

“If a description of *string*  $s$ ,  $d(s)$ , is of minimal length, [...] it is called a **minimal description** of  $s$ . Then the length of  $d(s)$ , [...] is the **Kolmogorov complexity** of  $s$ , written  $K(s)$ , where  $K(s) = |d(s)|$ ”

**Of course  $K(s)$  depends heavily on the Language  $L$  used to describe actions in  $K$ .**

**(e.g. Java, Esperanto, an Executable file, etc)**

**for Application-specific computing, really,  
we want to find the optimal language  $L$ , so that...**

# Remembering Fast and Slow

John von Neumann, 1946:

“We are forced to recognize the possibility of constructing a hierarchy of memories, each of which has greater capacity than the preceding, but which is less quickly accessible.”



# Thinking Fast and Slow

Daniel Kahneman

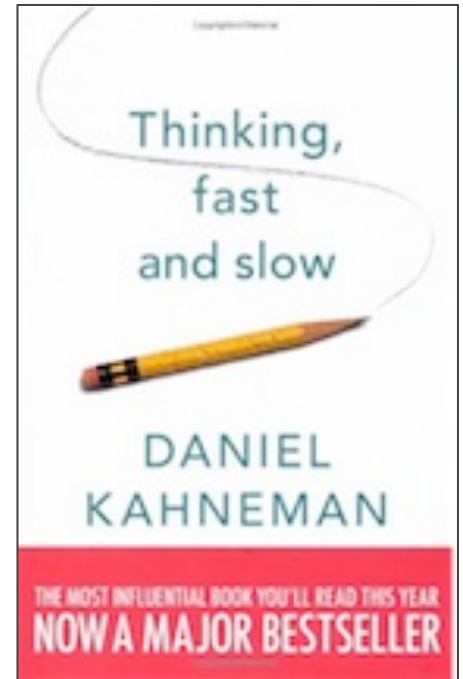
Nobel Prize in Economics, 2002

$17 \times 24 = ?$

Kahneman splits thinking into:

System 1: fast, hard to control ... 400

System 2: slow, easier to control ... 408



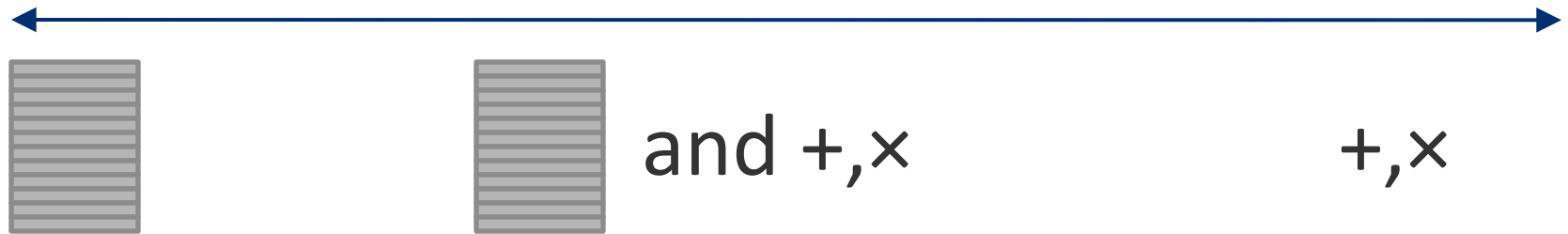
# Putting it all together on the arithmetic level

Computing  $\log(x)$  in the range  $[a,b]$  with  $\text{Error} < x$

*Table*

*Table+Polynomial*

*Polynomial*



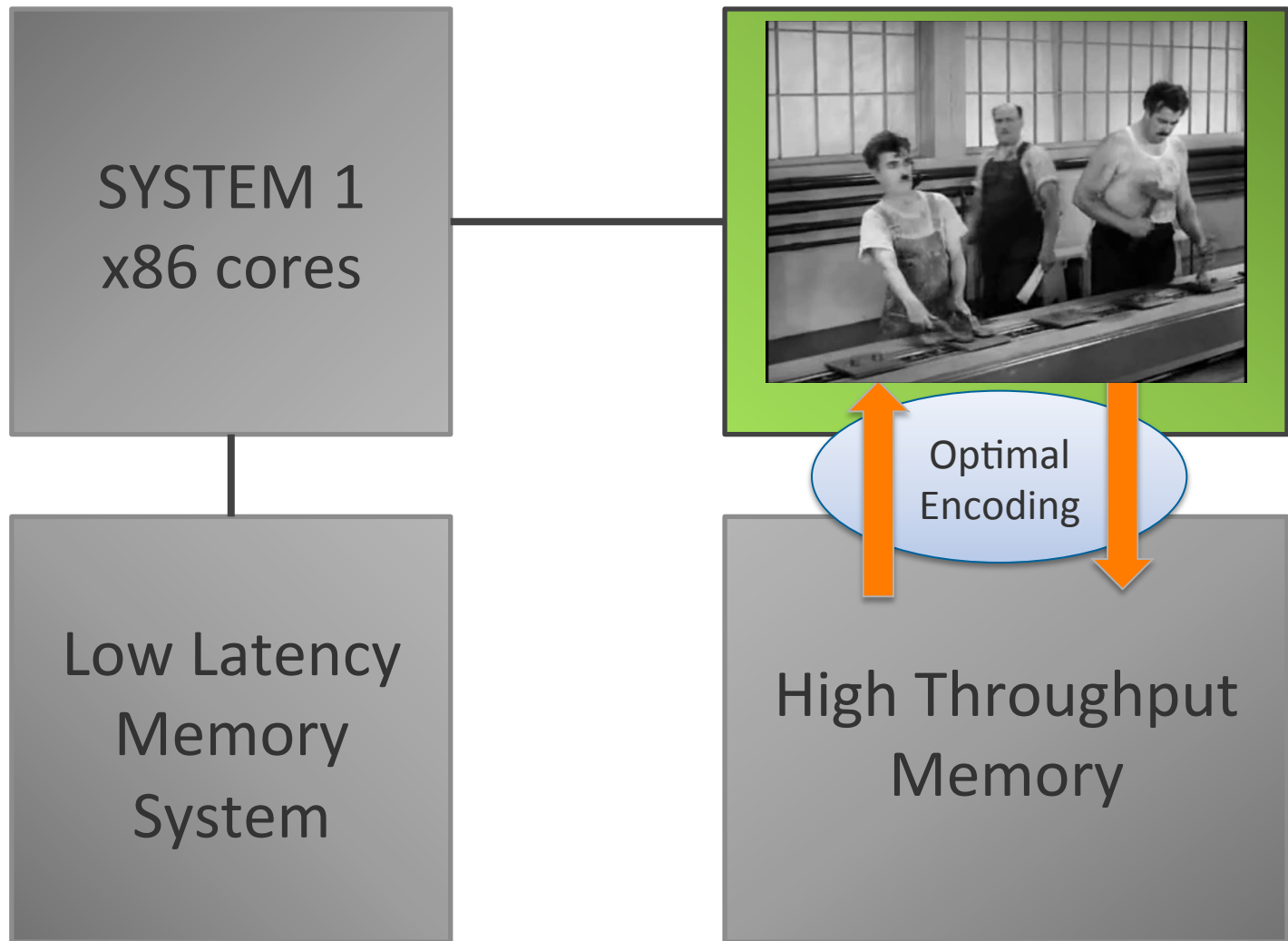
Tradeoff: number of coefficients, number of bits per coefficient, range versus precision of result and maximal versus average error of result

Dong-U Lee, Altaf Abdul Gaffar, Oskar Mencer, Wayne Luk

Optimizing Hardware Function Evaluation

IEEE Transactions on Computers. vol. 54, no. 12, pp. 1520-1531. Dec, 2005.

