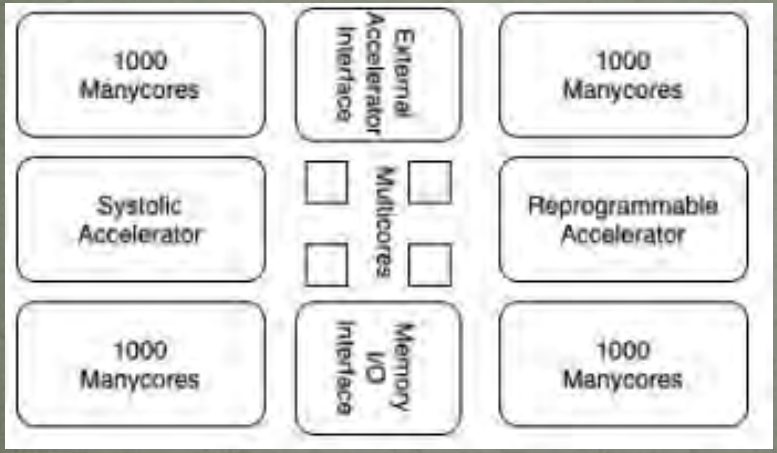
Ultimate DataFlow SuperComputing for BigData DeepAnalytics

V. Milutinovic, University of Indiana, Bloomington, USA

Acknowledgements: O. Mencer, Imperial College, London, GBR, **M.J. Flynn**, Stanford University, Palo Alto, USA,
M. Kotlar, School of Electrical Engineering, University of Belgrade, SRB



Chip Hardware Type	Estimated Transistor Count
One Manycore with Memory	3.29 million
4000 Manycores with Memory	11 800 million [17]
One Multicore with Memory	1 billion [18]
4 Multicore with Memory	4 billion
One Systolic Array	<1 billion [19]
One Reprogrammable Ultimate Dataflow	<69 billion [20]
Interface to I/O with external Memory	<100 million
Interface to External Accelerators	<100 million
TOTAL	<100 billion

Table 1. Basically, current efforts include about 30 billion transistors on a chip, and this article advocates that, for future 100 billion transistor chips, the most effective resources to include are those based on the dataflow principle. For some important applications, such resources bring significant speedups, that would fully justify the incorporation of additional 70 billion transistors. The speedups could be, in reality, from about 10x to about 100x, and the explanations follow in the rest of this article.

Major Sources of Inspiration

A. Richard Feynman:
   Impact of logic/arithmetic and memory/IO
   Compiler-generated execution graph

B. Ilya Prigogine:
   Impact of energy, entropy, order, and optimization
   Compiler-generated data separation

C. Daniel Kahneman:
   Impact of approximate computing on precision
   Compiler-controlled approx computing

D. Tim Hunt:
   Impact of system latency on precision
   Compiler-controlled system latency

The Major Axiom of Optimal Computing

A. Whenever the Technology changes,
the Fundamental Paradigm of Computer Architecture
has to change, too.

aSoG (not: FPGA)

B. If several paradigms are available,
the most suitable paradigm for adoption
is the one most effective for modern Applications.

BrontoData (not: ExaBigData)

Is the von Neumann Paradigm still the most effective one?

A. MultiCores?

B. ManyCores?

The Holy Trinity of Generalized Computing

Applications

Architecture

-Size
-Power
-Speedup
-Precision

Technology

The von Neumann Paradigm (1940s)

$$\lim_{i \to \infty} \left(\frac{TALU(i)}{TCOMM(i)} \right) \to \infty$$

Optimal Solution: Finite Automata

The Nobel Laureate Richard Feynman Observations

$$\lim_{i \to \infty} \left(\frac{TALU(i)}{TCOMM(i)} \right) \to 0 (t \to \infty)$$

Where is the technology now?

A. Closer to 1940s?
B. Closer to t->∞?

State of the Art in Technology Today

~~The Power Challenge~~

The Data Movement Challenge

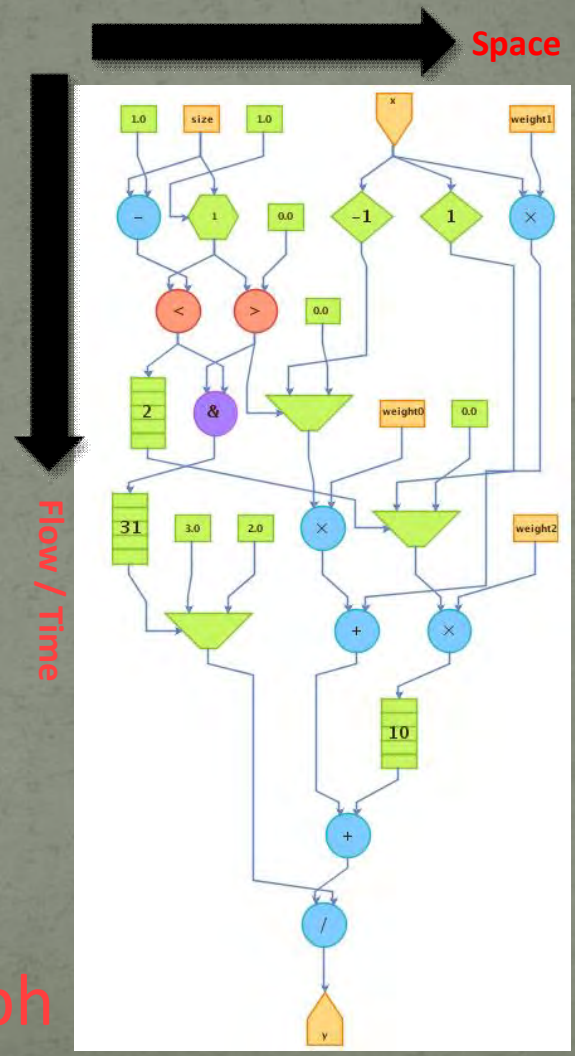
	2015	2020
Double precision FLOP	100pj	10pj

- Moving data off-chip will use **200x more energy than computing**!
- Moving data in 1940s was using **1/60x** ...
- Conclusion: We are getting close to the Feynman Asymptote!
- Important: Power and speed could be traded!

The Maxeler Technology Vision: MultiScale DataFlow

- Thinking in space rather than in time
- Difficult change in mindset to overcome
- Transformation of data through flow over time
- Instructions are parallelized across the available space

Optimal Solution: Execution Graph



Comparing the Two Approaches

- The Von-Neumann paradigm resembles an old wall clock

- The Feynman paradigm resembles lightning! Why?

Programming the Two Paradigms

von Neumann:

The Program Moves Data

Feynman:

The Program Configures Hardware

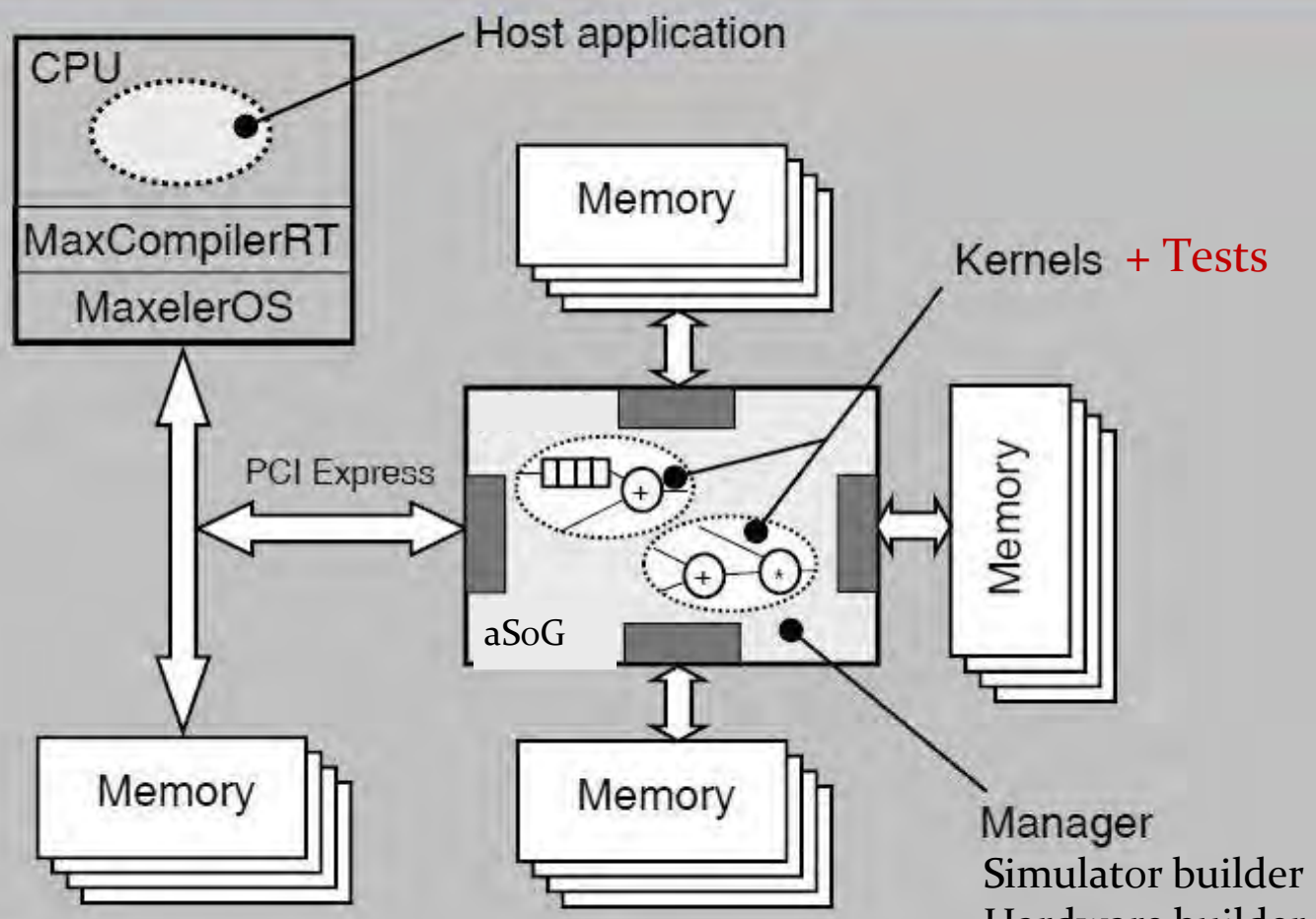
What moves data?

External sources till input.

Voltage difference through aSoG!

Voltage difference moves the important stuff!

The Maxeler Generic Architecture Application



Important: Supporting any CL and any OS!

2n+3

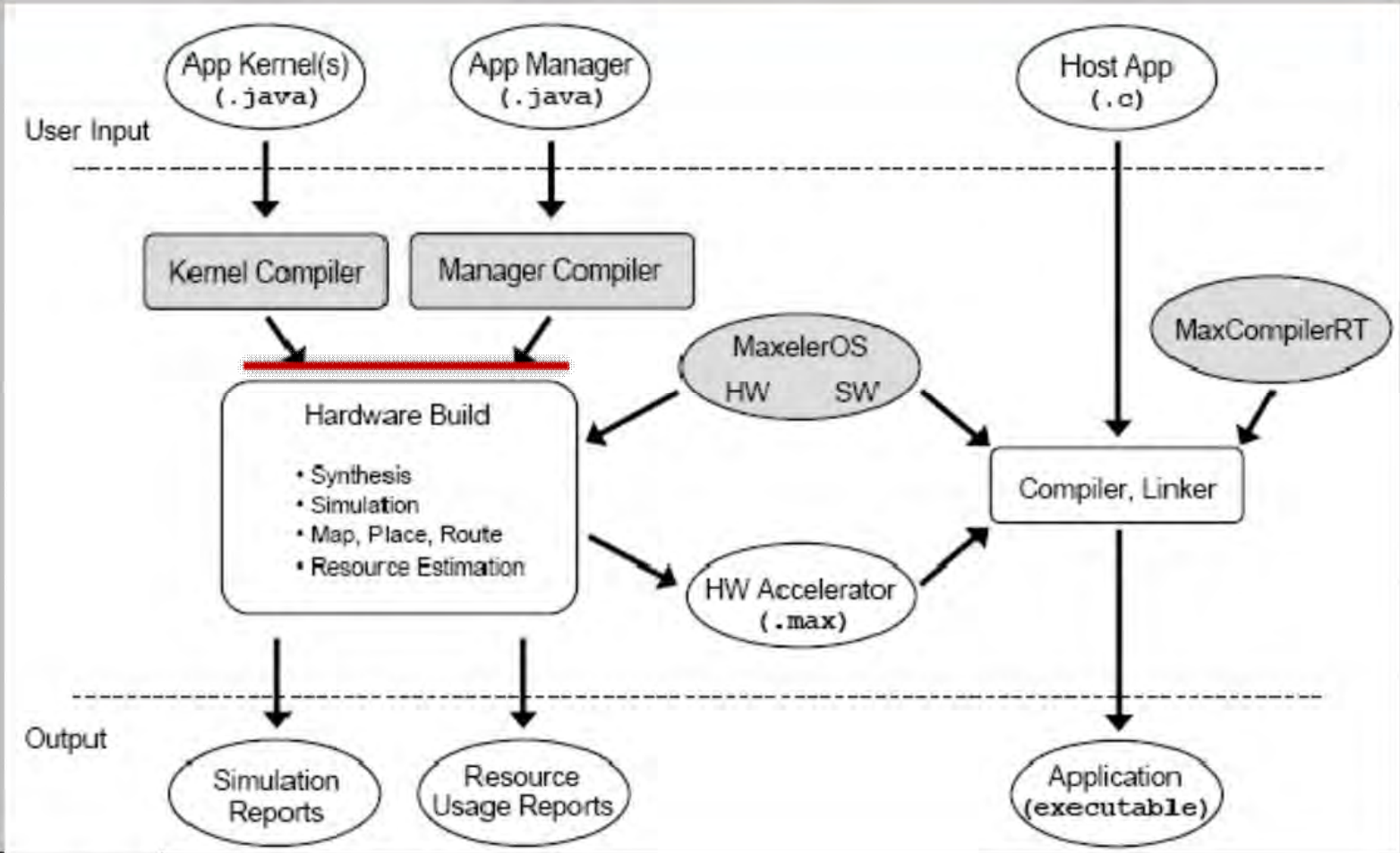
Why The Acceleration Approach?

Nobel Laureate Ilya Prigogine:
Injecting Energy to Decrease Entropy!

Corollary:
Burning energy to split spatial and temporal
decreases the entropy of computing
and enables the DataFlow compiler
to create a maximally effective execution graph.

Final goal:
The execution graph with the minimal length of edges.

MaxCompiler



.max -> ASIC brings 30% to 50% in Speedup and Power, at the expense of no reconfigurability and no flexibility!

Alliances Being Formed

- Intel acquired Altera
- Qualcomm and IBM teaming up with Xilinx

However:

C	C	C
OpenCL(I)	OpenCL(A)	MaxCompiler
Intel	Altera	Altera@MaxZ
1	X>>1	Y>>X

Nano Accelerators

- Invisible on the DataFlow Concept Level
- Invisible to DataFlow Programmers
- Visible to the MaxCompiler
- The MaxCompiler knows how to utilize them

Best protected by two aSoG (now FPGA) protection levels and two Vendor (e.g., Maxeler) protection levels!

Publications of Interest for NanoAcceleration

==========

1. *Flynn, M., Mencer, O., Milutinovic, V., Rakocevic, G., Stenstrom, P., Trobec, R., Valero, M.,*
   Moving from Petaflops (on Simple Benchmarks) to Petadata per Unit of Time and Power (On Sophisticated Benchmarks),
   Communications of the ACM (nano-acceleration), May 2013.

2. *Trobec, R., Vasiljevic, R., Tomasevic, M., Milutinovic, V., Beiveide, M., Valero, M.,*
   Interconnection Networks for SuperComputing,
   ACM Computing Surveys (nano-acceleration) , 2017.

---

3. *Milutinovic, V., Tomasevic, M., Markovic, B., Tremblay, M.,*
   The Split Temporal/Spatial Cache: Initial Performance Analysis,
   Proceedings of the SCIzzL-5, Santa Clara, California, USA, March 26, 1996, pp 72-78.

4. *Milutinovic, V., Tomasevic, M., Markovic, B., Tremblay, M.,*
   The Split Temporal/Spatial Cache: Initial Complexity Analysis,
   Proceedings of the SCIzzL-5, Santa Clara, California, USA, September, 1996.

---

5. *Milutinovic, V.,*
   A Comparison of Suboptimal Detection Algorithms Applied to the Additive Mix of Orthogonal Sinusoidal Signals,
   IEEE Transactions on Communications, Vol. COM-36, No. 5, May 1988, pp. 538-543.

6. *Milutinovic, V.,*
   Mapping of Neural Networks on the Honeycomb Architectures,
   Proceedings of the IEEE, Vol. 77, No 12, December 1989, pp. 1875-1878.

---

7. *Helbig, W., Milutinovic, V.,*
   The RCA's DCFL E/D MESFET GaAs 32-bit Experimental RISC Machine,
   IEEE Transactions on Computers, Vol. 36, No. 2, February 1989, pp. 263-274.

8. *Jovanovic, Z., Milutinovic, V.,*
   FPGA Accelerator for Floating-Point Matrix Multiplication,
   IEE Computers & Digital Techniques (nano-acceleration), 2012, 6, (4), pp. 249-256.
   The IET 2014 Premium Award for Computing & Digital Techniques.

==========

Inspired by:
Feynman

Inspired by:
Prigogine

Inspired by:
Kahneman

Inspired by:
Hunt

Feynman

proceedings OF THE IEEE
THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS december 198

$$l = \log_2(N_{pe}+1)$$

$$N_{total}(l) = (l-1)\left(2^{l-1}+2^{l-2}\right)$$

$$N_{bus}(l) = (l-2)\left(2^{l-2}+2^{l-3}\right)$$

$$N_{idle}(l) = (l-1)\left(2^{l-1}+2^{l-2}\right)-(l-2)\left(2^{l-2}+2^{l-3}\right)-\left(2^{l}-1\right)$$

$$N_{pe}(l) = 2^{l}-1$$

$$N_{total}(N_{pe}) = \left[\log_2(N_{pe}+1)-1\right]\left[\frac{N_{pe}+1}{2}+\frac{N_{pe}+1}{2^2}\right]$$

$$N_{total}(N_{pe}) = \left[\log_2(N_{pe}+1)-1\right]\left[\frac{3N_{pe}+3}{4}\right]$$

$$N_{total}(N_{pe}) = \frac{3}{4}(N_{pe}+1)\left[\log_2(N_{pe}+1)-1\right]$$

$$N_{bus}(l) = \frac{3}{8}(N_{pe}+1)\left[\log_2(N_{pe}+1)-2\right]$$

$$N_{idle}(N_{pe}) = \frac{3}{4}(N_{pe}+1)\left[\log_2(N_{pe}+1)-1\right]-\frac{3}{8}(N_{pe}+1)\left[\log_2(N_{pe}+1)-2\right]-N_{pe}$$

$$N_{idle}(l) = \frac{3}{8}(N_{pe}+1)\left[\log_2(N_{pe}+1)\right]-\frac{1}{8}(11N_{pe}+3)$$

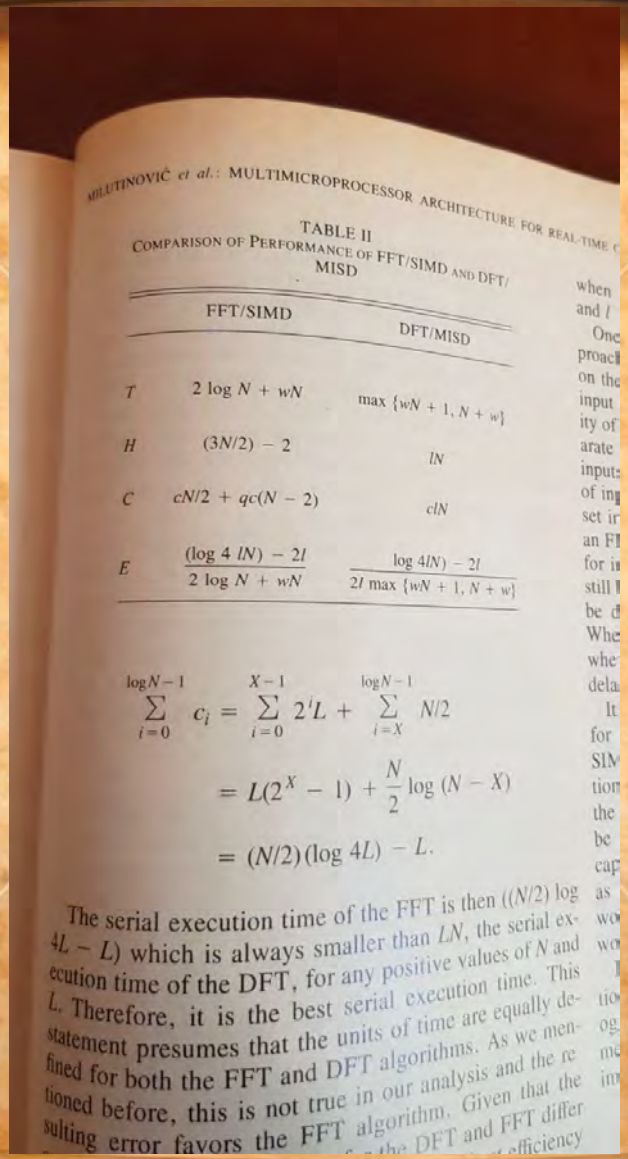
F.4.17.1

$$U(l) = \frac{N_{pe}(l)+N_{bus}(l)}{N_{total}(l)} = \frac{2^{l}-1+(l-2)\left(2^{l-2}+2^{l-3}\right)}{(l-1)\left(2^{l-1}+2^{l-2}\right)}$$

$$U(N_{pe}) = \frac{N_{pe}+\frac{3}{8}(N_{pe}+1)\left[\log_2(N_{pe}+1)-2\right]}{\frac{3}{4}(N_{pe}+1)\left[\log_2(N_{pe}+1)-1\right]}$$

F.4.18.1

Prigogine



MILUTINOVIĆ *et al.*: MULTIMICROPROCESSOR ARCHITECTURE FOR REAL-TIME

TABLE II
COMPARISON OF PERFORMANCE OF FFT/SIMD AND DFT/MISD

	FFT/SIMD	DFT/MISD
T	$2 \log N + wN$	$\max\{wN + 1, N + w\}$
H	$(3N/2) - 2$	lN
C	$cN/2 + qc(N - 2)$	clN
E	$\dfrac{(\log 4\,lN) - 2l}{2 \log N + wN}$	$\dfrac{\log 4lN) - 2l}{2l \max\{wN + 1, N + w\}}$

$$\sum_{i=0}^{\log N - 1} c_i = \sum_{i=0}^{X-1} 2^i L + \sum_{i=X}^{\log N - 1} N/2$$

$$= L(2^X - 1) + \frac{N}{2} \log (N - X)$$

$$= (N/2)(\log 4L) - L.$$

The serial execution time of the FFT is then $((N/2) \log$ $4L - L)$ which is always smaller than LN, the serial ex- ecution time of the DFT, for any positive values of N and L. Therefore, it is the best serial execution time. This statement presumes that the units of time are equally de- fined for both the FFT and DFT algorithms. As we men- tioned before, this is not true in our analysis and the re sulting error favors the FFT algorithm. Given that the

Kahneman

If $D_{SAS} > 0$ then a binary 0 is detected; otherwise a binary 1. The SAS avoids multiplication. Error probability for SAS is given by [9]

$$\lim_{M\to\infty} = Q\left[\frac{2E}{\alpha N_0}\right]^{1/2}. \tag{4}$$

Here E is defined by

$$E \equiv \int_0^T s^2(t)\,dt \tag{5}$$

and $Q(\alpha)$ is defined by

$$Q(\alpha) \equiv \frac{1}{\sqrt{2\pi}}\int_\alpha^\infty e^{-x^2/2}\,dx \tag{6}$$

where

$$\frac{N_0}{2}$$

refers to the two-sided power spectral density of the noise. For a sinusoidal waveform, the signal shape coefficient α is given by [9]

$$\alpha = \frac{\pi^2}{8}. \tag{7}$$

Performance degradation of SAS compared to DMF is equal to $10 \log_{10} \alpha$, and is dependent on the signal shape [9].

The WPD detection is based on the detection parameter D_{WPD} given by [5]

$$D_{WPD} = \sum_{i=1}^{M} \text{sign}\,(v_i) * |s_i|. \tag{8}$$

If $D_{WPD} > 0$ then a binary 0 is detected; otherwise a binary 1. The WPD also avoids multiplication. Error probability for WPD is given by [9]

$$\lim_{M\to\infty} P_e = Q\left(\frac{4E}{\pi N_0}\right)^{1/2}. \tag{9}$$

Performance degradation of WPD, compared to DMF, is equal to 1.96 dB, and is independent of the signal shape [9].

The BMF detection is based on the detection parameter D_{BMF} given by [3]

$$D_{BMF} \equiv \sum_{i=1}^{M} \text{sign}\,(v_i) * \text{sign}\,(s_i). \tag{10}$$

If $D_{BMF} > 0$ then a binary 0 is detected; otherwise a binary 1. The BMF avoids both multiplication and analog-to-digital conversion. Error probability for BMF is given by [9]

$$\lim_{M\to\infty} P_e = Q\left(\frac{4E}{\pi\alpha N_0}\right)^{1/2} \tag{11}$$

where $N_0/2$ refers to the two-sided power spectral density of the noise, and α refers to the signal shape [9]. Performance degradation of the BMF compared to the DMF is equal to 1.96 + $10 \log_{10} \alpha$, and is dependent on the signal shape.

The DMF detection is based on the detection parameter D_{DMF} given by [13]

$$D_{DMF} \equiv \sum_{i=1}^{M} v_i * s_i \tag{12}$$

If $D_{DMF} > 0$ then a binary 0 is detected; otherwise a binary 1.

Error probability for DMF is given by [2]

$$\lim_{M\to\infty} P_e = Q\left(\frac{2E}{N_0}\right)^{1/2}. \tag{13}$$

III. Additive Mix of Orthogonal Sinusoidal Signals

In this section, we introduce a concrete type of signal which is of interest for the analysis. We consider a specific form with $U = 32$ bits per signal frame of duration T, and $V = 1$ dibit (two bits) per carrier on the central frequency $f_n = 55 * (15 + 2n)$; $n = 1, \cdots, 16$. The analytic form of the signal is given by

$$s(t) = \sum_{n=1}^{N=16} \sin\,[2\pi f_n(t-kT) + \phi_{n,k}]$$
$$kT \leqq t \leqq (k+1)T$$
$$k = 0, 1, 2, \cdots. \tag{14}$$

Symbol $\phi_{n,k}$ refers to the phase of the signal on the central frequency f_n, during the signal frame with the index k. Therefore, n refers to the channel number and k refers to one particular signaling interval. Phase $\phi_{n,k}$ is given by

$$\phi_{n,k} = \frac{\pi}{4}\cdot l$$
$$l = 0, \cdots, 7$$
$$n = 1, \cdots, 16$$
$$k = 0, 1, 2, 3, \cdots. \tag{15}$$

As already indicated, this type of signal has been widely used in data transmission over the HF radio [2], and a similar type of signal can be used in other transmission media. In this type of signal, the variance of sample values is relatively large. Signal detection is based on the set of in-phase $[\Phi_{n,k}]$ and quadrature $[\Omega_{n,k}]$ projections of the receiving signal, which is distorted and corrupted by noise. These in-phase and quadrature components are given by

$$\Phi_{n,k} = \int_{kT+t_1}^{kT+t_1+T_0} s(t) * \cos\,[2\pi f_n(t-kT) + \phi_{n,k-1}]\,dt \tag{16}$$

$$\Omega_{n,k} = \int_{kT+t_1}^{kT+t_1+T_0} s(t) * \sin\,[2\pi f_n(t-kT) + \phi_{n,k-1}]\,dt \tag{17}$$

$$n = 1, \cdots, 16$$
$$k = 0, 1, 2, 3, \cdots.$$

Here t_1 is related to the beginning of the signaling interval. In the case of digital realization based on $J = 64$ samples, with the samples $s(t_j)$, $j = 0, 1, 2, \cdots, 64$ being $\Delta T_j = 1/(7040$ Hz) apart, we have

$$\phi_{n,k} = \sum_{j=1}^{j=64} s(t_j)\cdot\cos\,(2\pi f_n t_j + \phi_{n,k-1}) = \sum_{j=1}^{j=64} \phi_{j,n,k} \tag{18}$$

$$\Omega_{n,k} = \sum_{j=1}^{j=64} s(t_j)\cdot\sin\,(2\pi f_n t_j + \phi_{n,k-1}) = \sum_{j=1}^{j=64} \Omega_{j,n,k} \tag{19}$$

$$n = 1, \cdots, 16$$
$$k = 0, 1, 2, \cdots.$$

Finally, the phase difference between the received signal carrier $(\phi_{n,k})$ and the local oscillator $(\phi_{n,k-1})$ is measured. The

Hunt

In the case of DO looping, as indicated in Fig. 5, the computation of the new value for the loop control variable is followed by both writing the new value back into the multiport data memory, and testing the condition. The longer of these two operations will determine the total execution time.

$$T_{\mathrm{DO}} = \begin{cases} T_{\mathrm{GOTO}} + T_{A/C}(K_{A/C}=1,\ K_B;\ N_{\mathrm{OPER}} = N^{\mathrm{eff}}_{\mathrm{OPER}}); \\ \qquad (N_{G/\mathrm{RELA}} + N_{G/BR}) * T_G > T_{MM} \\ T_{\mathrm{GOTO}} + T_{A/C}(K_{A/C}=0;\ N_{\mathrm{OPER}} = N^{\mathrm{eff}}_{\mathrm{OPER}}); \\ \qquad (N_{G/\mathrm{RELA}} + N_{G/BR}) * T_G \le T_{MM} \end{cases} \tag{15}$$

where $N_{G/\mathrm{RELA}}$ is the number of gate delays for the unit which determines the value of the HLL relation, and $N^{\mathrm{eff}}_{\mathrm{OPER}} = \max \{(N_{OPER/I} + 1), N_{\mathrm{OPER}/F}, (N_{\mathrm{OPER}/S} + 1)\}$. Symbols I, F, and S refer to INITIAL, FINAL, and STEP expressions in the primary control statement. By this we conclude the derivation of execution-time formulas for assignments and control constructs.

Now we concentrate on execution-time formulas for call/return and low-level I/O. We assume the same number of input and output ports of the multport data memory (N_{MM}), and certain number of input and output subroutine parameters (N_{PARA}; $N_{\mathrm{PARA}} \neq N_{MM}$). Consequently,

$$\begin{aligned} T_{\mathrm{CALL/RETURN}} = T_{\mathrm{CALL}} + T_{\mathrm{RETURN}} = 2 \\ * \left[T_{PM} + \left(\max \left\{ 1, \left\lceil \frac{N_{\mathrm{PARA}}}{N_{MM}} \right\rceil \right\} - 1 \right) \right. \\ * \max \{ T_F,\ 2 * T_{MM} + T_G \} \\ + \max [N_{G/BR} * T_G],\ [\mathrm{sgn}\ (N_{\mathrm{PARA}}) \\ \left. * (2 * T_{MM} + T_G)] \} \right] \end{aligned} \tag{16}$$

https://www.maxeler.com/moving-from-petaflops-to-petadata/

MAXELER Technologies

Platforms Products Technology About Us

Search

MyMaxeler

Back...

Moving from Petaflops to Petadata

May 5, 2013

VIEWPOINT: Moving from Petaflops to Petadata

M. Flynn[‡||], O. Mencer[‡∓], V. Milutinovic[†], G. Rakocevic[§], P. Stenstrom[ϑ], R.Trobec[þ], M. Valero[£]

[‡]Maxeler Technologies, [||]Stanford University, [∓] Imperial College London, [†]University of Belgrade, [§]Mathematical Institute of the Serbian Academy of Sciences and Arts in Belgrade, [ϑ] Chalmers University of Technology, [þ]Jožef Stefan Institute, [£]Barcelona Supercomputing Centre

Communications of the ACM, Vol. 56 No. 5 May 2013, doi: 10.1145/2447976.2447989

Special Acknowledgements to: Simon Aglionby, Georgi Gaydadjiev, Itay Greenspon, and Nemanja Trifunovic

IF(2012)=3.80

Article Talk Read Edit View history Search

ACM Computing Surveys

From Wikipedia, the free encyclopedia

ACM Computing Surveys (CSUR) is a peer reviewed scientific journal published by the Association for Computing Machinery. The journal publishes survey articles and tutorials related to computer science and computing. It was founded in 1969; the first editor-in-chief was William S. Dorn.[1]

In ISI Journal Citation Reports, *ACM Computing Surveys* has the highest impact factor among all computer science journals.[2] In a 2008 ranking of computer science journals, *ACM Computing Surveys* received the highest rank "A*".[3]

ACM Computing Surveys	
Abbreviated title (ISO 4)	*ACM Comput. Surv.*
Discipline	Computer science
Language	English
Edited by	Sartaj K Sahni
Publication details	
Publisher	ACM (United States)
Publication history	1969–present
Frequency	Quarterly
Indexing	
ISSN	0360-0300 (print) 1557-7341 (web)
Links	

- Journal homepage
- Online access
- Online archive

See also [edit]

- *ACM Computing Reviews*

References [edit]

1. ^ Dorn, William S. (1969). "Editor's Preview...". *ACM Computing Surveys*. **1** (1): 2–5. doi:10.1145/356540.356542.
2. ^ "Journal Citation Reports". *ISI Web of Knowledge*. Retrieved 2009-10-03. "JCR Science Edition 2008"; subject categories "COMPUTER SCIENCE, ...".
3. ^ "Journal Rankings". *CORE: The Computing Research and Education Association of Australasia*. July 2008. Archived from the original on 29 March 2010. Retrieved 2010-03-19..

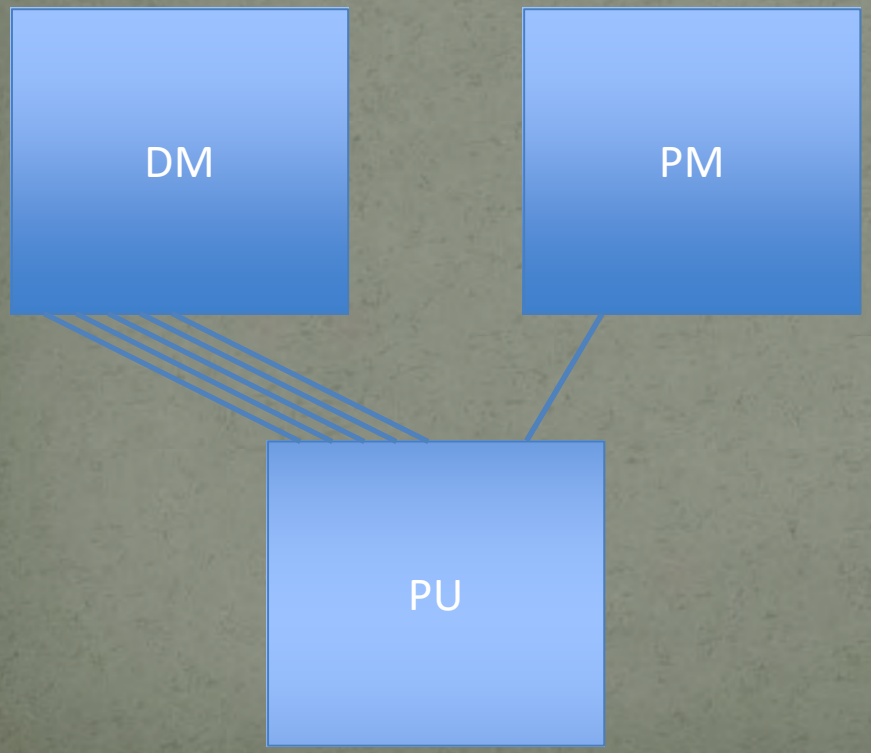
External links [edit]

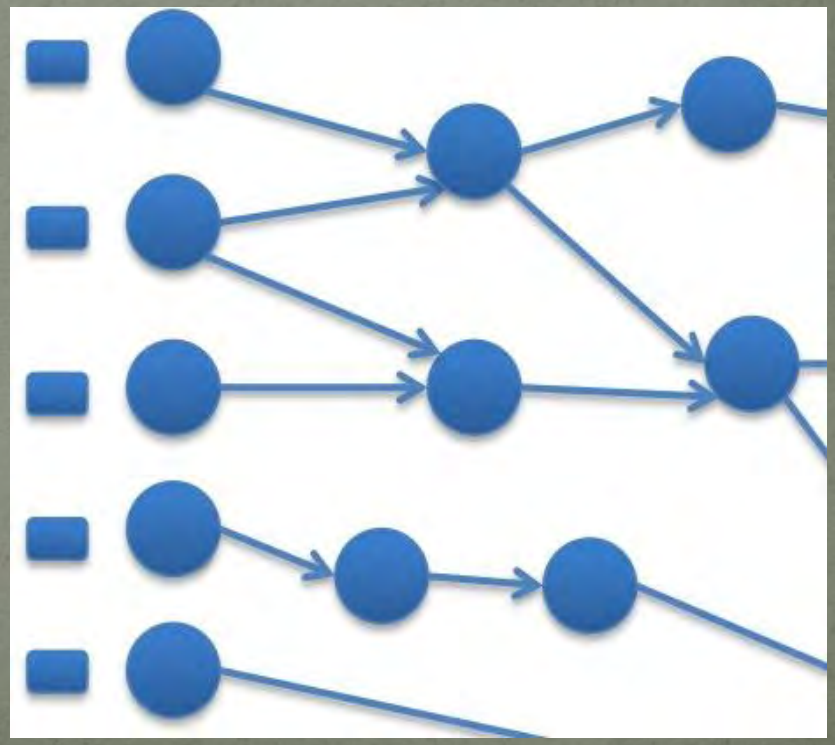
- *ACM Computing Surveys* home page.
- *ACM Computing Surveys* in ACM Digital Library.
- *ACM Computing Surveys* in DBLP.