# optimize microarchitecture with finite resources



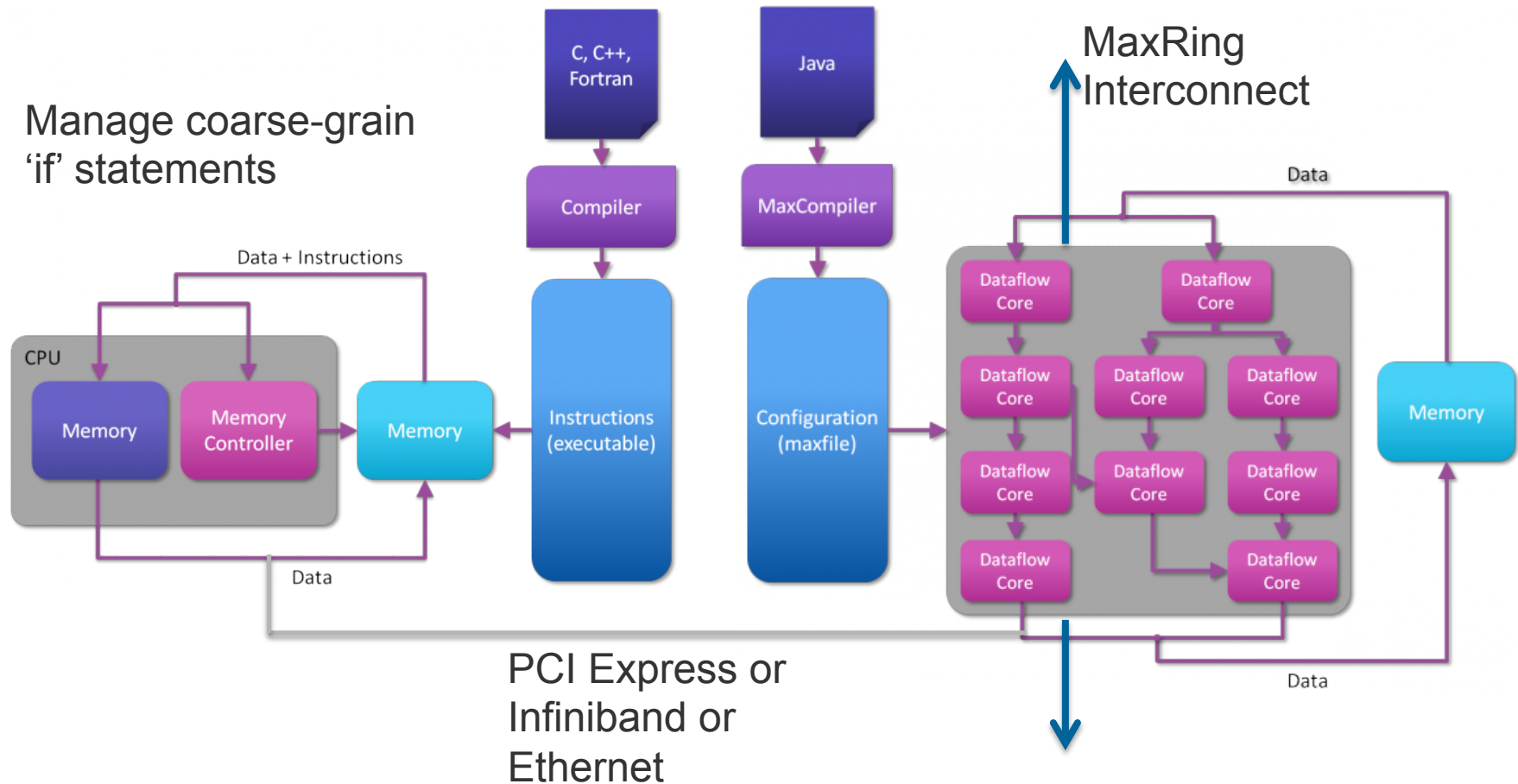
minimize data movement

# A Heterogeneous Computing System

Control (CPU)

Dataflow Engine (DFE)

Manage coarse-grain  
'if' statements

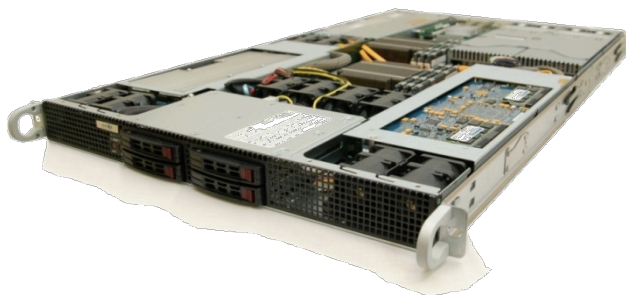


# Maxeler Dataflow Engines (DFEs)



## High Density DFEs

Intel Xeon CPU cores and up to 6 DFEs with 288GB of RAM



## The Dataflow Appliance

Dense compute with 8 DFEs, 384GB of RAM and dynamic allocation of DFEs to CPU servers with zero-copy RDMA access



## The Low Latency Appliance

Intel Xeon CPUs and 1-2 DFEs with direct links to up to six 10Gbit Ethernet connections