IF (2007) = 15

Essence: Feynman Enabled by Prigogine

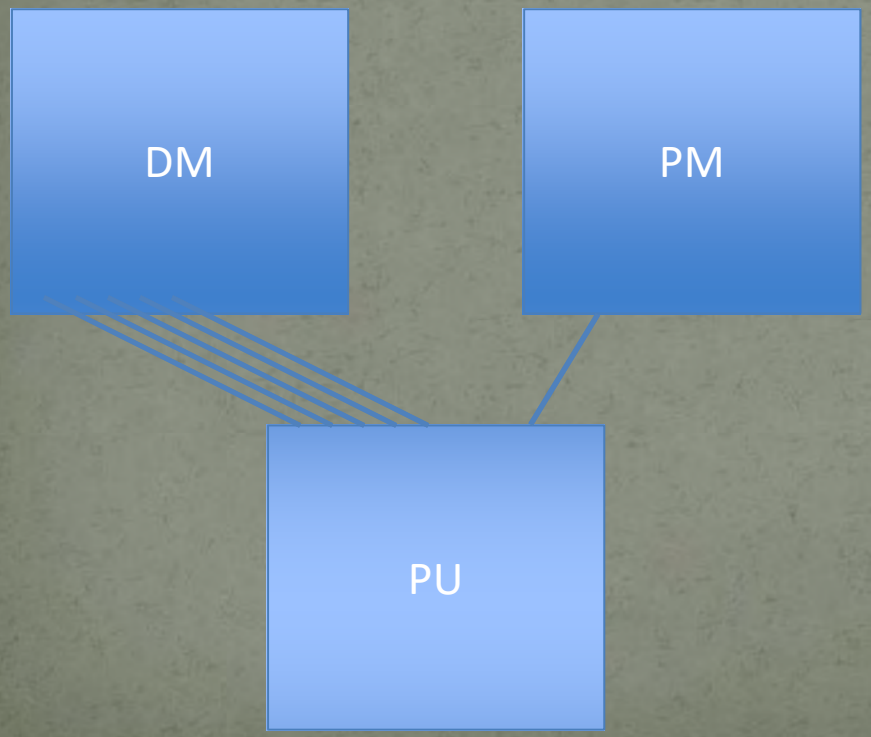
- TALU possible at zero power (Arithmetic+Logic)
- TCOMM not possible at zero power (MEM+MPS)

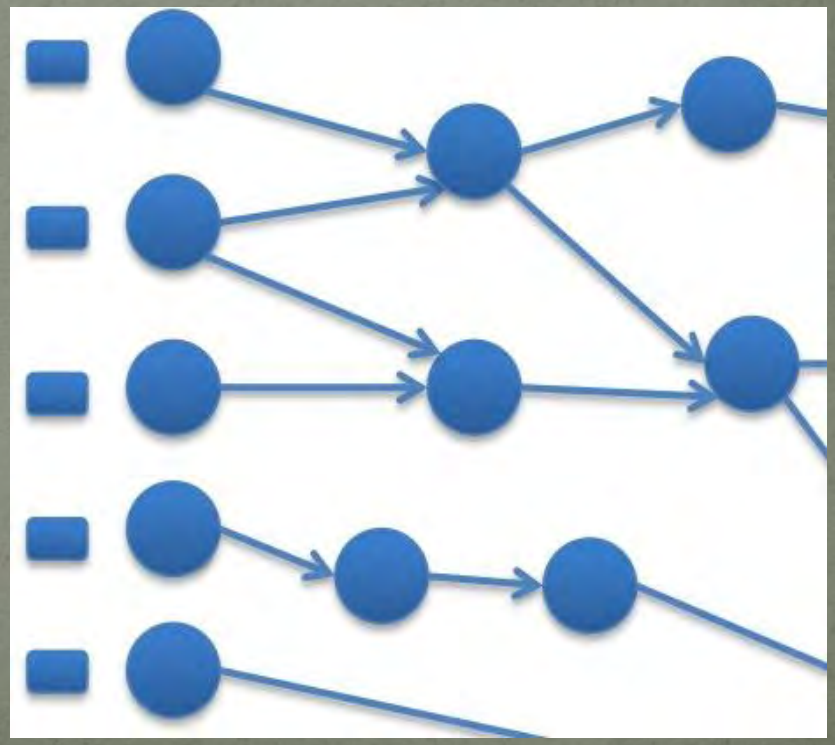




Essence: Feynman

- TALU possible at zero power (Arithmetic+Logic)
- TCOMM not possible at zero power (MEM+MPS)

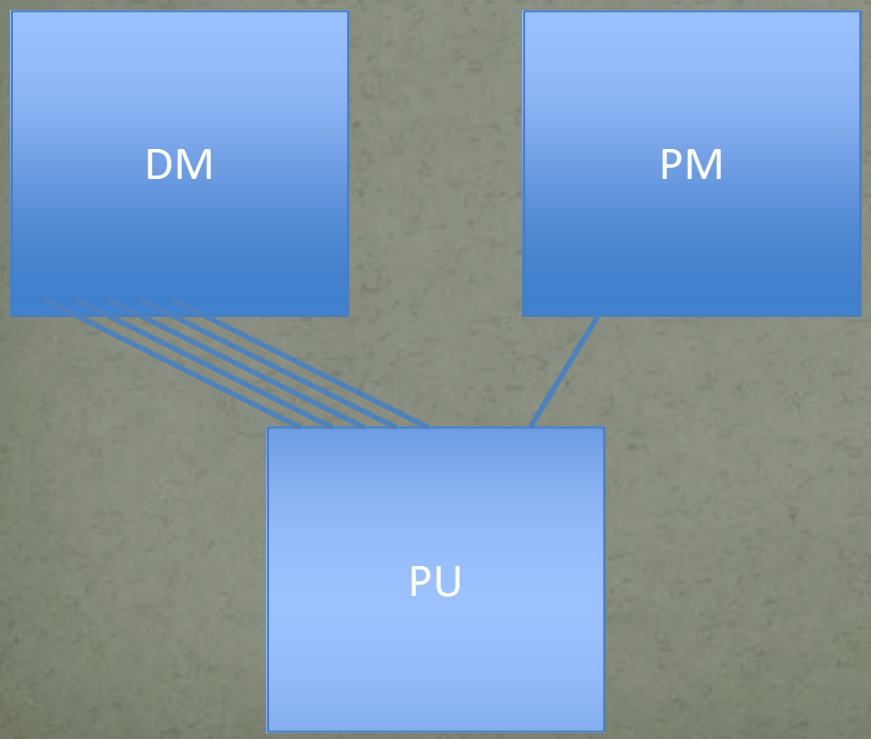


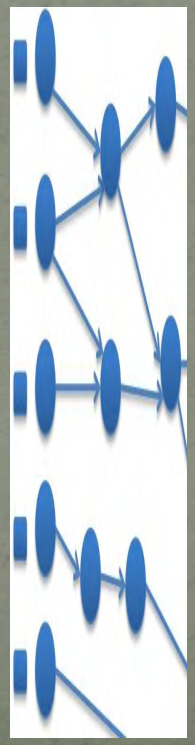


Essence: Feynman

Å TALU possible at zero power (Arithmetic+Logic)

Å TCOMM not possible at zero power (MEM+MPS)





Programming the Maxeler Technology Generic Acceleration Architecture

MaxJ, the Maxeler Java,

a DSL acting as a SuperSet of classical Java:

A. A vector of built-in domain-specific classes

B. Two sets of variables: SW + HW

MaxJ is a SubSet of OpenSPL,
created by the Imperial-Stanford-Tokyo-Tsinghua consortium.

Possible Future Mutations of OpenSPL:

MaxPython and/or MaxR
(lower Kolmogorov complexity)
MaxHaskel and/or MaxScala
(easier extension to approximate computing).

Approximate Computing for Better Precision: Kahneman

Note: Small approximations in one domain
may bring large benefits in another domain

Example: Weather forecast

A 15-bit computational precision (rather than the 64-bit precision)
may decrease the forecast precision for only 2%,
and at the same time,
may increase the grid precision 25 times,
and the forecast precision at grid intersections up to 10^4.

Easily doable in DataFlow, difficult to do in ControlFlow.

Delayed Decision for Better Precision: Hunt

Note: Small latencies in time domain
may bring large benefits in precision domains

Example: Optimal utilization of internal DataFlow pipelines

Compiler optimizations create internal pipelines
that experienced DataFlow programmers know how to utilize

BigDataAnalytics

Existing Maxeler-based publications:

20 [Size]
20 [Power]
20, 200 [Speedup]
20 [Precision]

Ultimate aSoG-based future:

20-200 [Size]
20-200 [Power]
20, 200, 2000, 20000 [Speedup]
20+ [Precision]

Applications

Architecture

Technology

Maxeler Dataflow Appliance

40U: Good for Cloud

- Software Based Solution
- Dataflow Computing in the Datacentre

MPC C Series

The CPU

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

MPC X Series

The Dataflow Appliance

Dense compute with 8 DFEs, 768GB of RAM and dynamic allocation of DFEs to CPU servers with zero-copy RDMA access

MPC N Series

The Networking Appliance

Intel Xeon CPUs and 4 DFEs with direct links to up to twelve 40Gbit Ethernet connections

1U: Good for Fog

MicroMAX.5: Good for Dew (Edge Processing of IoT Data)

The Major Application Successes

- Finances:
  - Credit derivatives
  - Risk assessment
  - Stability of economical systems
  - Evaluation of econo-political mechanisms
- GeoPhysics:
  - Oil&Gas
  - Weather forecast
  - Astronomy
  - Climate changes
- Science:
  - Physics
  - Chemistry
  - Biology
  - Genomics
- Engineering: Synergy of all the Above (ML, etc...)

Technology ignites our business. Be the spark.

J.P.Morgan

Innovation in Investment Banking Technology

Field Programmable Gate Arrays (FPGAs)

A Field Programmable Gate Array (FPGA) is **a silicon chip containing a matrix of configurable logic blocks (CLBs) that are connected through programmable interconnects.** By combining optimized use of available silicon with fine-grained parallelism, sustained acceleration improvements of over 300x can be achieved across a range of vanilla and complex mathematical models. The current work is the first time that FPGA technology has been employed at this scale to accelerate computational performance anywhere in the finance industry.

Power and Versatility

- Can accelerate performance by between 100 and 1,000x across a range of mathematical models, with the ability to perform a task in less than a second
- Can be reprogrammed and precisely configured to compute exact algorithm(s) at the desired level of numerical accuracy required by any given application, unlike normal microprocessors whose design is fixed by the manufacturer
- Can be deeply pipelined to achieve maximum parallelism from arithmetic, algorithms and data streaming

Key Business Challenges

- Reduce the execution time of existing applications to meet business and regulatory demands
- Decrease cost of running existing applications and developing new ones
- Provide fast, cost-effective extra computational capacity to address problems that are currently inextricable
- Achieve a step-change improvement in price-performance and end-to-end compute time across many applications

Key Benefits (Business/Clients)

- Competitive advantage to valuation, execution, risk management and complex scenario analyses by speeding up existing applications
- Lower cost of existing applications as hardware costs can be reduced by a factor between 100 and 1,000
- Ability to perform previously difficult calculations, such as complex trading strategies or risk evaluations of global portfolio simulations.

Technology Overview

- Low clock speed chips
- Maximal usage of available silicon resources
- Acceleration through use of fine-grained parallelism
- Reconfigurable hardware
- Silicon configurable to fit algorithm

LOB/Function(s) Impacted

- Credit & interest rates
- Equities & commodities
- Loan & mortgage modeling
- Finance & accounting
- High frequency trading
- Risk management & VaR

Industry/External Recognition

- Used by Cisco in all routers
- Simulation of real and theoretical systems
- Geophysics for oil and gas exploration
- Astrophysics & hydrodynamics
- Defense for cryptography
- Video games
- Genotyping

Technology ignites our business. Be the spark.

J.P.Morgan

Functionality Overview

Double precision floating point-capable FPGAs became commercially available in 2002, but it was the arrival of the Virtex 5 and 6 series chips from market leader Xilinx that really provided the scale required for the development of production-grade accelerated solutions. Using FPGAs in high performance compute solutions provides distinct advantages over conventional CPU clusters.

Operational Advantages

- Significantly increases performance for two main types of applications: those based around highly complex mathematical models and those using simpler algorithms that can be massively parallelized
- Enables a dramatic increase in compute density per cubic meter by using FPGAs as computational accelerators
- Consumes around 1% of the power of a single CPU core

Performance Improvements

- Performance improvements in the range 200-300x faster than the existing CPU cores used on the Compute BackBone (CBB) have been achieved in credit and interest rates hybrids businesses
- In equities, direct market access can run risk and loan stock at wire speed (3.5 micro secs) using a low-latency FPGA solution
- Benchmarked average throughput for J.P. Morgan's existing 40-node hybrid FPGA machine of 984MFlops/watt/cubic meter
- Potential standing at the top of the Green-500 ecological global supercomputer performance table

Development/Delivery

Timeline

- Initial porting of an algorithm can vary from one to three months depending on complexity.
- Production capabilities then depend on the scale of the application and the scope and intensity of the testing and reconciliation cycle

Partners

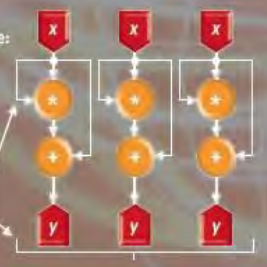
- London-based Applied Analytics group: includes three technology and business specialists with extensive experience in developing and delivering high performance solutions across a range of asset classes, models and lines of business
- Maxeler Technologies: external consultants trained in Imperial College, Stanford and MIT research labs

FPGAs at Work

- An algorithm is implemented as a special configuration of a general purpose electric circuit
- Connections between prefabricated wires are programmable
- Function of calculating elements is itself programmable
- FPGAs are two dimensional matrix-structures of configurable logic blocks (CLBs) surrounded by input/output blocks that enable communication with the rest of the environment

A very simple sample:

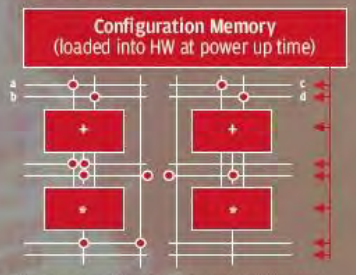
f(x) = 2x + x



Moving from a single calculation to a fine grained parallelism

A slightly more complex example:

e = (a+b)(c+d)*



Migrating algorithms from C++ to FPGAs involves doing a Fourier Transform from time domain execution to spatial domain execution in order to maximize computational throughput. **It's a paradigm shift to stream computing that provides acceleration of up to 1,000x compared to an Intel CPU.**

The know-how needed for deep security!

DSF Analytics - CME Grou

www.cmegroup.com/trading/interest-rates/dsf-analytics.html

Designed for educational use only using Maxeler Technologies' curve construction methodology. This tool uses delayed data and displayed results are indicative representations only.

Please hover your mouse pointer over column titles and links for further information.

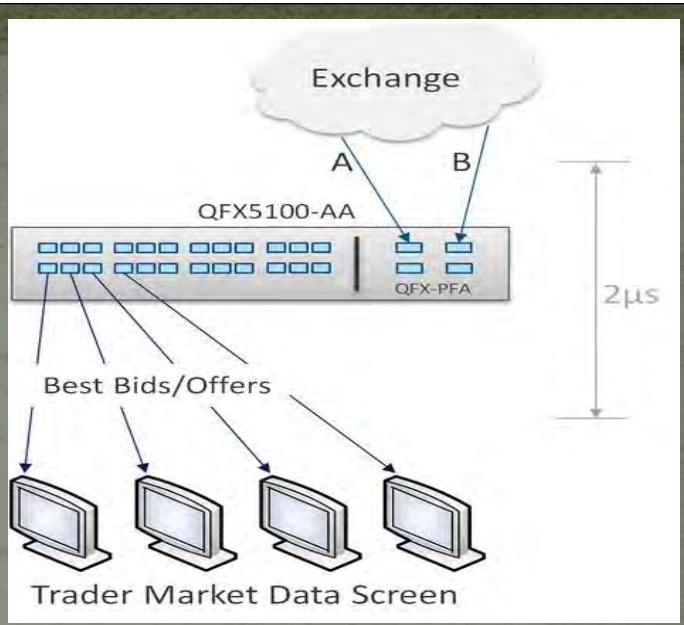
CME Ticker	Bloomberg Ticker	DSF Pricing					Timestamp
		Price	Coupon	PV01	NPV	Implied Rate	
T1UM4 2Y	CTPM4	100'057	0.750%	$19.97	$179.69	0.6600%	4:00:03 PM CT 4/4/2014
F1UM4 5Y	CFPM4	100'115	2.000%	$48.49	$359.38	1.9259%	4:00:03 PM CT 4/4/2014
N1UM4 10Y	CNPM4	100'225	3.000%	$90.16	$703.12	2.9220%	4:00:03 PM CT 4/4/2014
B1UM4 30Y	CBPM4	102'270	3.750%	$195.07	$2,843.75	3.6042%	4:00:03 PM CT 4/4/2014
T1UU4 2Y	CTPU4	100'085	1.000%	$19.93	$265.62	0.8668%	4:00:03 PM CT 4/4/2014
F1UU4 5Y	CFPU4	100'110	2.250%	$48.27	$343.75	2.1788%	4:00:03 PM CT 4/4/2014
N1UU4 10Y	CNPU4	101'125	3.250%	$89.55	$1,390.62	3.0948%	4:00:03 PM CT 4/4/2014
B1UU4 30Y	CBPU4	106'020	4.000%	$193.47	$6,062.50	3.6868%	4:00:03 PM CT 4/4/2014

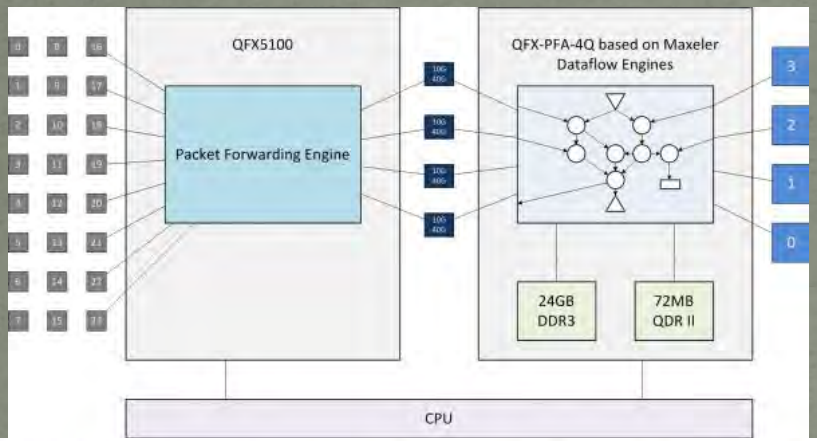
Quotes and analytics are updated every 15 minutes.