## MaxWorkstation

Desktop dataflow development system

## MaxRack

10, 20 or 40 node rack systems integrating compute, networking & storage

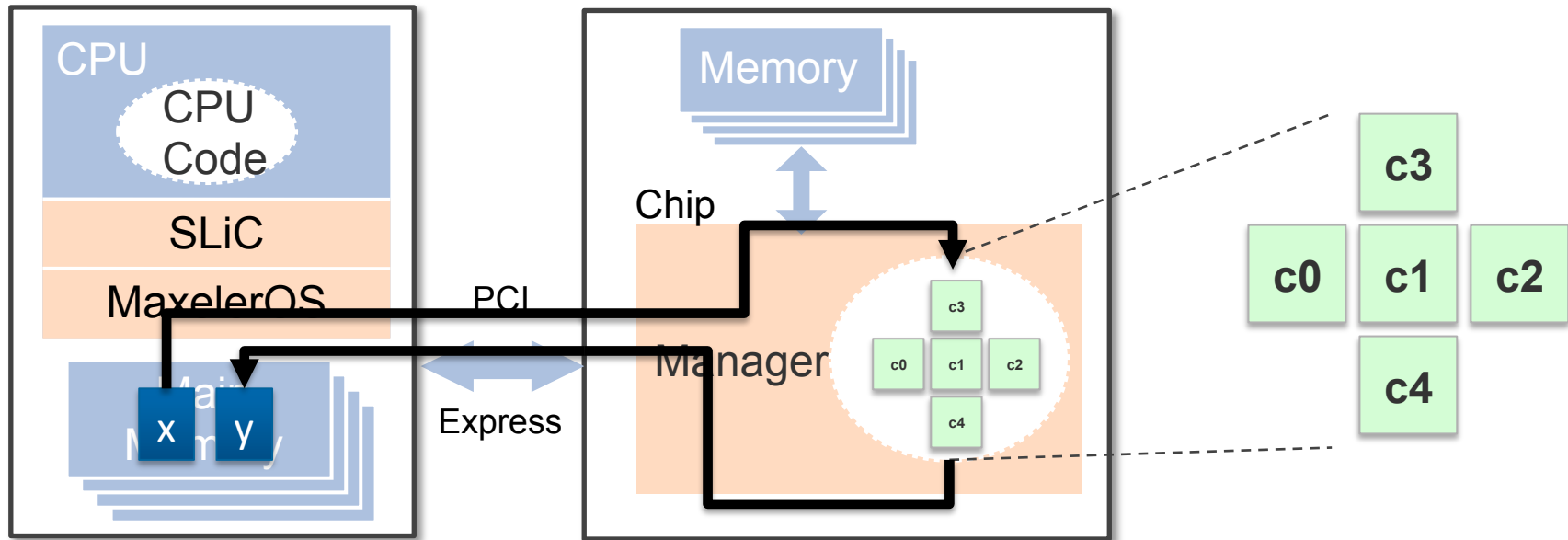
## MaxCloud

Hosted, on-demand, scalable accelerated compute

## Dataflow Engines

48GB DDR3, high-speed connectivity and dense configurable logic

# 2D convolution with MaxCompiler



CPU Code (.c)

```
#include "MaxSLiCInterface.h"
#include "Calc.max"
int *x, *y;

for (int i = L; i < DATA_SIZE - L; i++)
    y[i] = c0*x[i-1] + c1*x[i] +
    Calc(x, y, DATA_SIZE)
    c3 * x[i-L] + c4 * x[i+L]
```

Manager (.java)

```
Manager m = new
Manager("Calc");
Kernel k =
    new MyKernel();
m.setKernel(k);
m.setIO(
    link("x", PCIE),
    link("y", PCIE));
m.addMode(modeDefault());
m.build();
```

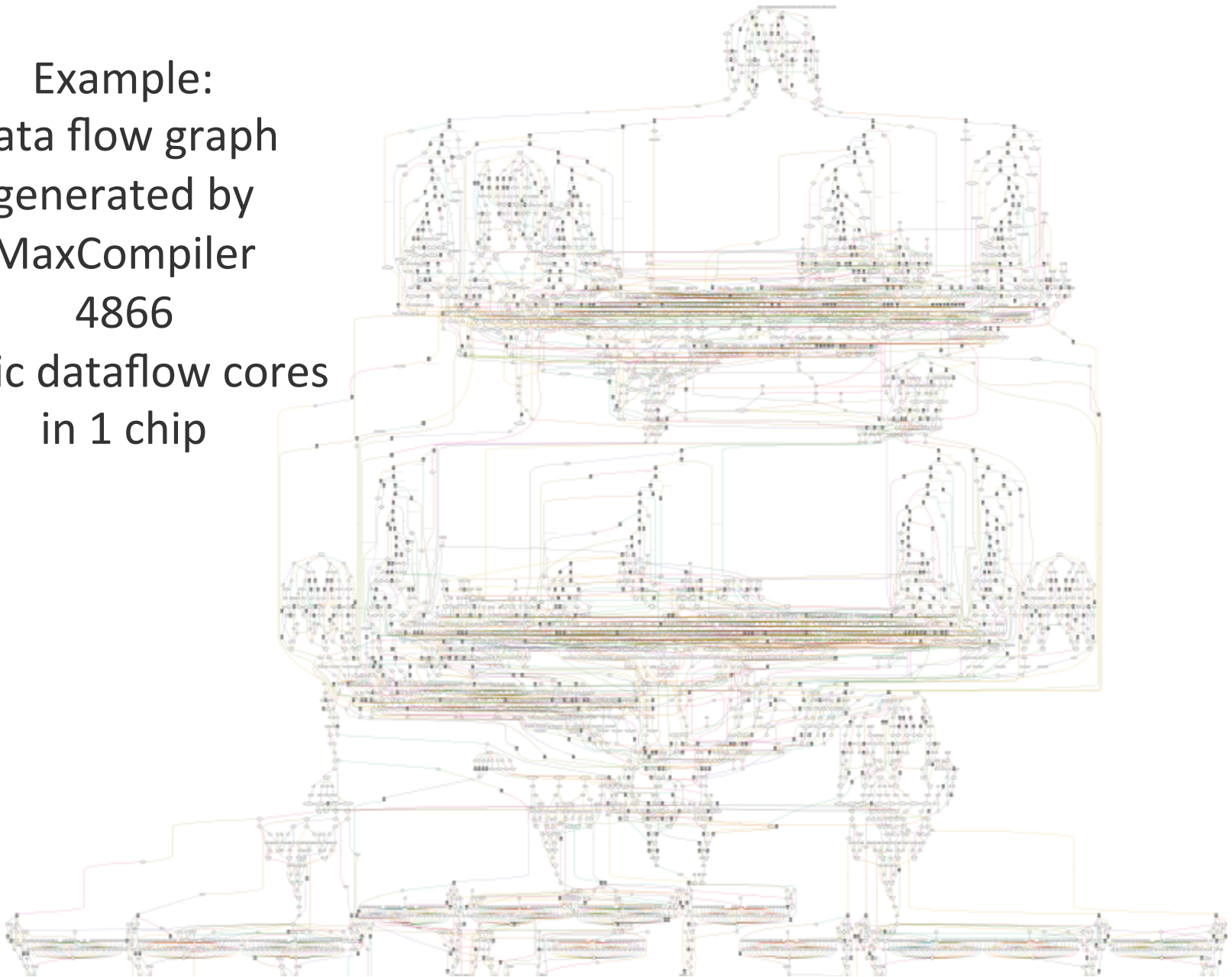
MyKernel (.java)

```
DFEvar x = io.input("x", hwInt(32));

DFEvar result = c0*FIFO(x,1) +
    c1*x +
    c2*FIFO(x,-1) +
    c3*FIFO(x,L) +
    c4*FIFO(x,-L);

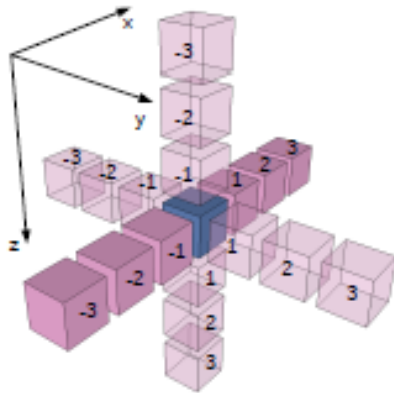
io.output("y", result, hwInt(32));
```

Example:  
data flow graph  
generated by  
MaxCompiler  
4866  
static dataflow cores  
in 1 chip

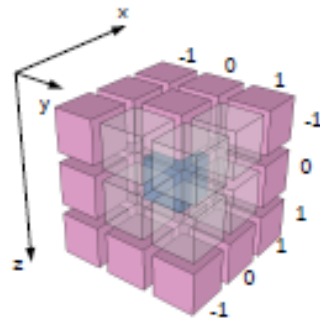


# Star versus Cube Stencil

## More Computation in Less Time?



(a) 7-point star stencil



(b) 3x3x3 cube stencil

Figure 1: 2 Alternative stencil choices.