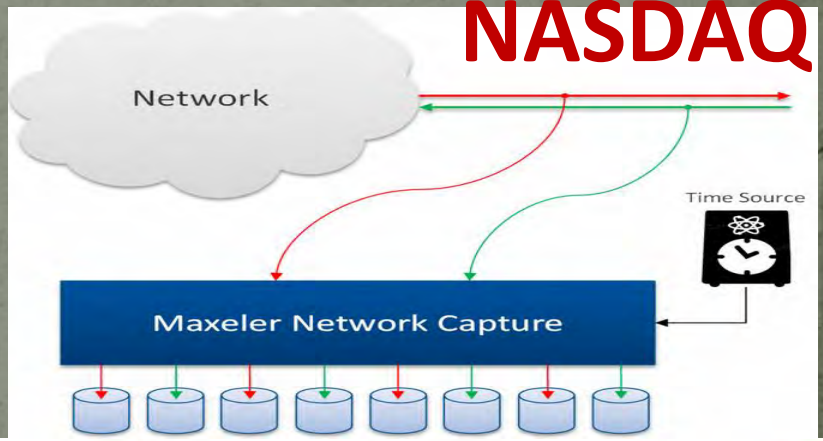
Analytics powered by Maxeler Technologies©

Instrument	CPU 1U-Node	Max 1U-Node	Comparison
European Swaptions	848,000	35,544,000	42x
American Options	38,400,000	720,000,000	19x
European Options	32,000,000	7,080,000,000	221x
Bermudan Swaptions	296	6,666	23x
Vanilla Swaps	176,000	32,800,000	186x
CDS	432,000	13,904,000	32x
CDS Bootstrap	14,000	872,000	62x

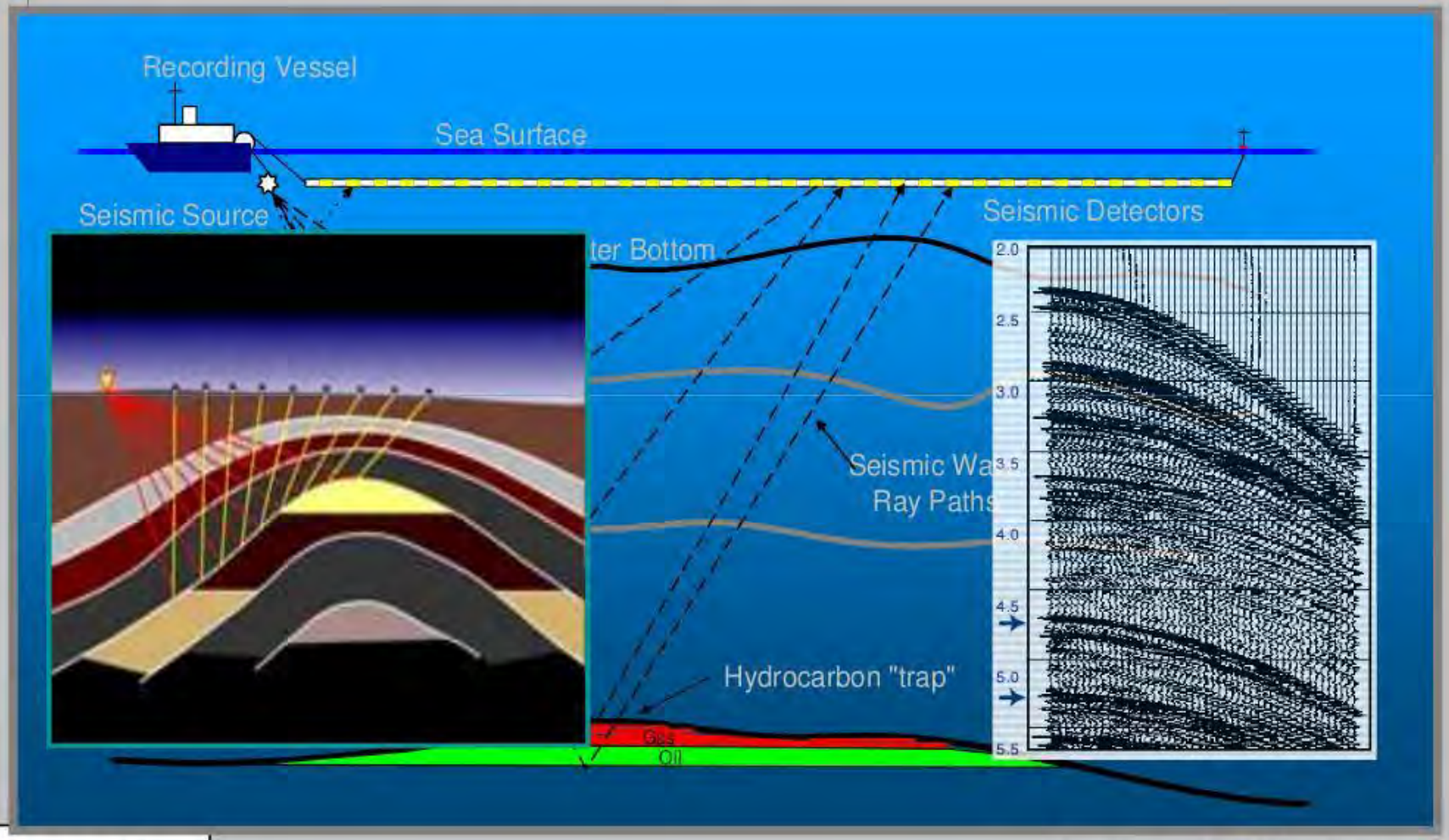
Juniper for Online Trading







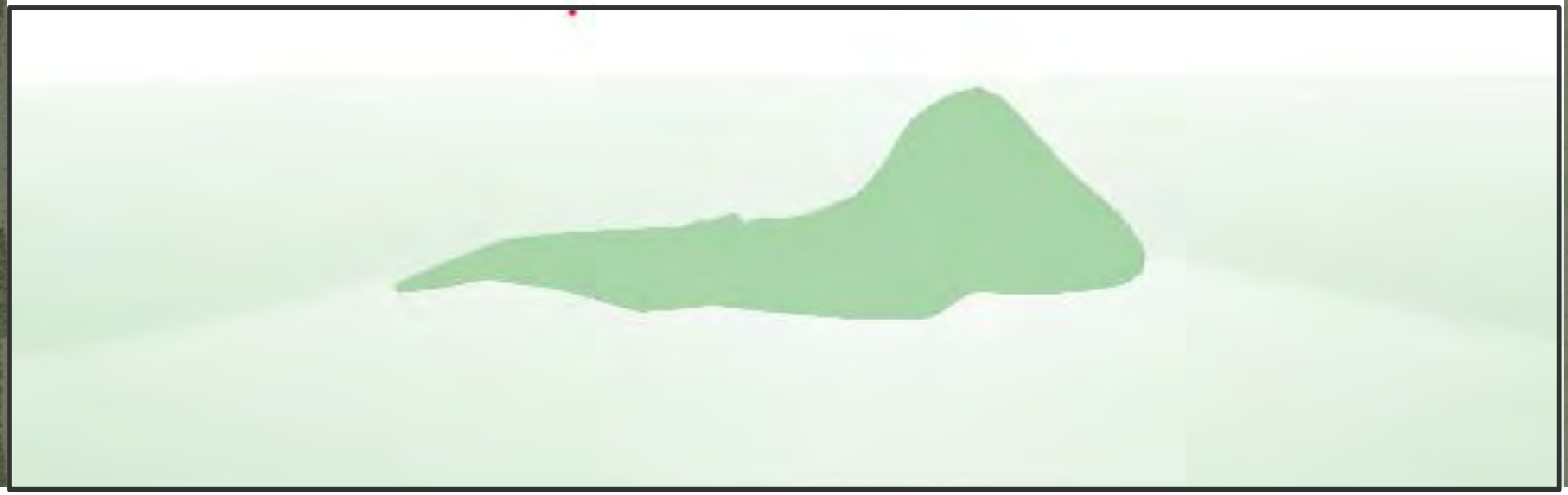
Seismic Data Acquisition



Courtesy of Schlumberger

MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING

Seismic Imaging



Å Running on MaxNode servers

- 8 parallel compute pipelines per chip
- 10x less power: 150MHz vs 1.5GHz
- 30x faster than microprocessors

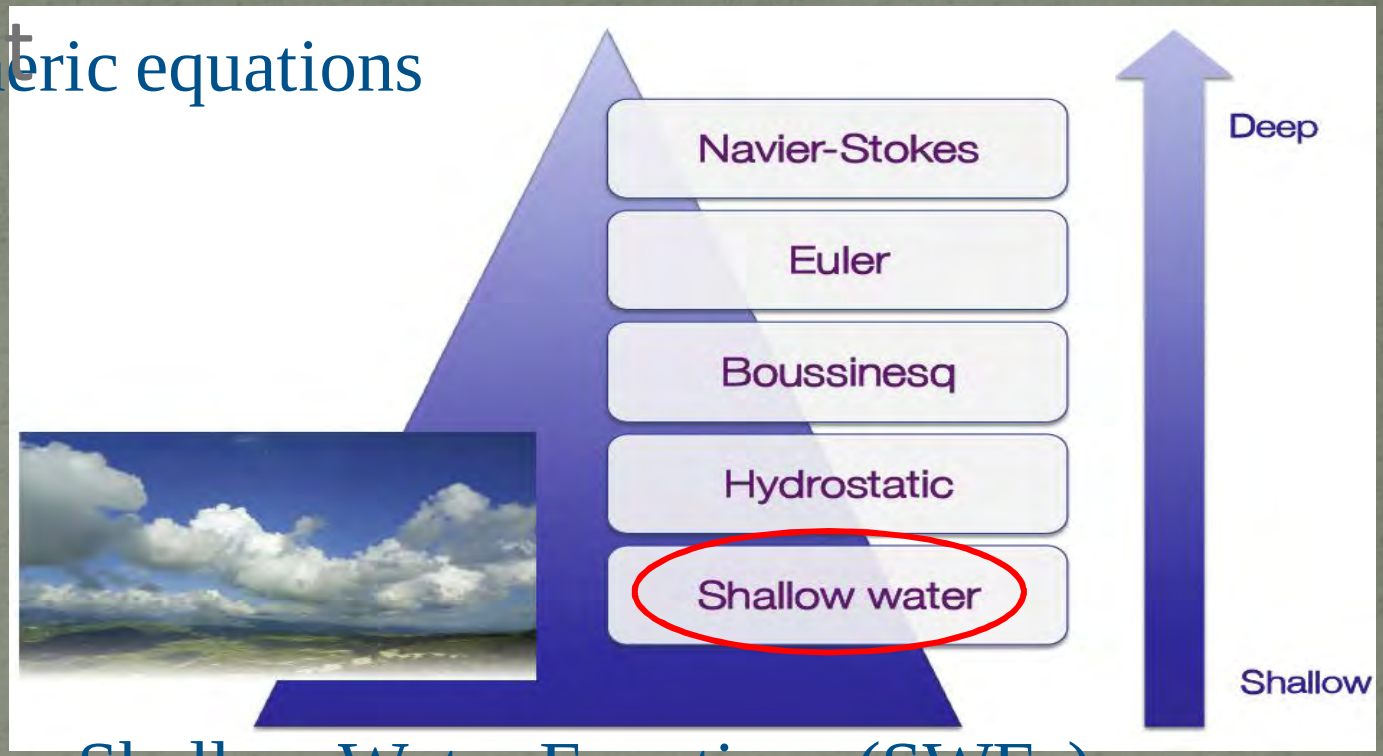
An Implementation of the Acoustic Wave Equation on FPGAs

T. Nemeth[†], J. Stefani[†], W. Liu[†], R. Dimond[‡], O. Pell[‡], R.Ergas[§]

[†]Chevron, [‡]Maxeler, [§]Formerly Chevron, SEG 2008

Global Weather Simulation: Size is Relevant

- Atmospheric equations



- Equations: Shallow Water Equations (SWEs)

$$\frac{\partial Q}{\partial t} + \frac{1}{\Lambda}\frac{\partial(\Lambda F^1)}{\partial x^1} + \frac{1}{\Lambda}\frac{\partial(\Lambda F^1)}{\partial x^2} + S = 0$$

[L. Gan, H. Fu, W. Luk, C. Yang, W. Xue, X. Huang, Y. Zhang, and G. Yang, Accelerating solvers for global atmospheric equations through mixed-precision data flow engine, FPL2013]

Weather Model – Performance Gain

Platform	Performance ()	Speedup
6-core CPU	4.66K	1
Tianhe-1A node	110.38K	23x
MaxWorkstation	468.1K	**100x**
MaxNode	1.54M	**330x**

14x

Meshsize: 1024 × 1024 × 6

MaxNode speedup over Tianhe node: 14 times

Weather Model -- Power Efficiency

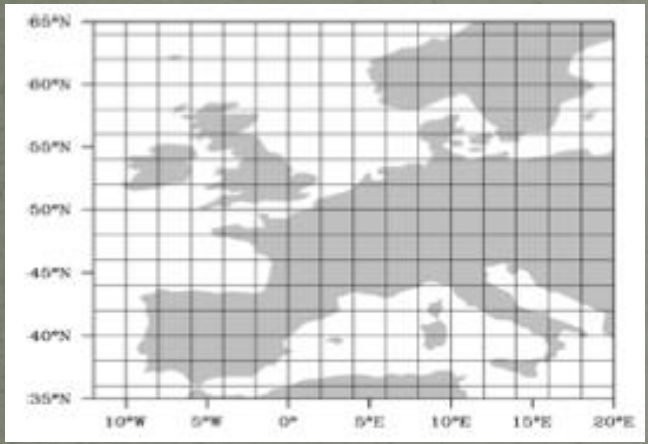
Platform	Power Efficiency ( )	Speedup
6-core CPU	20.71	1
Tianhe-1A node	306.6	14.8x
MaxWorkstation	2.52K	**121.6x**
MaxNode	3K	**144.9x**

9 x

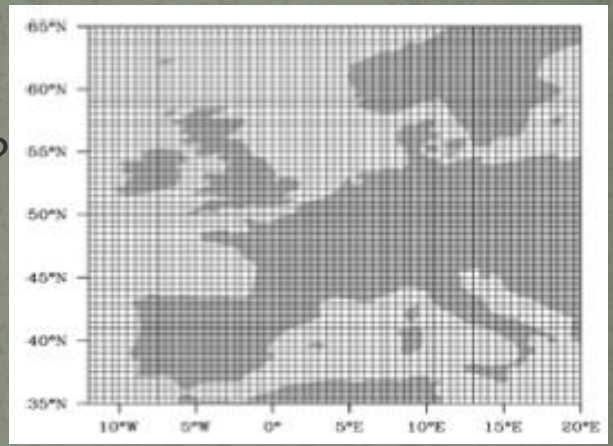
Meshsize: $1024 \times 1024 \times 6$

MaxNode is 9 times more power efficient

Weather and Climate Models: Precision



Which one is better?

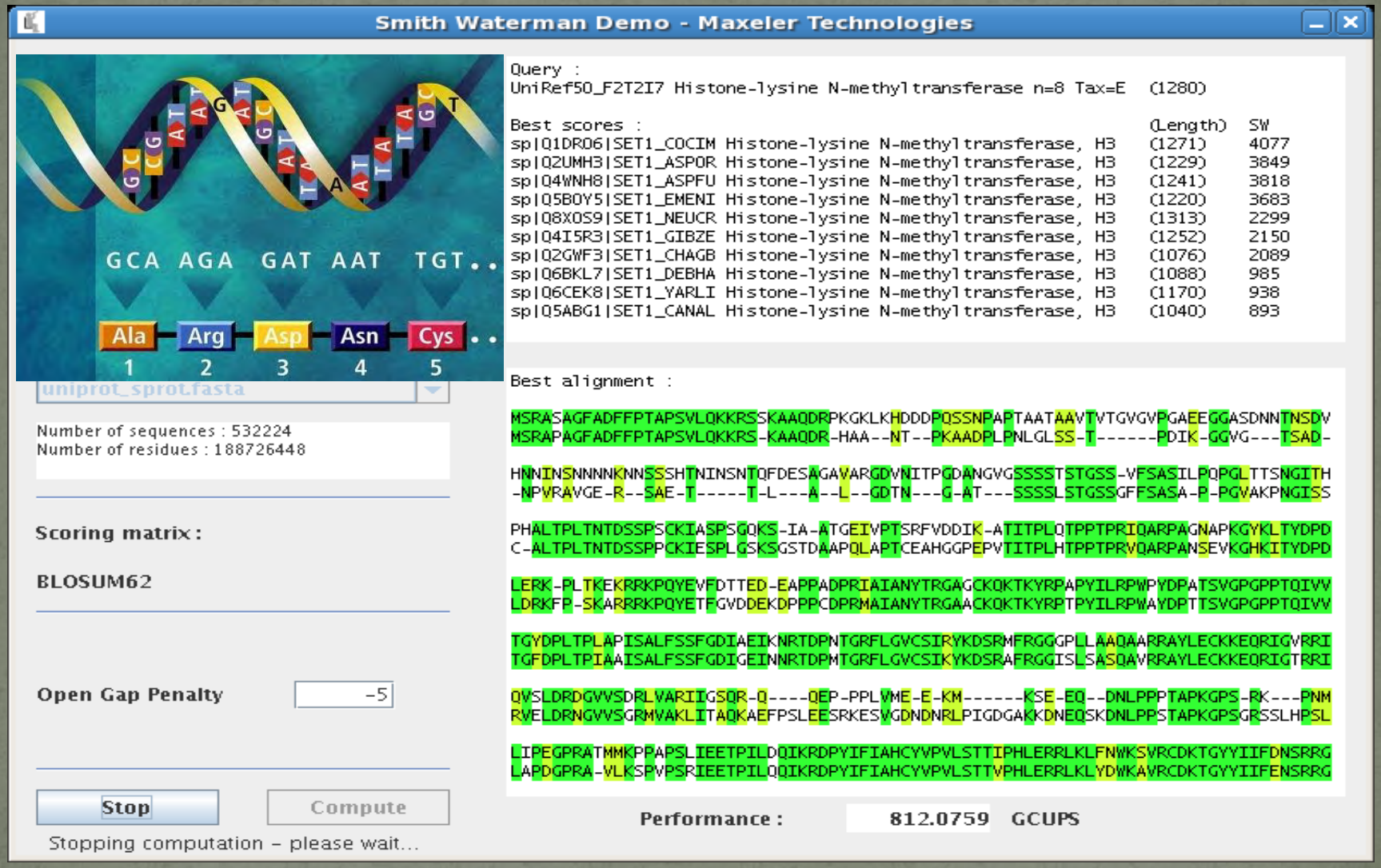


Finer grid and higher precision are obviously preferred
but the computational requirements will increase ❾ Power usage → $$

What about using reduced precision? (15 bits instead of 64 double precision FP)

We use only **15 bits** for 98% of the computation:

Maxeler Running Smith Waterman

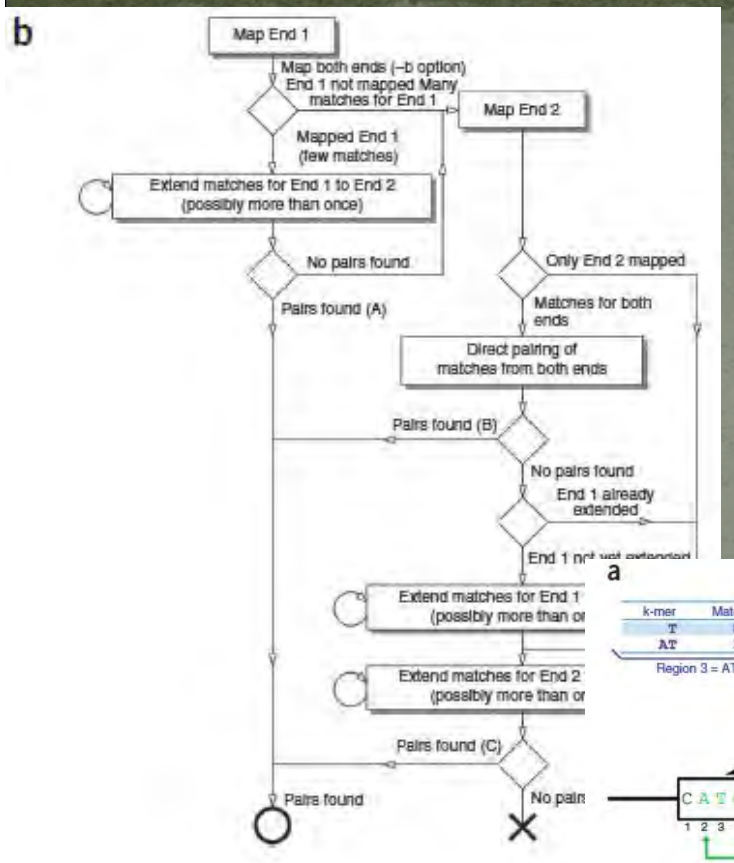


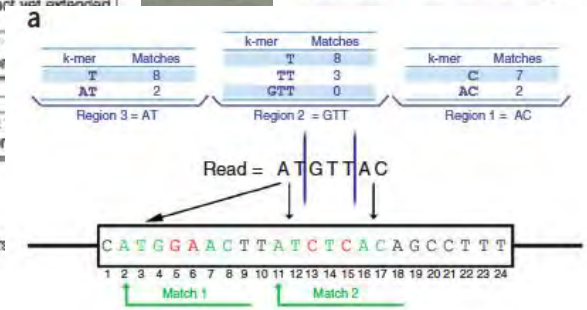
THE UNIVERSITY OF TOKYO

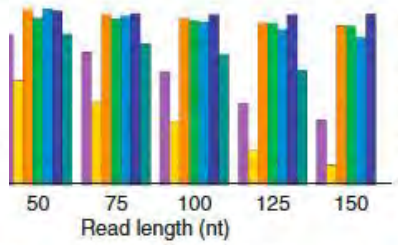
The GEM mapper: fast, accurate and versatile alignment by filtration