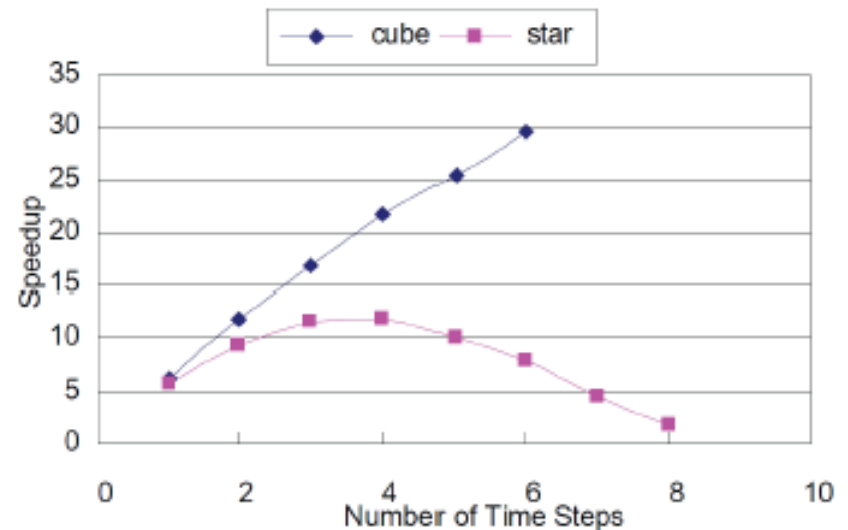
19 MADDs

27 MADDs

Local Buffer = 6 slices

Local Buffer = 3 slices

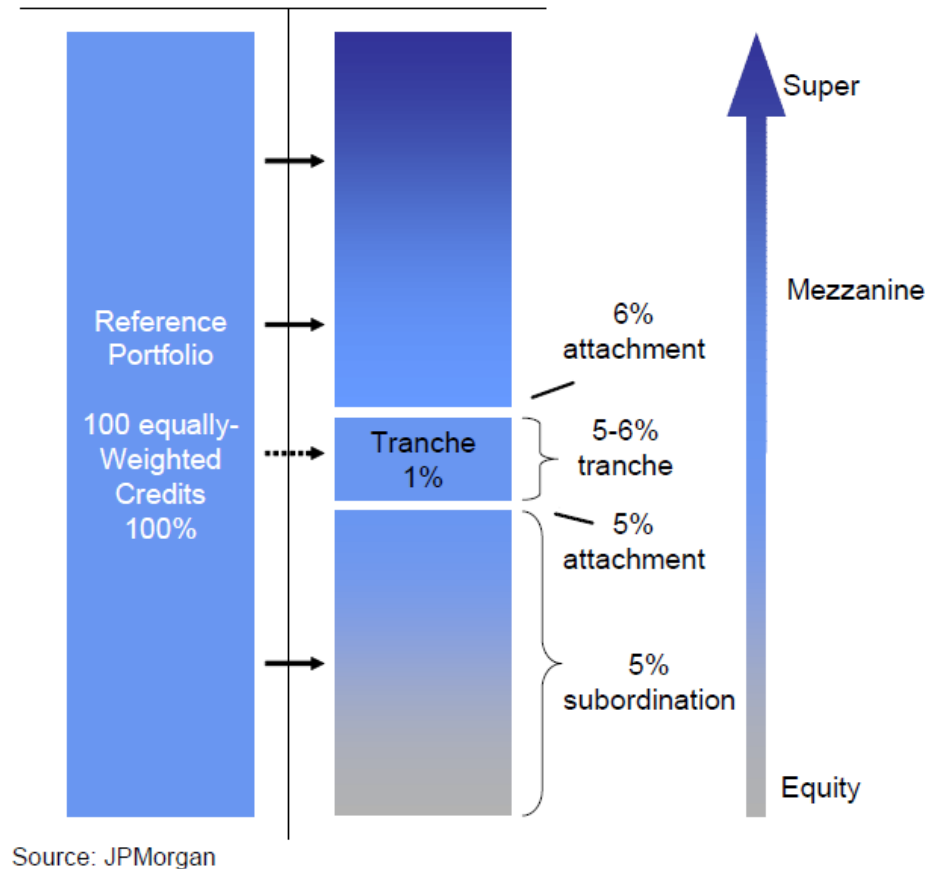
Local temporal parallelism  
⇒ cascade in space  
⇒ twice as many timesteps