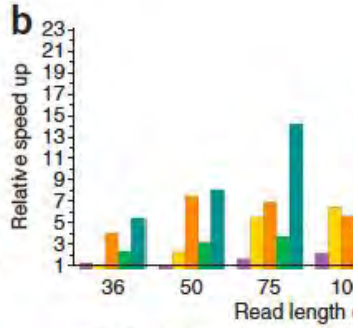
Niederman-Wuensch

Santiago Marco-Sola[1], Michael Sammeth[1], Roderic Guigó[2] & Paolo Ribeca[1,3]



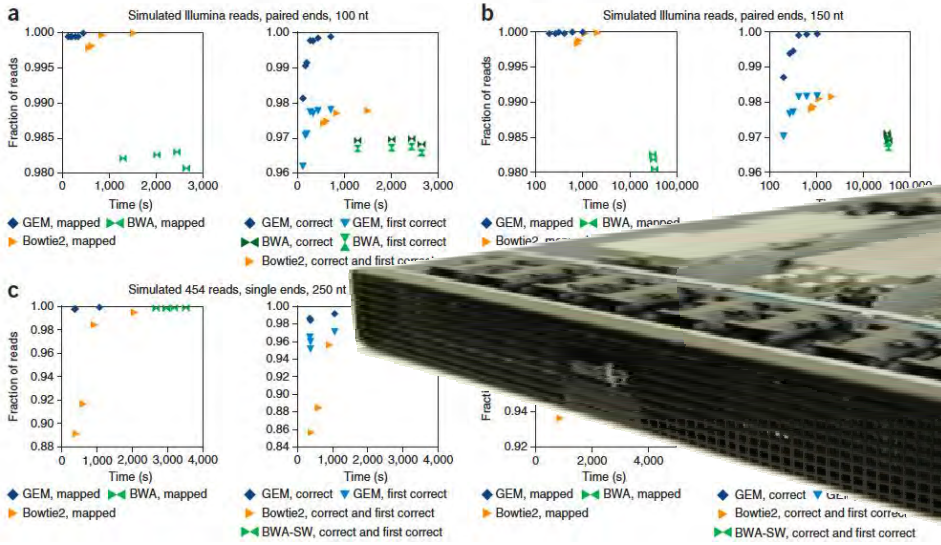


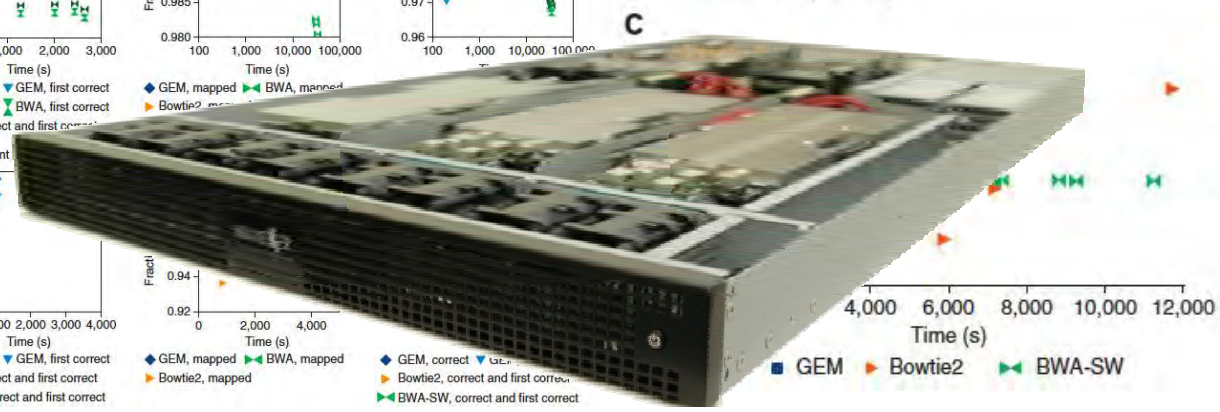


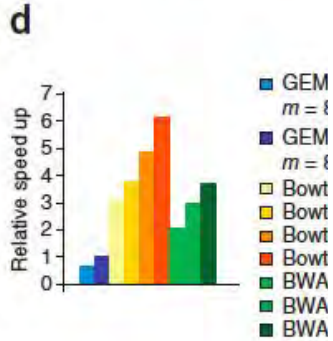


SOAP2 ($m = 3$)
Bowtie2 (very-sensitive)
GEM ($m = 3$)
MrsFAST ($m = 4\%$)
Bowtie ($m = 3$)
BWA ($m = 4\%$)
GEM ($m = 4\%$, e = 4%)

SOAP2 ($m = 3$)
Bowtie2 (very-fast)
MrsFAST ($m = 4\%$)

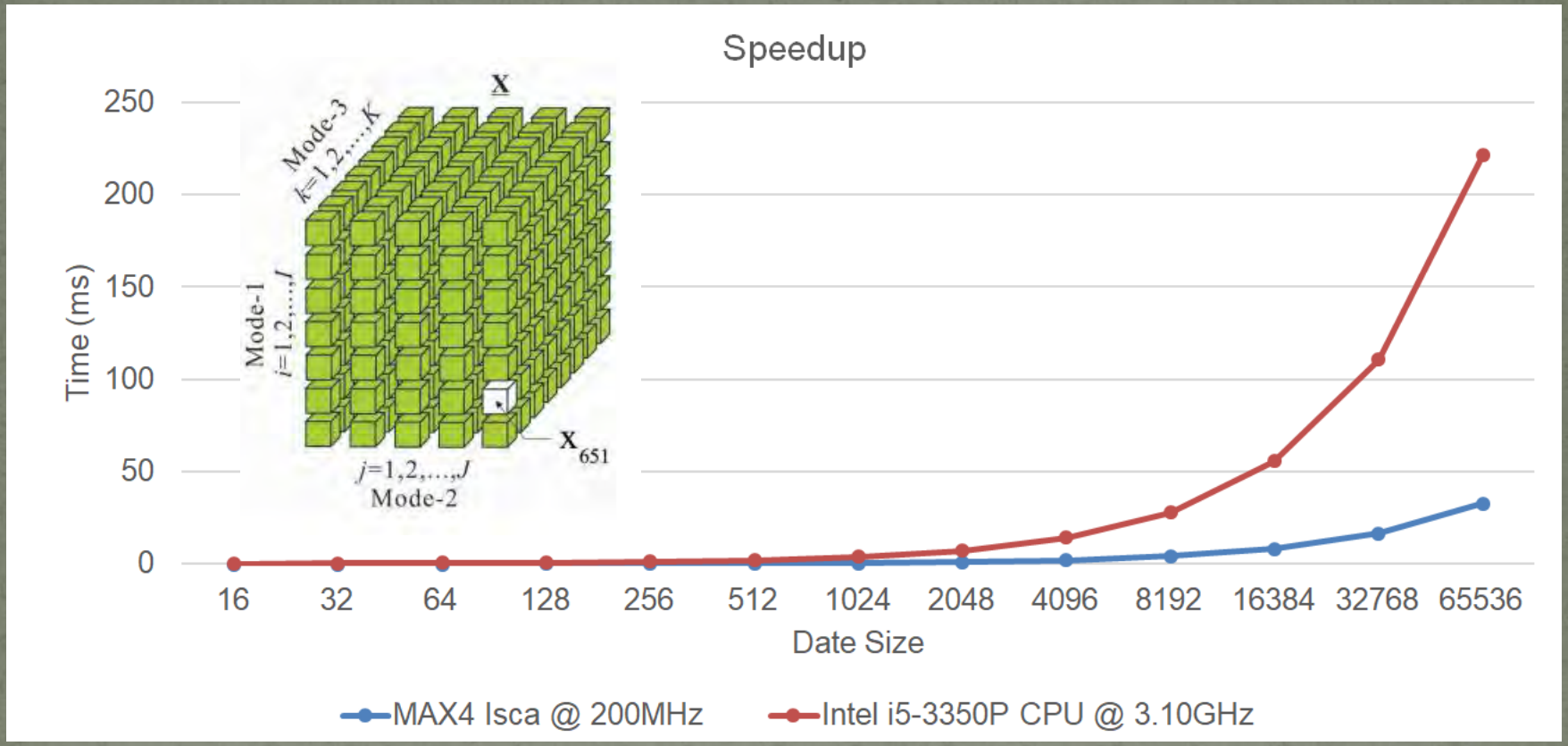






Analysis of the Tensor Calculus Operations on DataFlow

(PhD Thesis by Milog Kotlar, on DataFlow-based Machine Learning)

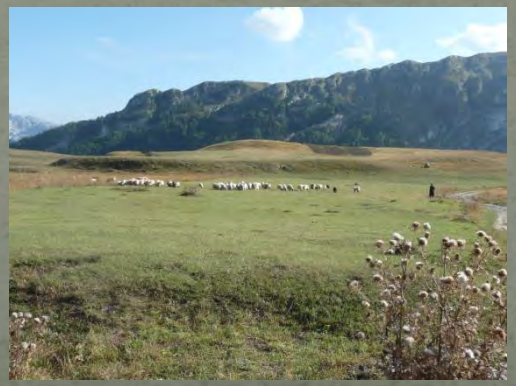


The speedup of **6.75x** achieved as early as for KiloData (Perceptron), with **10x** less on-chip transistors and the power savings of **4.6x**

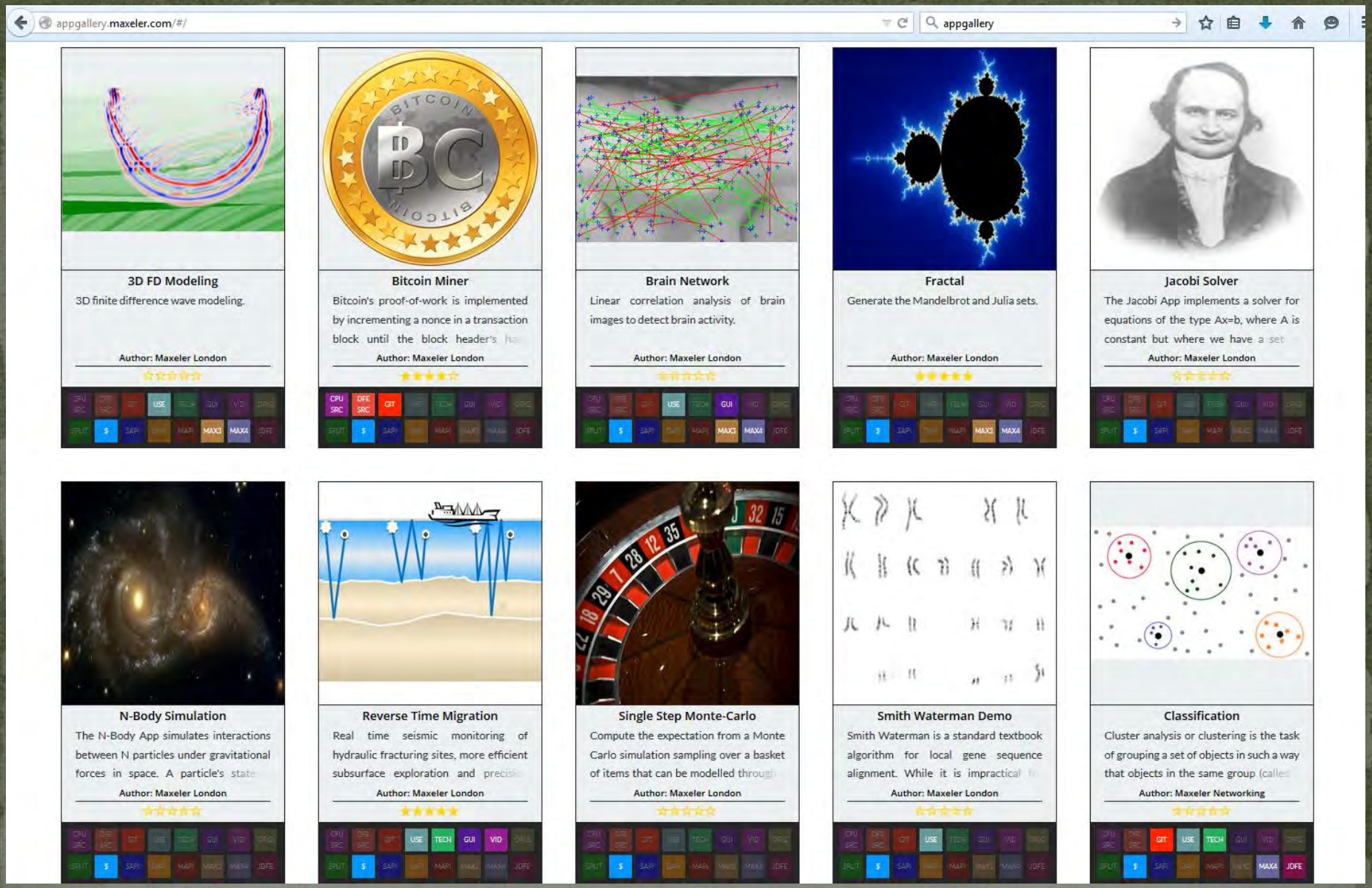
Conditions for the Y-Chart-Based iKernelization of Loops @ML

(PhD Thesis by Nenad Korolija, on the Mapping of Algorithms onto DataFlow)

1.	BigData (RAM vs. STREAM)	$O(n^2)$
2.	Code reusability (WORO vs. WORM)	+
3.	Overall application tolerance to latency	+
4.	Over 95% of run time in loops	++
5.	Reusability of the data in loops	++
6.	Potential for utilization of pipes	O(n)



Essentials for speedup:
algorithmic modifications,
pipeline utilization,
data choreography,
decision making on precision



appgallery.maxeler.com

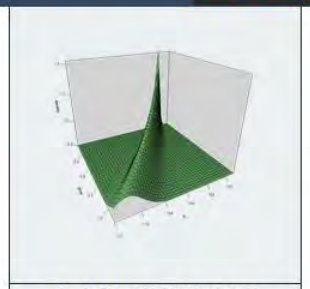
http://www.mi.sanu.ac.rs/~appgallery.maxeler/

Apps | Maxeler AppGallery

appgallery.maxeler.com/#/

Search

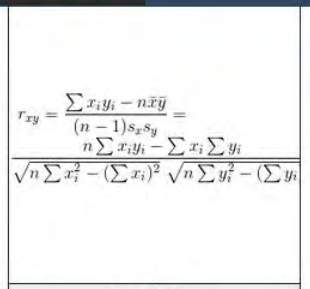
Apps | Filter by: | Maximum Performance Eco-System, Beta Release (v0.1) | Log In



Black-Scholes Option Pricer

This App uses a Monte-Carlo simulation for tail risk analysis. Given a horizon time, a set of up to 6 underliers and

Author: Maxeler Analytics

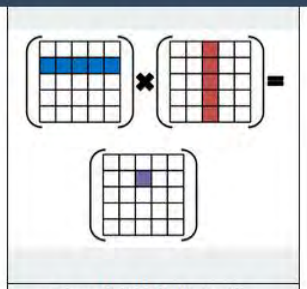


$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{(n-1)s_x s_y} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\,\sqrt{n\sum y_i^2 - (\sum y_i}}$$

Correlation

Correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together

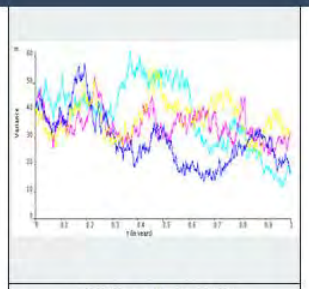
Author: Maxeler Analytics



Dense Matrix Multiplication

A matrix is dense if most of its elements are non-zero. Multiplication involves a dot product between every row of

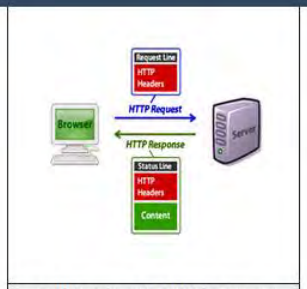
Author: Maxeler Analytics



Heston option pricer

Monte Carlo options pricer.

Author: Maxeler Analytics



Low-Latency HTTP Web-Server

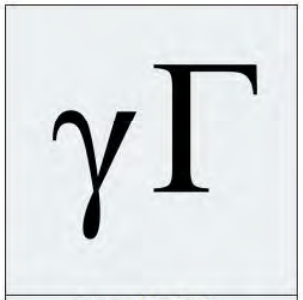
This App implements an HTTP Web-Server in a DFE. The App serves static webpages directly from LMEM

Author: Maxeler Belgrade

Non-Central Chi Square

Non-Central Chi Square Distribution, that generate lots of sample from complex distribution.