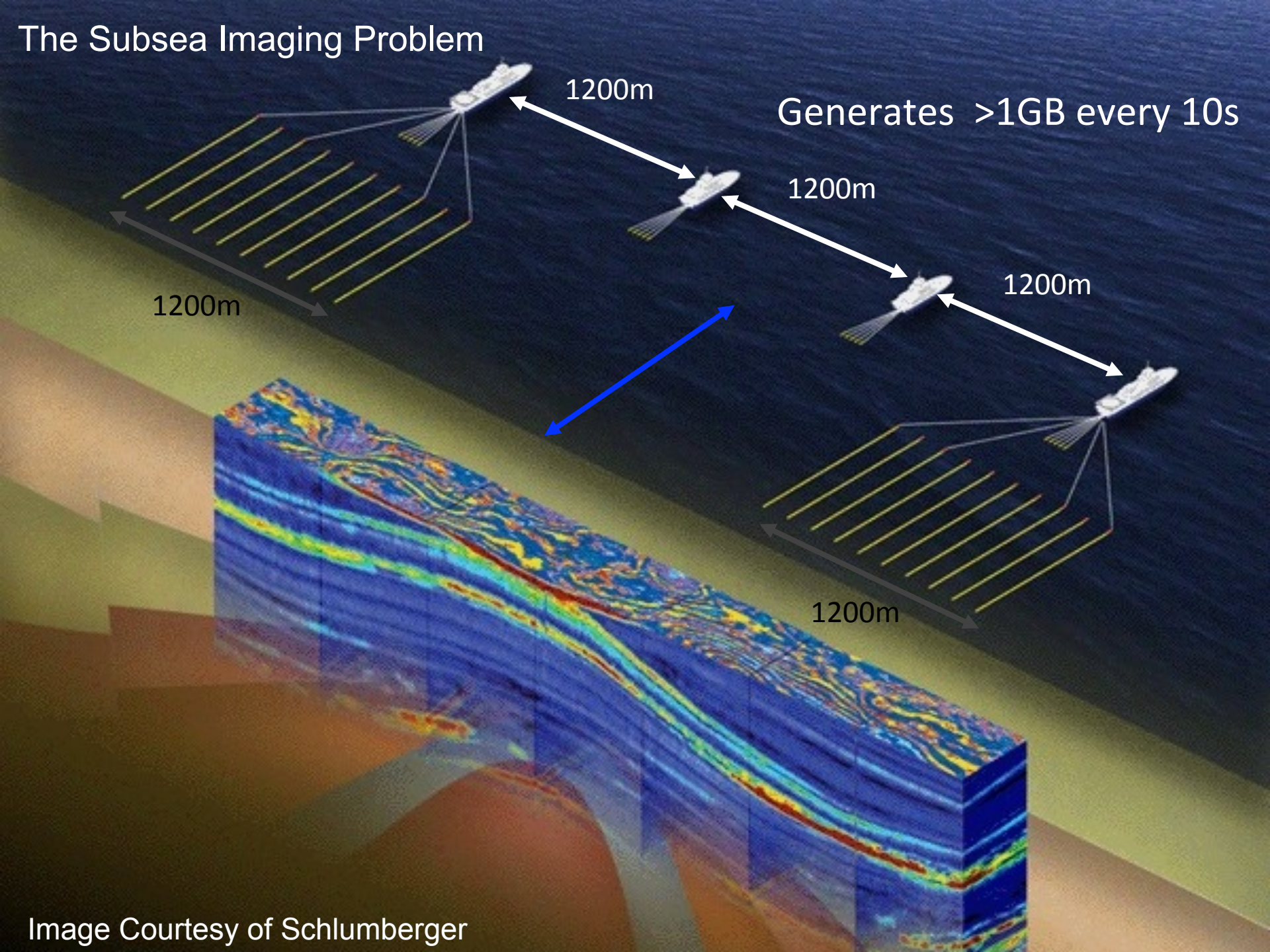
# JP Morgan Credit Derivatives Pricing

American Finance Technology Award for Most Cutting Edge Technology Winner, Dec 2011

- Moving overnight run to real-time intra-day
- **Compute the probability of default and associated price and risk.**
- Power consumption per node drops from 250W to 235W per node, with 22x less nodes.

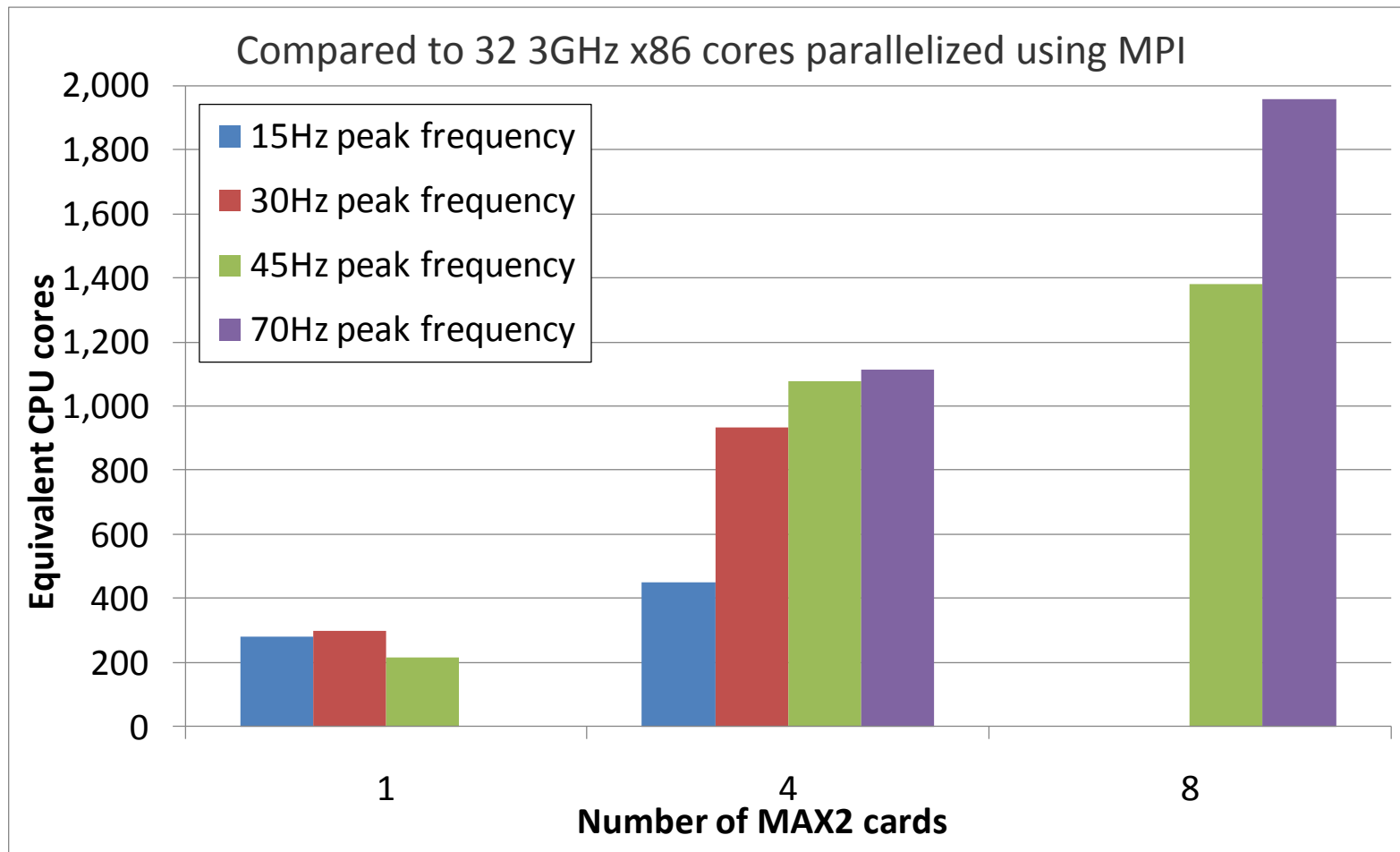


# The Subsea Imaging Problem



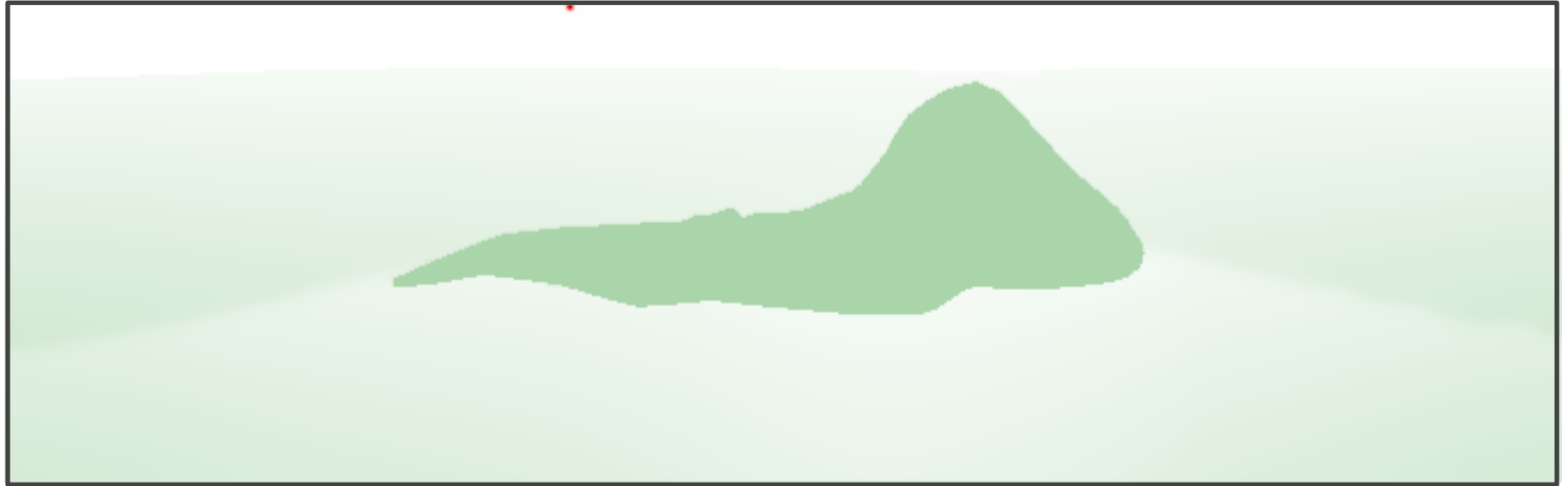
# 3D Finite Difference

2 MaxNodes equiv. to 1,900 Intel CPU cores



FD modeling beyond 70Hz with FPGA acceleration, D. Oriato, O. Pell, C. Andreoletti and N. Bienati. Maxeler Technologies, Eni E&P Division, SEG 2010 HPC Workshop, Denver, Oct 2010.

# Running with SEG Salt Model



- ✓ Running on MPC-C servers
  - 8 parallel compute pipelines per chip
  - 150MHz => low power consumption!
  - 30x advantage over same size CPU-only implementation

Beyond Traditional Microprocessors for Geoscience High-Performance Computing Applications  
O. Lindtjorn, R. G. Clapp, O. Pell, O. Mencer, M. J. Flynn and H. Fu, Stanford, Schlumberger,  
Maxeler, Tsinghua University, IEEE Micro, vol. 31, no. 2, March/April 2011.

# Smith Waterman Algorithm

GCA AGA GAT AAT TGT...

Ala Arg Asp Asn Cys ...

1 2 3 4 5

uniprot\_sprot.fasta

Number of sequences : 532224  
Number of residues : 188726448

**Scoring matrix :**

BLOSUM62

**Open Gap Penalty**

**Stop** **Compute**

Stopping computation - please wait...

**Query :**  
UniRef50\_F2T2I7 Histone-lysine N-methyltransferase n=8 Tax=E (1280)

**Best scores :**

Accession	Protein Name	Length	SW
sp Q1DR06 SET1_COCIM	Histone-lysine N-methyltransferase, H3	(1271)	4077
sp Q2UMH3 SET1_ASPOR	Histone-lysine N-methyltransferase, H3	(1229)	3849
sp Q4WNH8 SET1_ASPFU	Histone-lysine N-methyltransferase, H3	(1241)	3818
sp Q5B0Y5 SET1_EMENI	Histone-lysine N-methyltransferase, H3	(1220)	3683
sp Q8X0S9 SET1_NEUCR	Histone-lysine N-methyltransferase, H3	(1313)	2299
sp Q4I5R3 SET1_GIBZE	Histone-lysine N-methyltransferase, H3	(1252)	2150
sp Q2GWF3 SET1_CHAGB	Histone-lysine N-methyltransferase, H3	(1076)	2089
sp Q6BKL7 SET1_DEBHA	Histone-lysine N-methyltransferase, H3	(1088)	985
sp Q6CEK8 SET1_YARLI	Histone-lysine N-methyltransferase, H3	(1170)	938
sp Q5ABG1 SET1_CANAL	Histone-lysine N-methyltransferase, H3	(1040)	893

**Best alignment :**

```

MSRASAGFADFFPTAPSVLQKKRSSKAAQDRPKGKLKHDDDPQSSNPAPTAAATAAVTVTGVGVPGAEEGGASDNNNTNSDV
MSRAPAGFADFFPTAPSVLQKKRS-KAAQDR-HAA--NT--PKAADPLPNLGLSS-T-----PDIK-GGVG---TSAD-

HNNINSSNNNNKNNSSHTNINSNTQFDESAGAVARGDVNITPGDANGVGSSSSTSTGSS-VFSASILPQPLTTSNGITH
-NPVRAVGE-R--SAE-T-----T-L--A--L--GDTN---G-AT---SSSSLSTGSSGFFSASA-P-PGVAKPNGISS

PHALTPLTNTDSSSPCKIASPSGQKS-IA-ATGEIVPTSRFVDDIK-ATITPLQTPPTPRIQARPAQNAPKGYKLTYPD
C-ALTPLTNTDSSSPCKIESPLGSKSGSTDAAPQLAPTCEAHGGPEPVTITPLHTPPTPRVQARPANSEVKGHKITYPD

LERK-PLTKEKRRKPQYEVFDTTED-EAPPADPRIAIAANYTRGAGCKQKTKYRPAFYILRPWPYDPA TSVGPGPPTQIVV
LDRKFP-SKARRRKPOYETEGVDDEKDPPPCDPRMAIAANYTRGAA CKQKTKYRPTFYILRPWAYDPTTSVGPGPPTQIVV

TGYPDLTPLAIPISALFSSFQDIAETKNRTDPNTGRFLGVCSIRYKDSRMFRGGGPLLAAQAARRAYLECKKEQRIQVRI
TGFDP LPIAAISALFSSFQDIEINNRTDPM TGRFLGVCSIKYKDSRAFRGGISLSASQAVRRAYLECKKEQRIQTRRI

QVSLDRDGVVSDRLVARIIGSOR-Q-----DEP-PPLVME-E-KM-----KSE-EQ---DNLPPPTAPKGPS-RK---PNM
RVELDRNGVVSGRMVAKLITAKAEFFPSLEESRKESVGDNDNRLPIGDGAKKDNEQSKDNLPPSTAPKGPSGRSSLHPSL

LIPEGPRATMMKPPAPSLIEETPILDQIKRDPYIFIAHCYVPVLSTTIPHLELRLKLFNWKSVRCDKTGYYIIFDNSRRG
LAPDGPRAL-VLKSPVPSRIETPILQDIKRDYIFIAHCYVPVLSTTVPHLELRLKLYDWKAVRCDKTGYYIIFENSRRG

```

**Performance :** 812.0759 GCUPS



# Compute Appliance for Medical Use

- Add dense compute directly to BioMedical device
- Remove the need for a large HPC datacenter