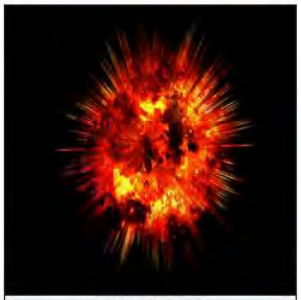
Author: Maxeler California



Gamma Distribution

Generate random numbers according to the Gamma statistical distribution, using the Gamma rejection method. As

Author: Maxeler Intern



Heat Equation

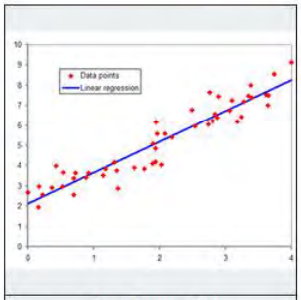
Simulate heat spreading according to explicit methods, which is characterized by the following equation: T(p, t+1) =

Author: Maxeler Intern

Implicit Heat Equation

Simulate heat spreading according to implicit methods, which is characterized by the following equations: b = T(p,

Author: Maxeler Intern



Linear Regression

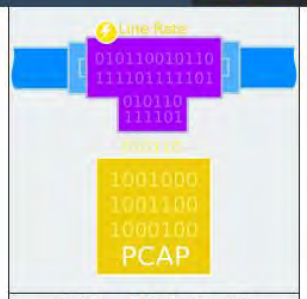
In statistics, linear regression is an approach for modeling the relationship between a scalar dependent variable a

Author: Maxeler Intern

Apps | Maxeler AppGallery

appgallery.maxeler.com/#/

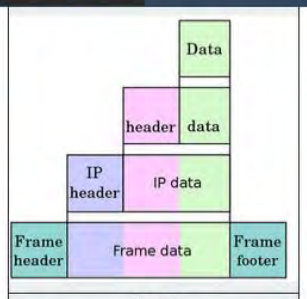
Apps | Filter by: | Maximum Performance Eco-System, Beta Release (v0.1) | Log In



High Speed Packet Capture

Provides line-rate packet capture at bursts of up to 24GB in size. The application configures pairs of DFE

Author: Maxeler Networking



Regex

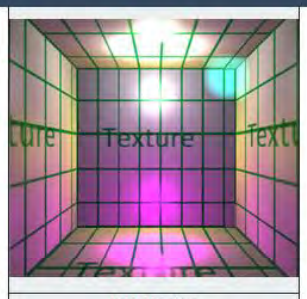
The Regex app takes a regular expression and builds a state machine to implement the regular expression

Author: Maxeler Networking

JPEG Decoder

The JPEG compression algorithm is at its best on photographs and paintings of realistic scenes with smooth variation

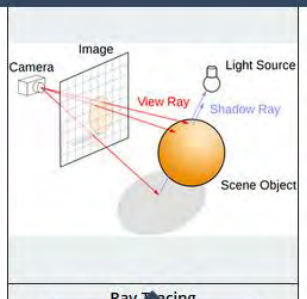
Author: Marko Stupar



Ray Casting

In computer graphics, ray tracing is a technique for generating an image by tracing the path of light through pixels

Author: Marko Stupar

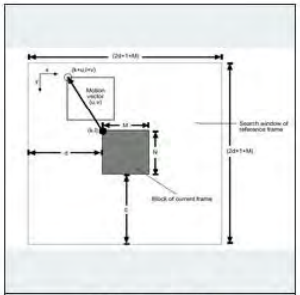


Ray Tracing

Click to see more details!

In ... is a technique for generating an image by tracing the path of light through pixels

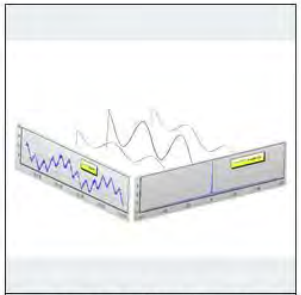
Author: Marko Stupar



Motion Estimation

Motion Estimation is used in video encoding to describe a video frame by motion vectors from other frame

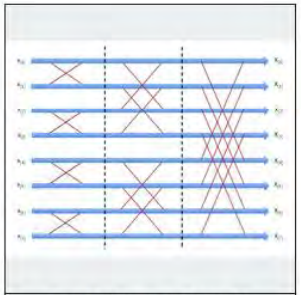
Author: Maxeler Intern



Fast Fourier Transform 2D

This application performs a 2D Fast Fourier Transform on a two dimensional array of complex numbers.

Author: Maxeler London



Fast Fourier Transform 1D

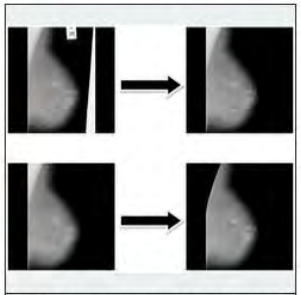
This application performs a one dimensional Fast Fourier Transform on a one dimensional array of compl

Author: Maxeler London

Hybrid Coin Miner

A coin miner application that can mine SHA-256 based coins and Scrypt based coins simultaneously.

Author: Maxeler London



Breast Mammogram ROI Extraction

This App extracts the region of interest from breast mammogram images. Basically, this app removes pecto

Author: Faculty of Engineering, University of Kragu evac, BioIRC Ltd Kragujevac

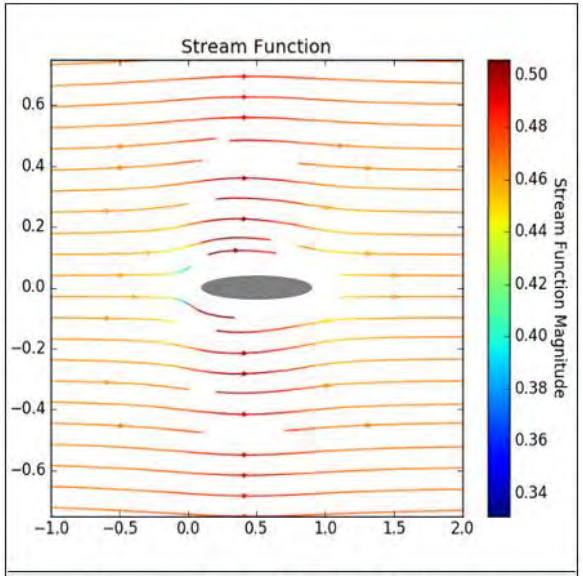
Lz-77

An implementation of Lz77 compression algorithm. Performance: Throughput: 8 input bytes per cycle clock Compressio

Authors: Bisheng Huang ,Xinyu Niu

☆☆☆☆☆

CPU SRC | DFE SRC | GIT | USE | TECH | GUI | VID | ORIG
SPLIT | $ | SAPI | DAPI | MAPI | MAX3 | MAX4 | JDFE



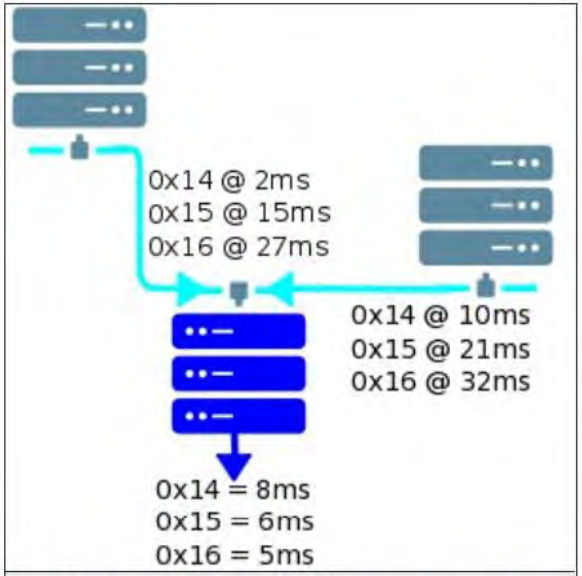
Unstructured Finite Volume

The Airfoil application demonstrates the capability of Maxeler Dataflow Engines to perform Unstructured Me

Author: David Packwood

★★★★★

CPU SRC | DFE SRC | GIT | USE | TECH | GUI | VID | ORIG
SPLIT | $ | SAPI | DAPI | MAPI | MAX3 | MAX4 | JDFE



Network Latency Measurement

An application to measure the latency between two network connections by sending in identical packets an

Author: Kyle Wallpe

☆☆☆☆☆

CPU SRC | DFE SRC | GIT | USE | TECH | GUI | VID | ORIG
SPLIT | $ | SAPI | DAPI | MAPI | MAX3 | MAX4 | JDFE

appgallery.maxeler.com

http://www.mi.sanu.ac.rs/~appgallery.maxeler/

webide.maxeler.com
https://maxeler.mi.sanu.ac.rs

Project: harvard

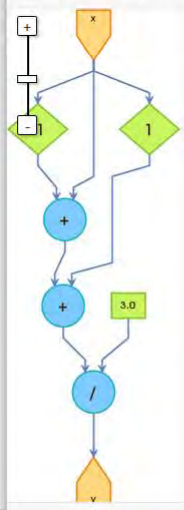
milutinovic

CPUCode/MovingAverageSimpleCpuCode.c

Save Undo Redo Settings

```c
/**
 * Document: MaxCompiler Tutorial (maxcompiler-tutorial)
 * Chapter: 3      Example: 1      Name: Moving Average Simple
 * MaxFile name: MovingAverageSimple
 * Summary:
 *   CPU code for the three point moving average design.
 */
#include "Maxfiles.h"           // Includes .max files
#include <MaxSLiCInterface.h>   // Simple Live CPU interface

float dataIn[8] = { 1, 0, 2, 0, 4, 1, 8, 3 };
float dataOut[8];

int main()
{
    printf("Running DFE\n");
    MovingAverageSimple(8, dataIn, dataOut);

    for (int i = 1; i < 7; i++) // Ignore edge values
        printf("dataOut[%d] = %f\n", i, dataOut[i]);

    return 0;
}
```

Graphs/MovingAverageSimpl

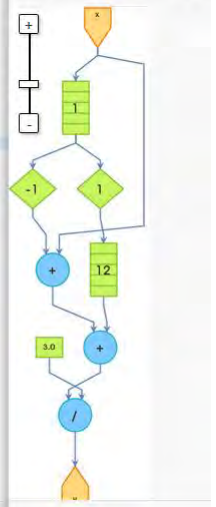


EngineCode/src/movingaveragesimple/MovingAverageSimpleKernel.maxj

Save Undo Redo Settings

```java
/**
 * Document: MaxCompiler Tutorial (maxcompiler-tutorial)
 * Chapter: 3      Example: 1      Name: Moving Average Simple
 * MaxFile name: MovingAverageSimple
 * Summary:
 *  Computes a three point moving average over the input stream
 */
package movingaveragesimple;

import com.maxeler.maxcompiler.v2.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v2.kernelcompiler.KernelParameters;
import com.maxeler.maxcompiler.v2.kernelcompiler.types.base.DFEVar;

class MovingAverageSimpleKernel extends Kernel {

    MovingAverageSimpleKernel(KernelParameters parameters) {
        super(parameters);

        DFEVar x = io.input("x", dfeFloat(8, 24));

        DFEVar prev = stream.offset(x, -1);
        DFEVar next = stream.offset(x, 1);
        DFEVar sum = prev + x + next;
        DFEVar result = sum / 3;

        io.output("y", result, dfeFloat(8, 24));
    }
}
```

Graphs/MovingAverage



webide.maxeler.com
https://maxeler.mi.sanu.ac.rs


MAXELER
Technologies
MAXIMUM PERFORMANCE COMPUTING
HPC
Low latency
Cloud


MAXE
Techno
spective

MultiCore

DualCore?

Which way are the horses going?

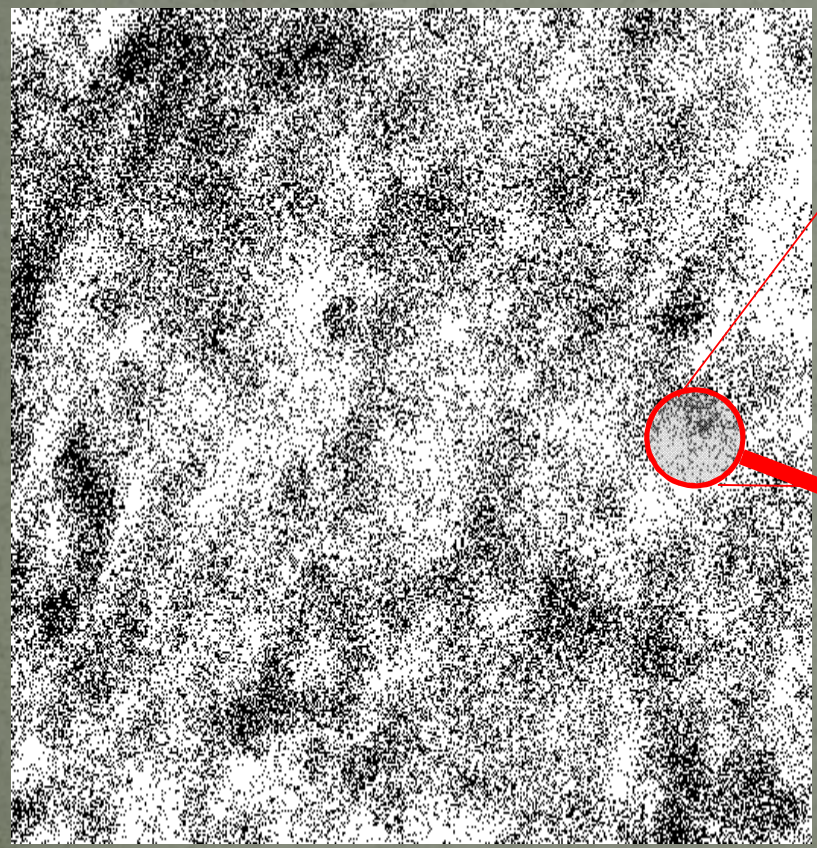
ManyCore

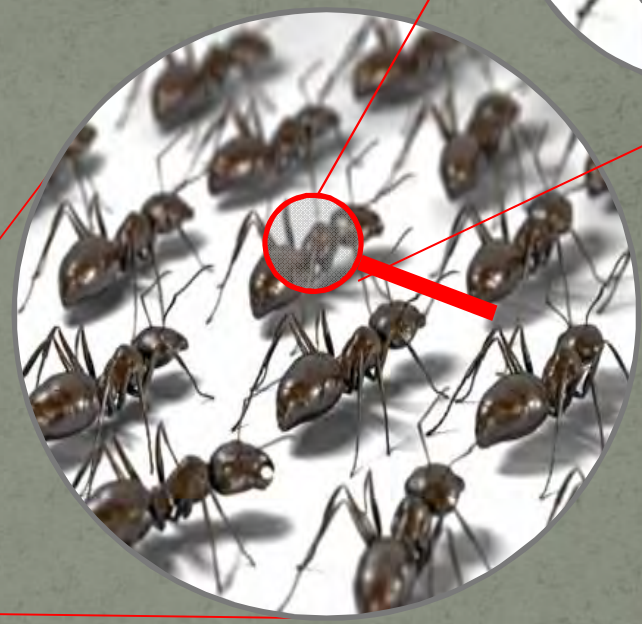
Å Is it possible
to use 2000 chicken instead of two horses?

?
==

ŏ What is better, real and anecdotic?

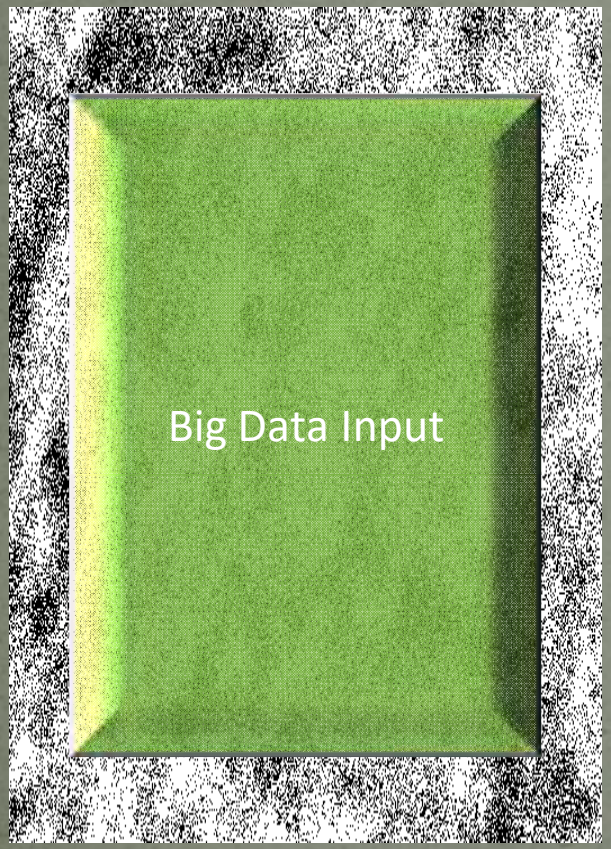
DataFlow





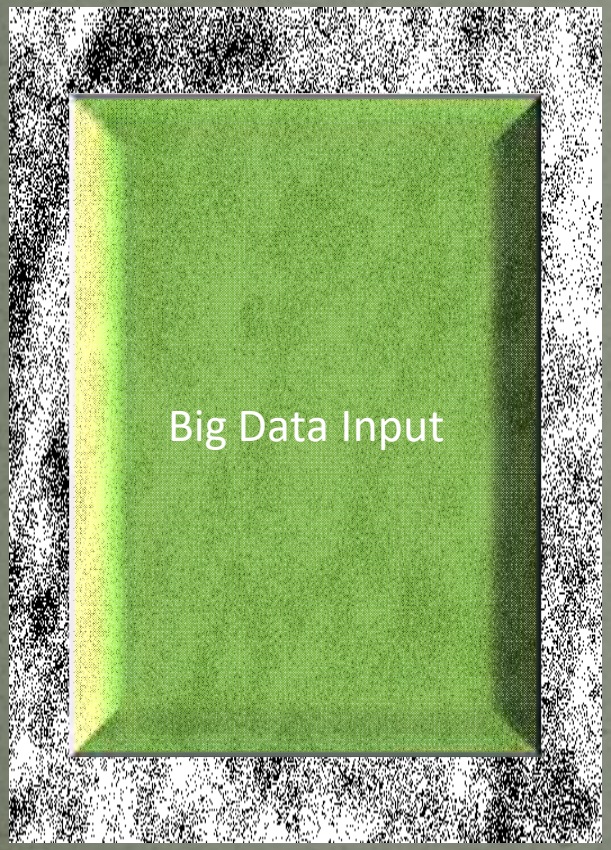
How about 2 000 000 ants?

DataFlow



DataFlow

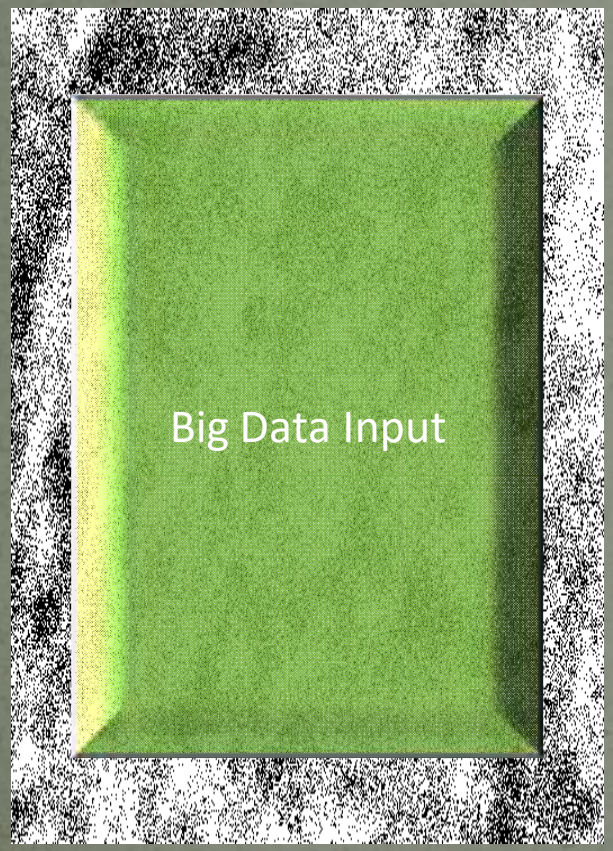
aSoG



DataFlow

aSoG

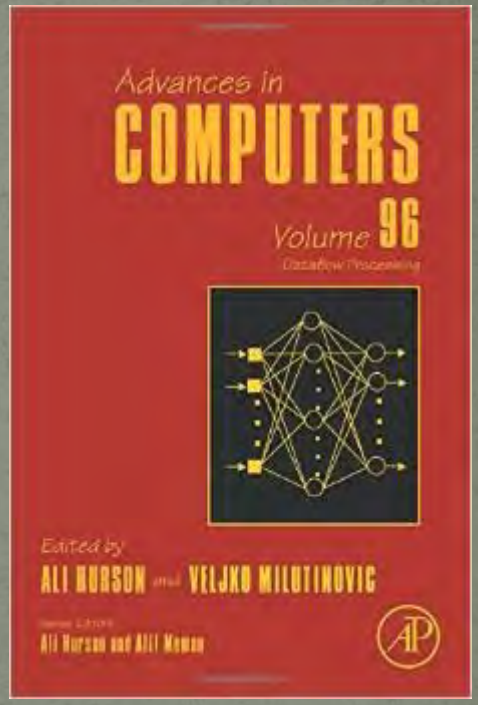
Bronto



Marmelade

An Edited Book Covering the Applications

- http://www.amazon.com/Dataflow-Processing-Volume-Advances-Computers/dp/0128021349
- http://www.elsevier.com/books/dataflow-processing/milutinovic/978-0-12-802134-7

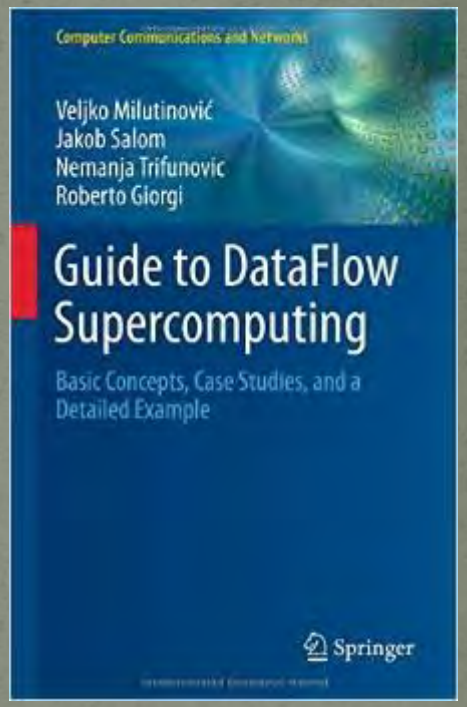


Indexed by: WoS (SCI)

Contributions welcome for the follow-ups: Vol. 102 + Vol. 104 + etcé

An Original Book Covering the Essence

- http://www.amazon.com/Guide-DataFlow-Supercomputing-Concepts-Communications/dp/3319162284
- http://www.springer.com/gp/book/9783319162287



The first source to use the term the Feynman Paradigm in contrast with the Von Neumann Paradigm

NEWS

6/27/2014
10:09 AM

Google I/O: Hello Dataflow, Goodbye MapReduce

Charles Babcock
News

Connect Directly

4 COMMENTS
COMMENT NOW

Google introduces Dataflow to handle streams and batches of big data, replacing MapReduce and challenging other public cloud services.

Google I/O this year was overwhelmingly dominated by consumer technology, the end user interface, and extension of the Android universe into a new class of mobile devices, the computer you wear on your wrist.

At the same time, there were one or two enterprise-scale data handling and cloud computing gems scattered among all the end user announcements.

Hadoop Jobs: 9 Ways To Get Hired

(Click image for larger view and slideshow.)

Alibaba recently did the same!

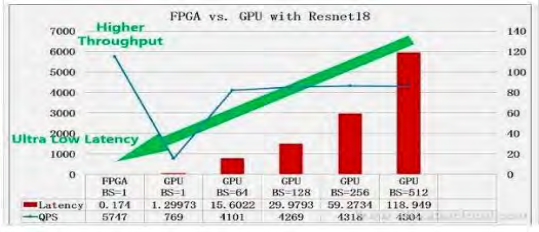
newelectronics
The site for electronic design engineers

Intel says logic is faster than GPUs

News | In Depth | Products & Suppliers | Magazine | Videos | Blogs | Events | About Us

Home > Technology > Article

Alibaba recently claimed the same!



Comment | Share 15 | Tweet 34 | Share 1 | Share 622 | share 994

Intel's Programmable Systems Group takes its first step towards FPGA based system in package portfolio

Speaking in 2012, Danny Biran – then Altera's senior VP for corporate strategy – said he saw a time when the company would be offering 'standard products' – devices featuring an FPGA, with different dice integrated in the package. "It's also possible these devices may integrate customer specific circuits if the business case is good enough," he noted.

Jordan Inkeles, Altera's director of product marketing for high end FPGAs

There was a lot going on behind the scenes then; already, Altera was talking with Intel about using its foundry service to build 'Generation 10' devices, eventually being acquired by Intel in 2015.

Now the first fruit of that work has appeared in the form of Stratix 10 MX. Designed to meet the needs of those developing high end communications systems, the device integrates stacked memory dice alongside an FPGA die, providing users with a memory bandwidth of up to 1Tbyte/s.

28 June 2016

QoL

Maxeler is one of the Top 10 HPC projects
to impact QoL in the World :)

Scientific Computing
[www.scientificcomputing.com/articles/2014/11]

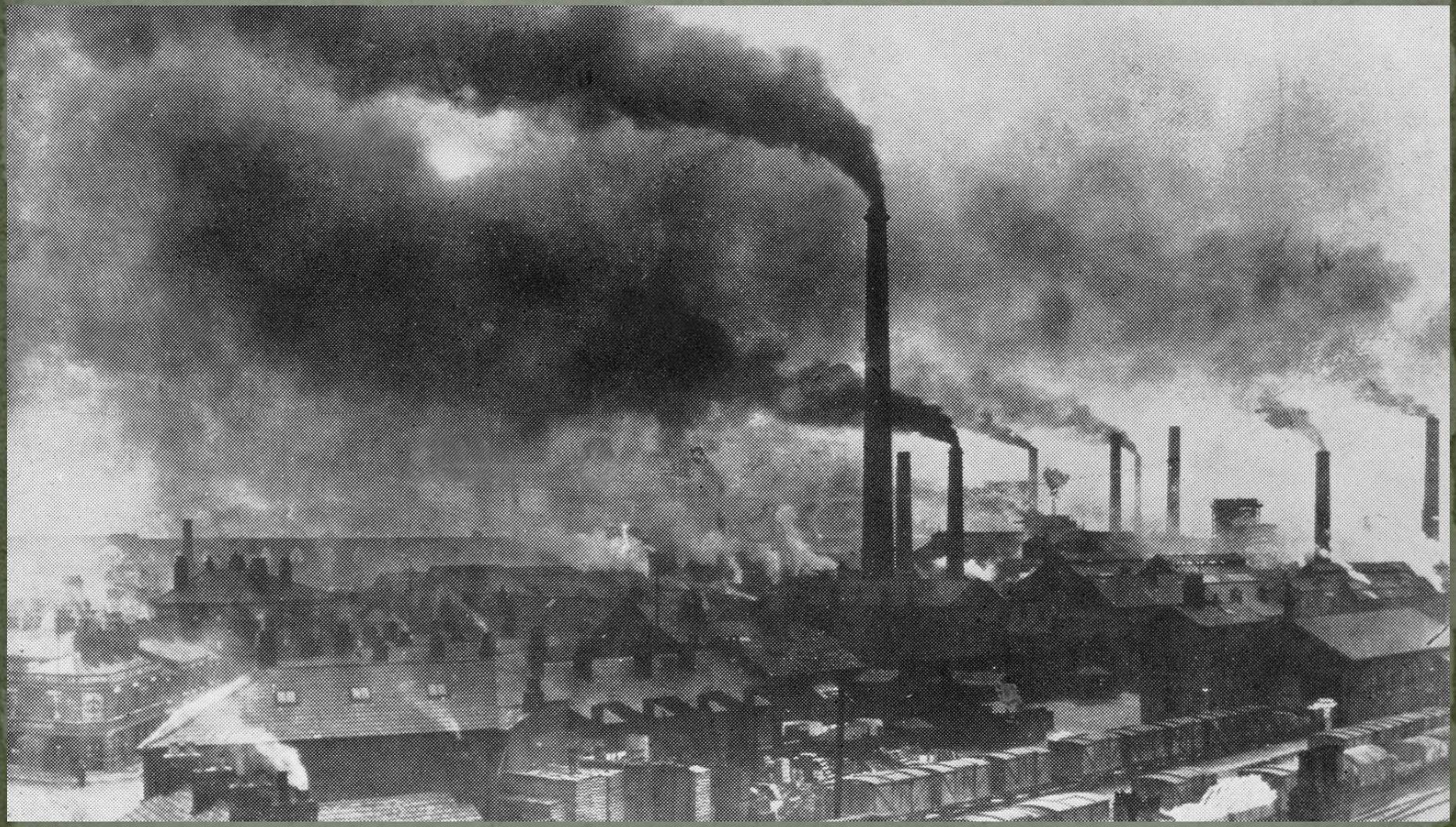
by

Don Johnson

of

Lawrence Livermore National Labs
[editor@ScientificComputing.com]

How About QoL?



DataFlow

SW

HW

AW

Essence of the Paradigm:

For Big Data algorithms
and for the same hardware price as before,
achieving:

a) speed-up, 20-200

b) monthly electricity bills, reduced 20 times

c) size, 20 times smaller

d) precision, X times better

The major issues of engineering are: design cost and design complexity.
Remember, economy has its own rules: production count and market demand!

Why is DataFlow so Much Faster?

Å Factor: 20 to 200

MultiCore/ManyCore

Machine Level Code

DataFlow

Gate Transfer Level

Why are Electricity Bills so Small?

Å Factor: 20

MultiCore/ManyCore

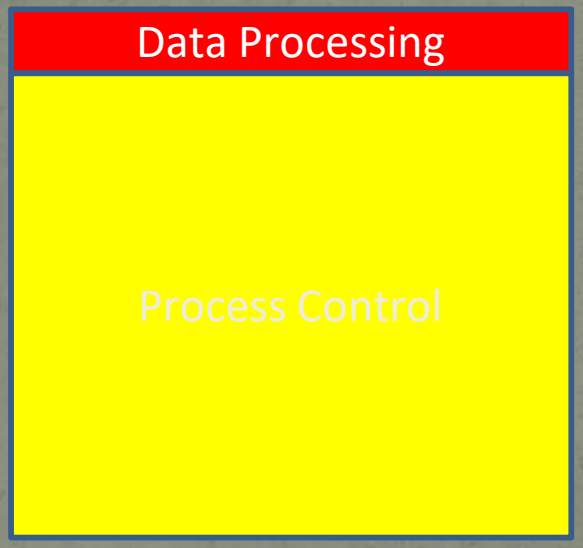
DataFlow

$$P = kfU^2$$

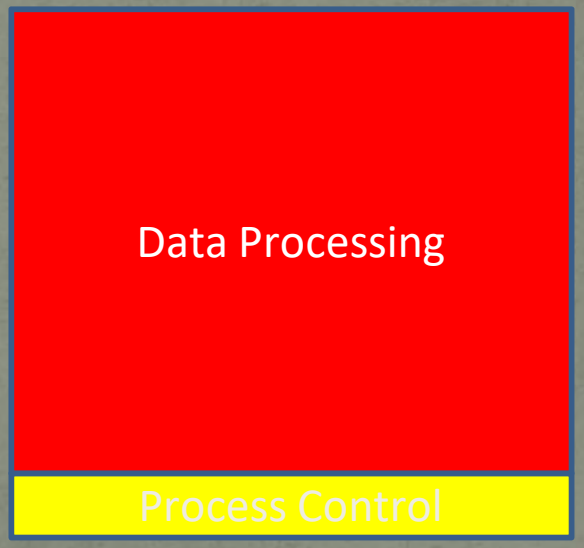
Why is the Cubic Foot so Small?

Å Factor: 20

MultiCore/ManyCore



DataFlow



Why is the Precision Better?

Å Factor: X

M →

N ↓

Bits	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54
18	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
20	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3
22	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3
24	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3
26	2	2	2	2	4	4	4	4	4	4	4	4	4	6	6	6	6	6	6
28	2	2	2	2	4	4	4	4	4	4	4	4	4	6	6	6	6	6	6
30	2	2	2	2	4	4	4	4	4	4	4	4	4	6	6	6	6	6	6
32	2	2	2	2	4	4	4	4	4	4	4	4	4	6	6	6	6	6	6
34	2	2	2	2	4	4	4	4	4	4	4	4	4	6	6	6	6	6	6
36	2	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	6	6	7
38	2	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	6	6	7
40	2	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	6	6	7
42	2	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	6	6	7
44	3	3	3	3	6	6	6	6	6	6	6	6	6	9	9	9	9	9	9
46	3	3	3	3	6	6	6	6	6	6	6	6	6	9	9	9	9	9	9
48	3	3	3	3	6	6	6	6	6	6	6	6	6	9	9	9	9	9	9
50	3	3	3	3	6	6	6	6	6	6	6	6	6	9	9	9	9	9	9
52	3	3	3	3	6	6	6	6	6	6	6	6	6	9	9	9	9	9	9
54	3	4	4	4	6	6	6	6	6	7	7	7	7	9	9	9	9	9	10

Successes of 2020, 2021, and 2022

Hitachi Cloud
Amazon AWS

BQCD
Quark

Endorsed by Jerome Friedman

Special thanks to: Jerome Friedman, Dan Shechtman, Tim Hunt, and Sheldon Glashow

Q&A

vm@etf.rs

US 20180189063A1

(19) **United States**

(12) **Patent Application Publication**

FLEMING et al.

(10) **Pub. No.: US 2018/0189063 A1**

(43) **Pub. Date: Jul. 5, 2018**

(54) **PROCESSORS, METHODS, AND SYSTEMS WITH A CONFIGURABLE SPATIAL ACCELERATOR**

(71) Applicant: **Intel Corporation**, Santa Clara, CA (US)

(72) Inventors: **KERMIN FLEMING**, Hudson, MA (US); **KENT D. GLOSSOP**, Merrimack, NH (US); **SIMON C. STEELY, Jr.**, Hudson, NH (US)

(21) Appl. No.: **15/396,395**

(22) Filed: **Dec. 30, 2016**

Publication Classification

(51) **Int. Cl.**

G06F 9/30 (2006.01)

G06F 13/42 (2006.01)

(52) **U.S. Cl.**

CPC ***G06F 9/3016*** (2013.01); ***G06F 13/4221*** (2013.01)

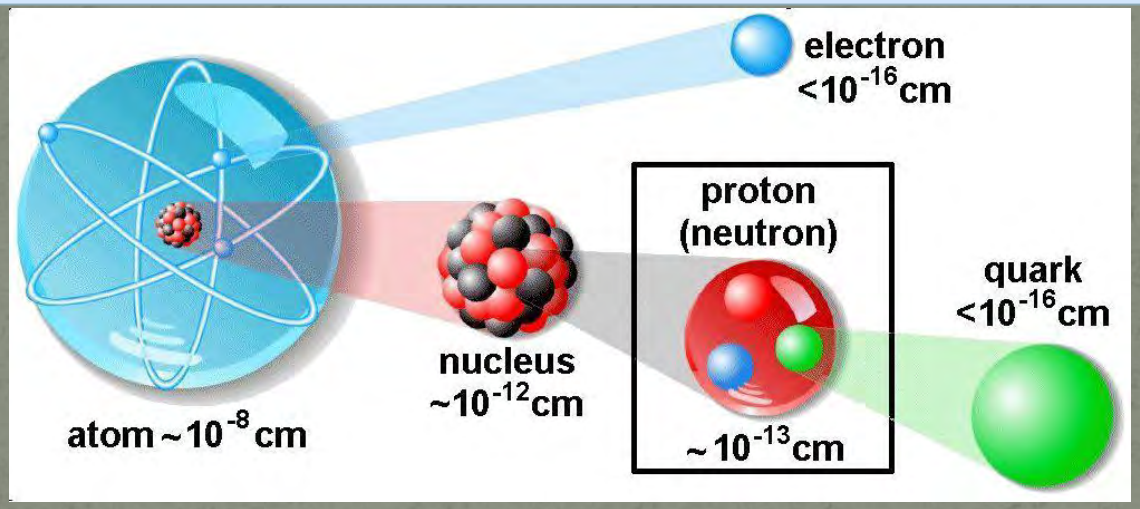
(57) **ABSTRACT**

Systems, methods, and apparatuses relating to a configurable spatial accelerator are described. In one embodiment, a processor includes a core with a decoder to decode an instruction into a decoded instruction and an execution unit to execute the decoded instruction to perform a first operation; a plurality of processing elements; and an interconnect network between the plurality of processing elements to receive an input of a dataflow graph comprising a plurality of nodes, wherein the dataflow graph is to be overlaid into the interconnect network and the plurality of processing elements with each node represented as a dataflow operator in the plurality of processing elements, and the plurality of processing elements are to perform a second operation by a respective, incoming operand set arriving at each of the dataflow operators of the plurality of processing elements.

BQCD on a Maxeler Dataflow Computer

BQCD on a Dataflow Computer

Porting BQCD from BlueGene to a Maxeler Dataflow Computer



- Quantum Chromodynamics (QCD): models interactions of subatomic particles
- Lattice QCD (LQCD): its discretisation, suitable for numerical computation
- Berlin QCD (BQCD): most popular implementation of the LQCD algorithm
- Conjugate Gradient (CG): Majority of the compute time (benchmark: 68%)
  - CG iteratively solves linear algebra problem of form $M\boldsymbol{x} = \boldsymbol{b}$
  - Operator M contains Wilson-dslash and Clover operators
- **PROJECT TARGET 40x speedup of CG part of BQCD, followed by speedup of the entire application by 20x comparing same size boxes Dataflow vs BlueGene/Q**

Maxeler QCD - Deployment

Maxeler QCD solution is deployed at Jülich Supercomputing Center, running on a Maxeler Dataflow system.