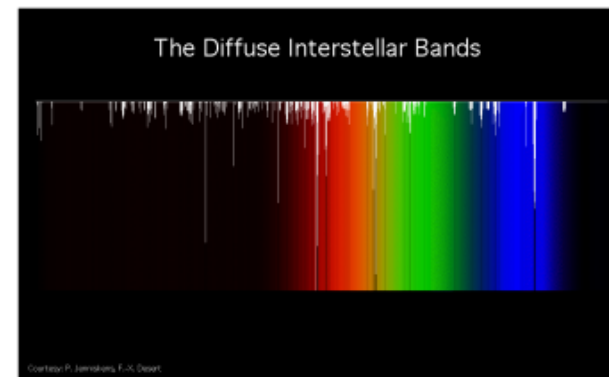
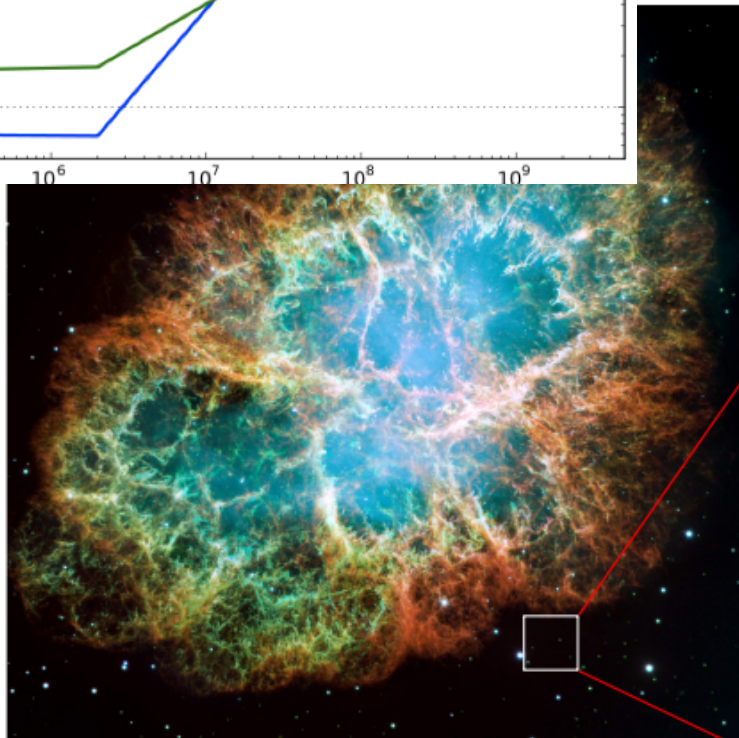
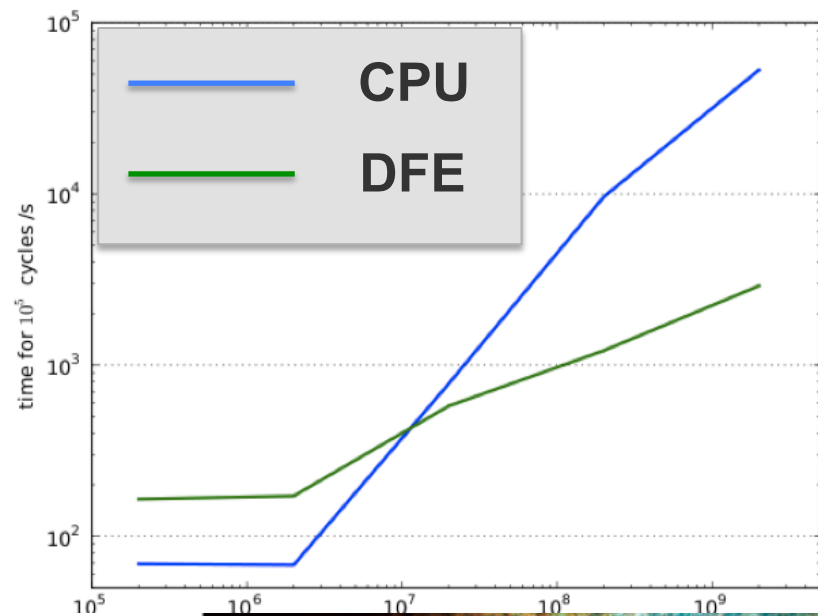
A compute appliance eliminates the need for unscalable datacenter solutions requiring expensive high-speed networking, large HPC infrastructure, large HPC infrastructure operations teams, and immense power consumption.

- A dataflow compute appliance makes solutions more portable, scalable, and moves computations from batch jobs to real time.



## MAX-UP: Astro Chemistry

coronene:  $C_{24}H_{12}$



<http://www.nasa.gov/>

P. Jenniskens and F.-X. Desert, *Astronomy and Astrophysics Supp. Ser.* **106**, 39–78 (1994)

# MAX-UP: Acceleration of an algorithm based on the Gross Pitaevskii equation.



University of  
Belgrade

Execution time for two iterations:

CPU execution time: 50s

Dataflow execution time: 4.9s

Achieved speedup: 10.2x

Programmer:

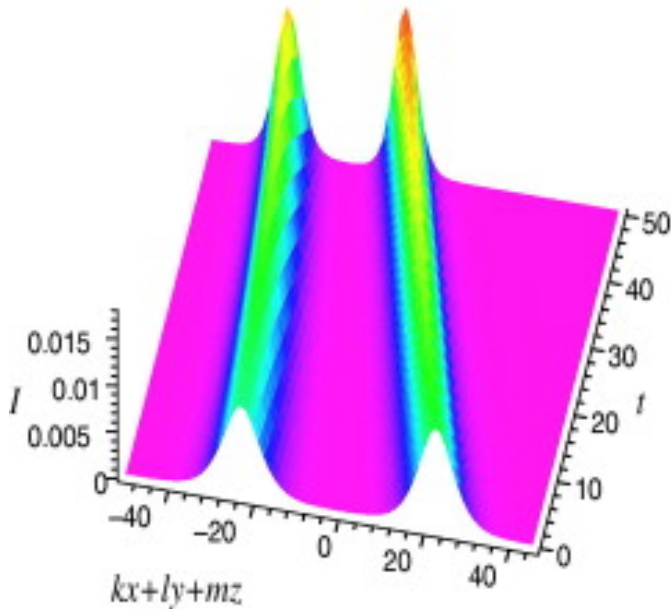
Sasha Stojanovic

[stojsasa@etf.bg.ac.rs](mailto:stojsasa@etf.bg.ac.rs)