	2 racks of Jülich BlueGene/Q machine	On-premise Maxeler Dataflow system: scale to 1PF equivalent	Factor
Volume	6.75 m^3	0.87 m^3	7.76
Overall Time to Solution	1576.60 s	689 s	2.29
Overall Energy to Solution	169.6 kWh	4.42 kWh	**38.4**
Volume x TTS	10,642.05 m^3s	599.43 m^3s	**17.8**

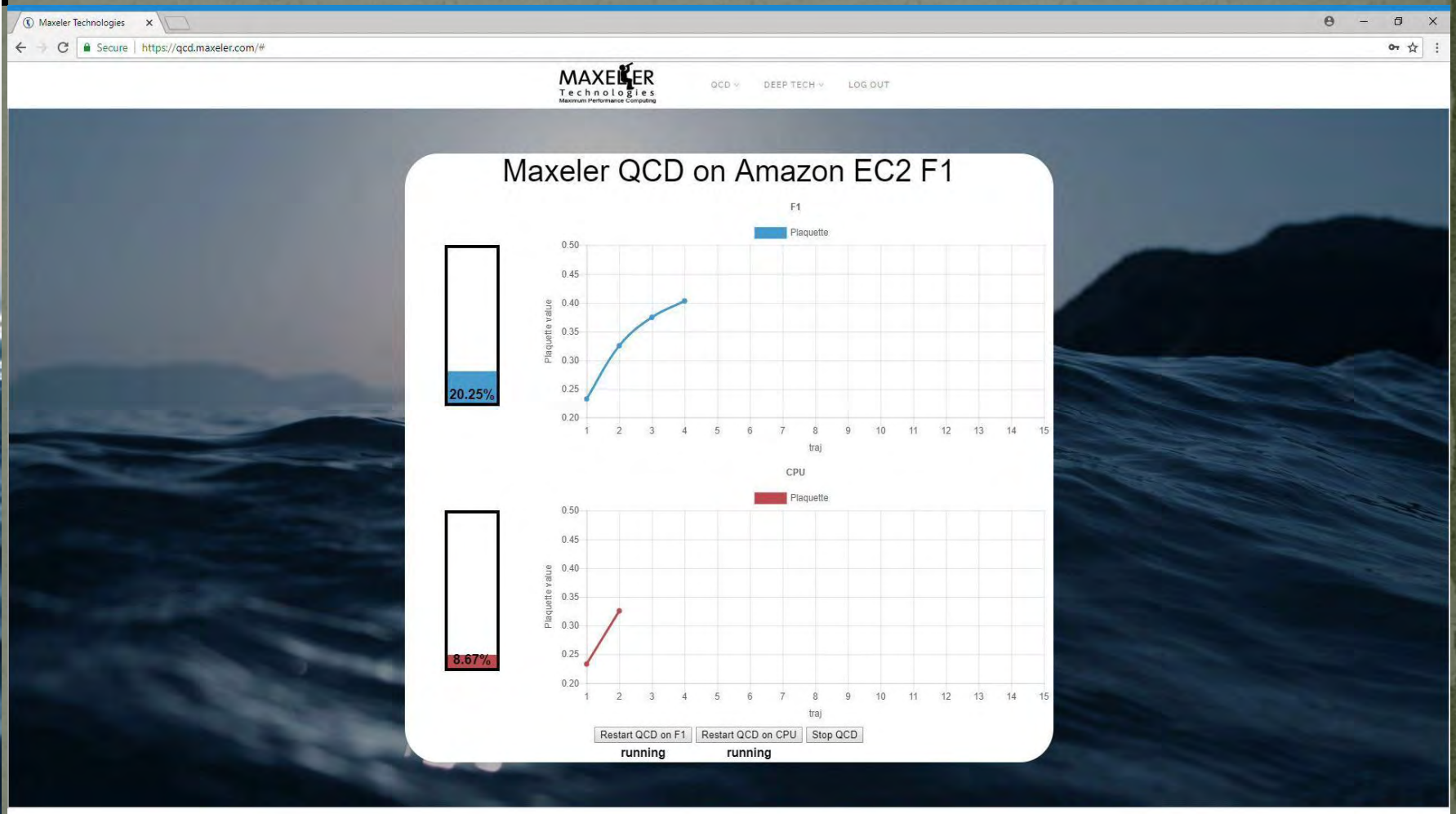
- Evaluate 64x64x64x64 problem, 5 MC steps, 200 HMC steps **Volume x TTS x ETS = 682**

Maxeler QCD on Amazon EC2 F1

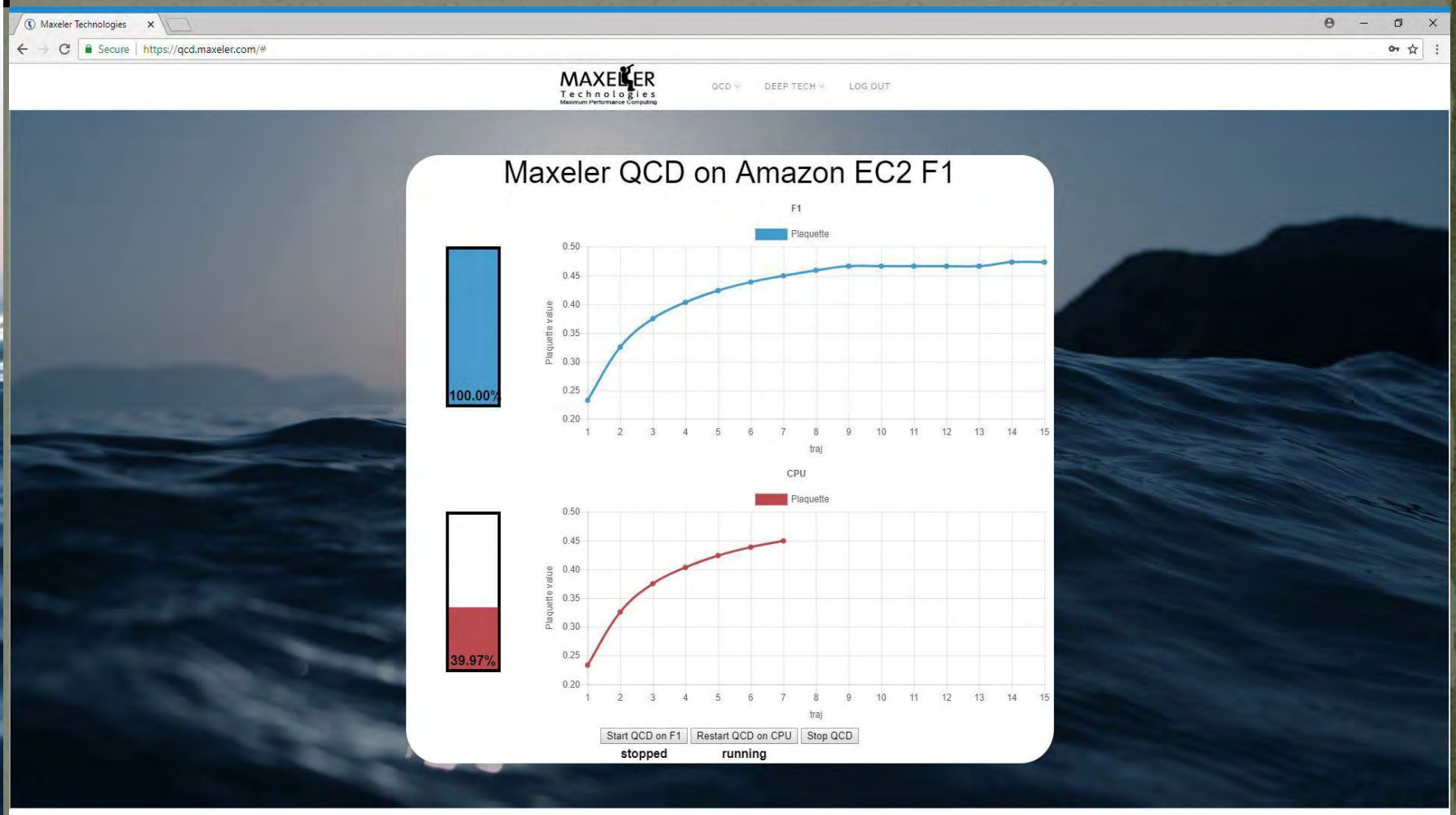
The Maxeler QCD solution is now running on Amazon EC2 F1:
Software portable from Maxeler Dataflow system
to Amazon Cloud

- Elastic computing: expand from on-premise to Cloud
- Scale up computation as workload grows
- Accelerated HPC as a service

QCD Demo


Maxeler Technologies
https://qcd.maxeler.com/#
MAXELER Technologies
QCD
DEEP TECH
LOG OUT
Maxeler QCD on Amazon EC2 F1
F1
Plaquette
Plaquette value
traj
20.25%
CPU
Plaquette
Plaquette value
traj
8.67%
Restart QCD on F1
Restart QCD on CPU
Stop QCD
running
running

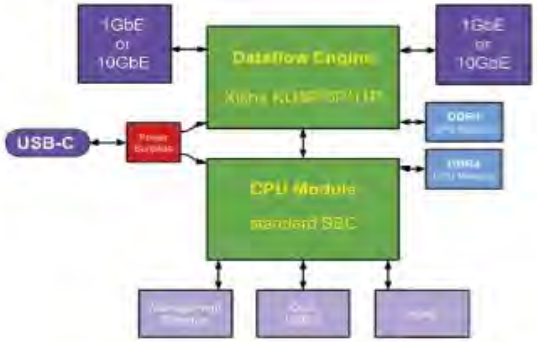
QCD Demo



powered by aws

MAXELER Technologies
Maximum Performance Computing

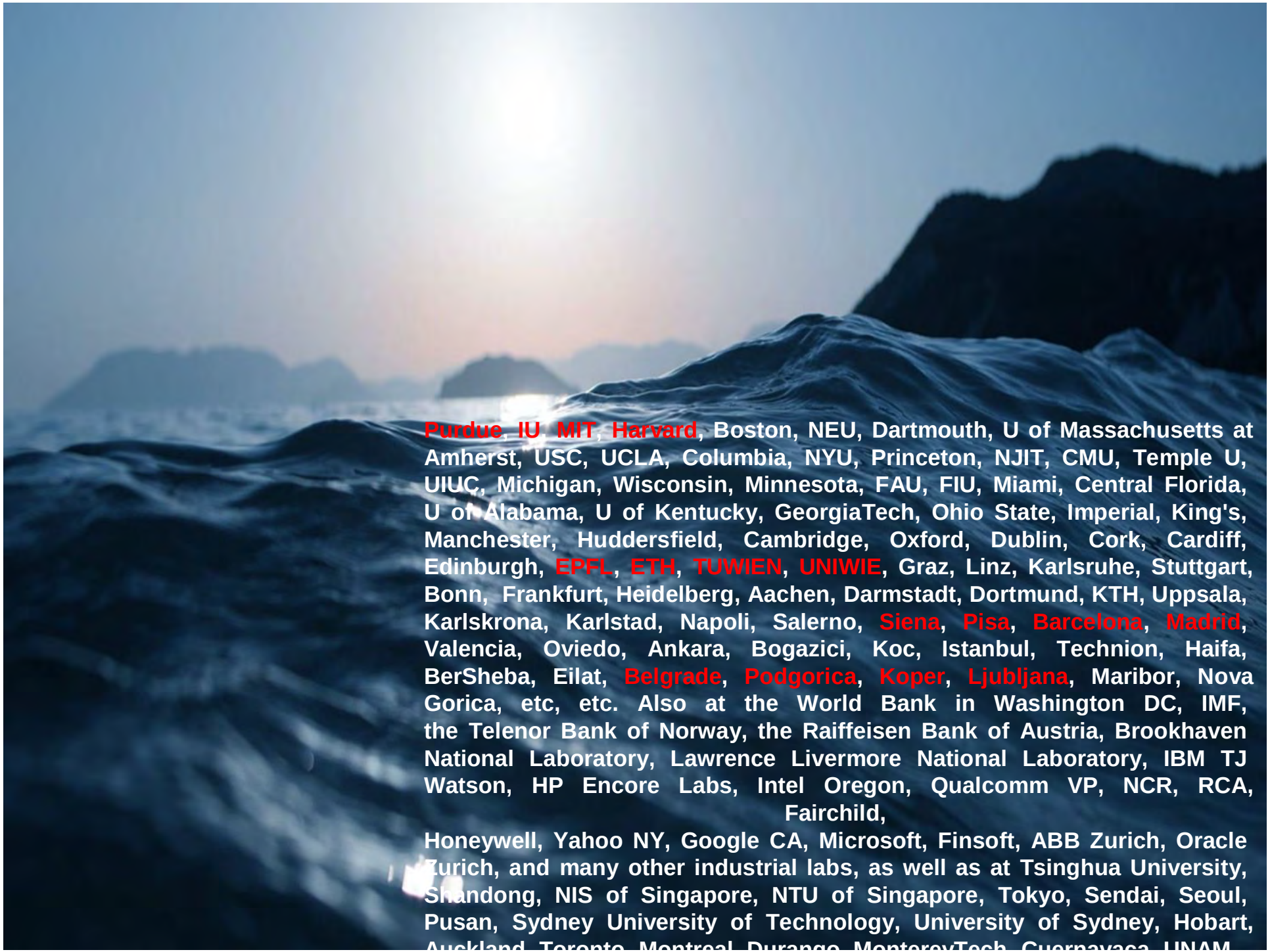
miniMAX5
Edge of IoT Platform



Dimensions	5.7 in (144mm) Wide x 5.7 in Deep (144mm) x 2.2 in High (57mm), excluding power supply		
Form factor	Desktop enclosure, fanless design. Wall or rail mount options available		
Weight	34 oz (950g), excluding power supply		
Power Supply	Separate wall plug unit, providing 60W of USB-PD power from 100-240V, 50-60Hz mains		
Input and Output (Standard Ports)	**Ethernet**	1GbE or 10GbE (copper or fibre)	SFP+ Cage
	USB-C	Power input over USB-PD (Min 15V 3A supply required) USB-3 SuperSpeed II I/O on same connector supporting DisplayPort Alternate mode	
Input and Output (Optional Ports)	**Management LAN**	1GbE	RJ45
	USB	Dual USB-3 Type A ports	
	Video Output	HDMI Type A	
Controlflow Engine	**CPU**	AMD 3rd Generation R- or G-Series - choose from - Quad Core Merlin Falcon RX-416GD @1.6GHz - Dual Core Brown Falcon GX-217I @1.7GHz	Other SBC options available on request
	Memory	2x 4GB DDR4-2133 SODIMM, total 8GB	Higher or lower capacities as required
	Storage	64GB Solid State Memory	
	Operating System	Linux - CentOS 7	Other OS options available on request
Dataflow Engine	**FPGA**	Xilinx Kintex Ultrascale Plus series - choose from - KU5P (217K LUTs, 544 BRAMs, 1,824 DSPs) - KU3P (163K LUTs, 408 BRAMs, 1,368 DSPs) - KU11P (299K LUTs, 680 BRAMs, 2,928 DSPs)	KU5P fitted as standard
	Memory	1x 16GB DDR4-2400 SODIMM	Or 8GB or 32GB

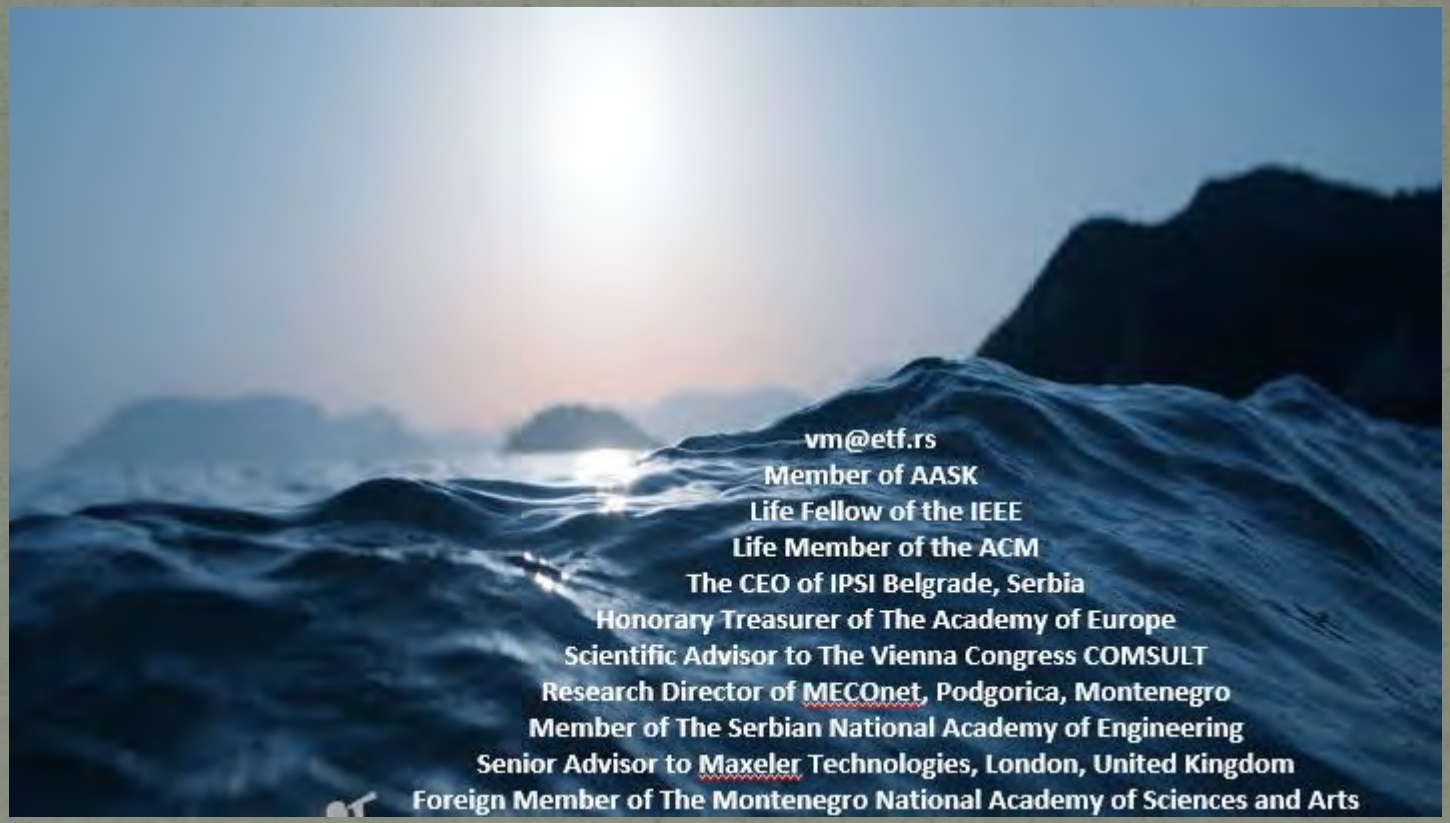
subEdge: microMAX5

www.maxeler.com



Purdue, IU, MIT, Harvard, Boston, NEU, Dartmouth, U of Massachusetts at Amherst, USC, UCLA, Columbia, NYU, Princeton, NJIT, CMU, Temple U, UIUC, Michigan, Wisconsin, Minnesota, FAU, FIU, Miami, Central Florida, U of Alabama, U of Kentucky, GeorgiaTech, Ohio State, Imperial, King's, Manchester, Huddersfield, Cambridge, Oxford, Dublin, Cork, Cardiff, Edinburgh, EPFL, ETH, TUWIEN, UNIWIE, Graz, Linz, Karlsruhe, Stuttgart, Bonn, Frankfurt, Heidelberg, Aachen, Darmstadt, Dortmund, KTH, Uppsala, Karlskrona, Karlstad, Napoli, Salerno, Siena, Pisa, Barcelona, Madrid, Valencia, Oviedo, Ankara, Bogazici, Koc, Istanbul, Technion, Haifa, BerSheba, Eilat, Belgrade, Podgorica, Koper, Ljubljana, Maribor, Nova Gorica, etc, etc. Also at the World Bank in Washington DC, IMF, the Telenor Bank of Norway, the Raiffeisen Bank of Austria, Brookhaven National Laboratory, Lawrence Livermore National Laboratory, IBM TJ Watson, HP Encore Labs, Intel Oregon, Qualcomm VP, NCR, RCA, Fairchild,

Honeywell, Yahoo NY, Google CA, Microsoft, Finsoft, ABB Zurich, Oracle Zurich, and many other industrial labs, as well as at Tsinghua University, Shandong, NIS of Singapore, NTU of Singapore, Tokyo, Sendai, Seoul, Pusan, Sydney University of Technology, University of Sydney, Hobart, Auckland, Toronto, Montreal, Durango, MontereyTech, Cuernavaca, UNAM



vm@etf.rs
Member of AASK
Life Fellow of the IEEE
Life Member of the ACM
The CEO of IPSI Belgrade, Serbia
Honorary Treasurer of The Academy of Europe
Scientific Advisor to The Vienna Congress COMSULT
Research Director of MECOnet, Podgorica, Montenegro
Member of The Serbian National Academy of Engineering
Senior Advisor to Maxeler Technologies, London, United Kingdom
Foreign Member of The Montenegro National Academy of Sciences and Arts