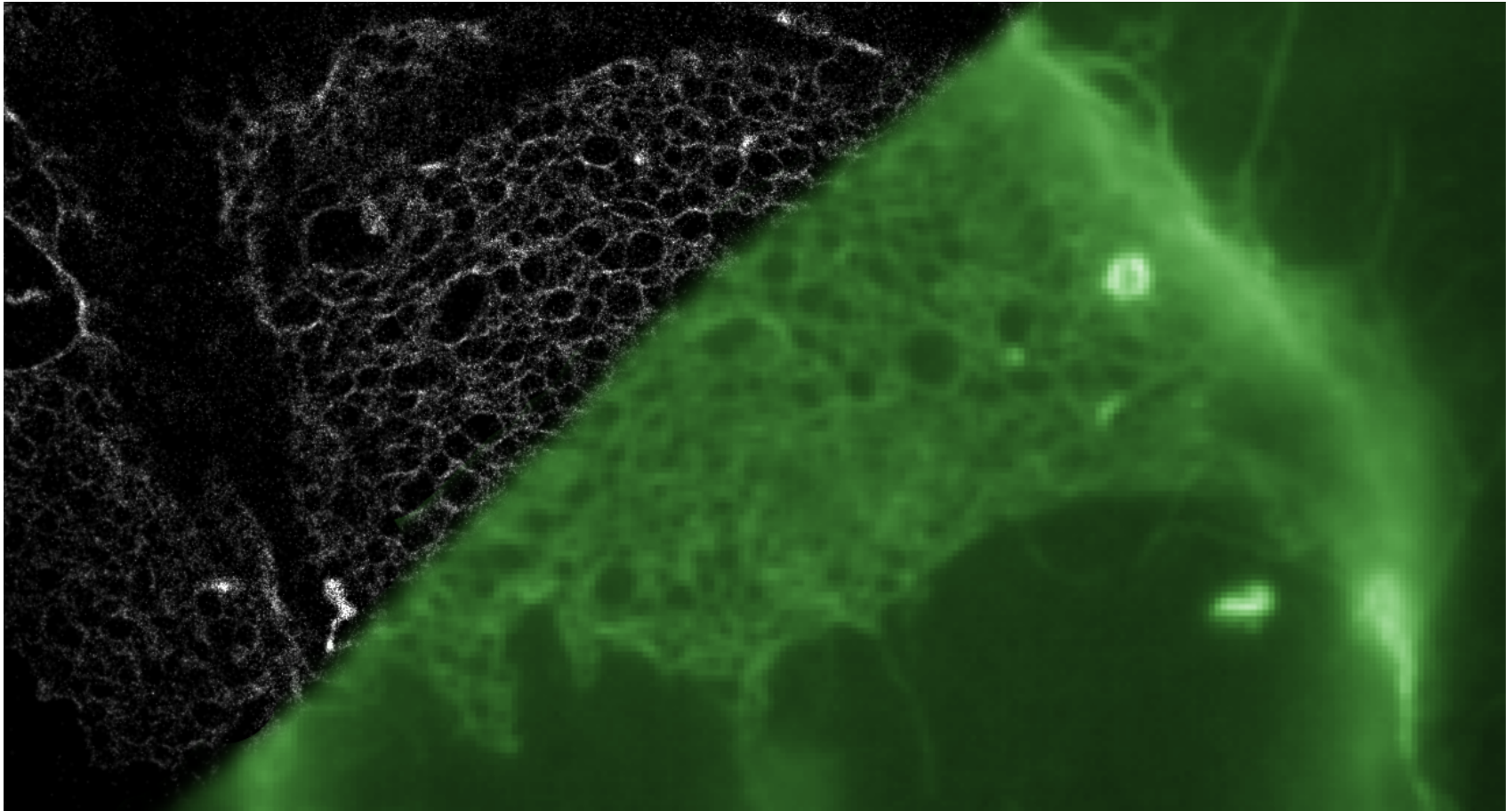
Advisor:

Veljko Milutinovic

[vm@etf.bg.ac.rs](mailto:vm@etf.bg.ac.rs)



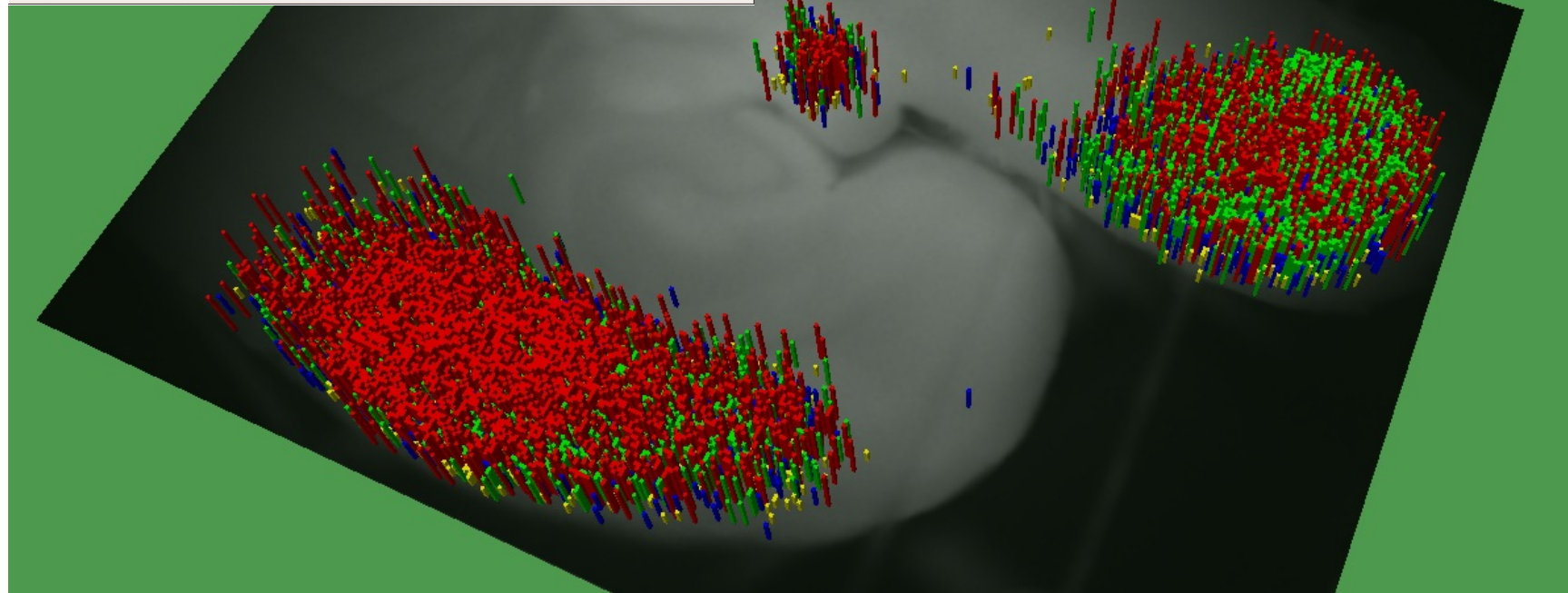
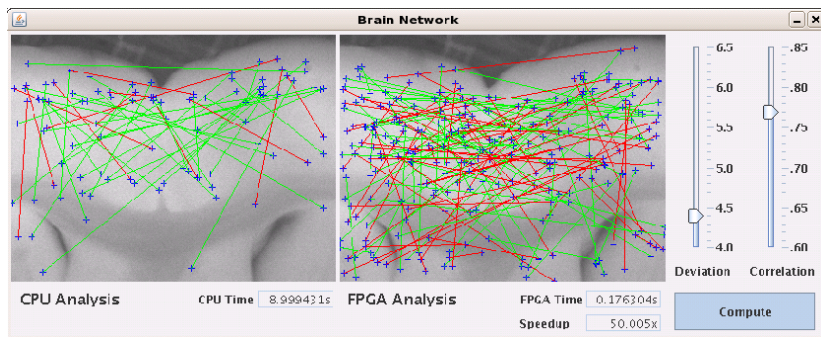
Note: Comparison is made between a single core CPU without using any of extensions available in today's modern processors and a single dataflow implementation on the oldest and smallest MAX 2 Maxeler card. Using modern CPUs equipped with several cores, each one with extensions for parallel processing of data, and modern Maxeler card with higher capacity, bandwidth to memory and host, and working frequency, will result in decreased execution times in both cases, but it is expected that the speedup will not change significantly.



Localization microscopy enhances the resolution of fluorescence light microscopy (shown in green) by about an order of magnitude. Single fluorescent molecules act as switchable markers. Their detected signals can be fitted with a two-dimensional Gaussian distribution and thus located with sub-pixel resolution. Using MaxCompiler we achieved an **acceleration of 225** in signal detection and fitting.



KIRCHHOFF-  
INSTITUTE  
FOR PHYSICS



**UIC** University of Illinois  
at Chicago

### Brain network analysis

Dynamic community analysis on corpus callosum to identify functional regions. By accelerating the linear correlation calculation, Maxeler dataflow engines can build brain networks on-the-fly while the lab experiment is running.

# Maxeler University Program Members

