Area-Time-Precision on Demand

The Space-Time-Value Challenges of Reconfigurable Accelerator Design

Georgi Gaydadjiev

Workshop on Reconfigurable Computing January 17, 2024 Munich, Germany



Thinking Vertically about Computing Problems



James Wilkinson on Value+Error

Computation can be described as ideal infinite precision results + error (J. Wilkinson)

OR

as a multiscale discretization of

Space, Time and Value (STV)







Optimizations at all abstraction levels



| Multiple scales of computing | Important features for optimization |
|---------------------------------|--|
| complete system level | \Rightarrow balance compute, storage and IO |
| parallel node level | ⇒ maximize utilization of compute and interconnect |
| microarchitecture level | \Rightarrow minimize data movements |
| arithmetic level | ⇒ tradeoff range, precision and accuracy = discretize in Time, Space and Value |
| bit level | \Rightarrow encode and add redundancy |
| transistor level | => manipulate '0' and '1' |

and more, e.g., trade/hide Communication (Time) for/behind Computation (Space), etc

Easy it is not (and not really new)

Slotnick's law (of effort):

"The parallel approach to computing does require that some original thinking be done about numerical analysis and data management in order to secure efficient use.

In an environment which has represented the absence of the need to think as the highest virtue this is a decided disadvantage."



Daniel Slotnick (1931-1985) Chief Architect of Illiac IV





Depends heavily on what is computed

Imaging: What does it mean for the result to be good enough?



IEEE Floating Point: Bit-accurate IEEE Floating Point, needed?

Accounting: Computing certain exact digits? Decimal? Binary?

Risk: Qualitative feedback might be enough? **1 bit:** will it rain or not?

Optimize representation for arithmetic and data movements

Floating Point

- Vary mantissa & exponent sizes
- Radix-4, radix 16, etc
- Block floating point
- Decimal floating point, etc

Advanced

- Logarithmic numbers
- Modulo Arithmetic (Chinese Remainder Theorem)
- Redundant Numbers

Integer

- Fixed Point
- Dual fixed point

Encode the wave field (STV):

- Predictive coding
- Arithmetic coding
- Lossless vs lossy
 - Wavelets
 - Curvelets, de-noising, etc



Limits on Computing + and × [Shmuel Winograd, 1965]

Bounds on Addition

- Binary: O(log n)
- Residue Number System: O(log 2log α(N))
- Redundant Number System: O(1)

Bounds on Multiplication

- Binary: O(log n)
- Residue Number System: O(log 2log $\beta(N)$)
- Using Tables: O(2[log n/2]+2+[log 2n/2])
- Logarithmic Number System: O(Addition)

However, Binary and Log numbers are easy to compare, others are not!

Also, constant multiplication complexity depends on the number of '1's





From '1's to distance between '1's

Rational Approximations & Continued Fractions

The M-log-Fraction



 $= [0; 2^{M_1}, -(2^{-M_1} + 2^{M_2}), (2^{-M_2} + 2^{M_3}), -(2^{-M_3} + 2^{M_4}), (2^{-M_4} + 2^{M_5})...]$

Oskar Mencer, Rational Arithmetic Units in Computer Systems PhD Thesis, Stanford University, 2000.



Tradeoff compute versus memory



- uniform vs non-uniform
- number of table entries
- how many coefficients

- polynomial or rational approx
- continued fractions
- multi-partite tables

Underlying hardware/technology changes the optimum



in Practice: Tradeoff Representation, Memory and Arithmetic

Minimal Latency (Optimized for Latency)

| 24 | sin: <i>tp2</i> , 24, | 78 | sin: <i>tp2</i> , 28, 348 | sin: <i>tp2</i> , 32, 792 | sin: <i>tp2</i> , 32, 3168 | sin: <i>tp2</i> , 40, 7872 | sin: <i>tp3</i> , 42, 5504 |
|----|-------------------------|-----|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| | log: <i>po2</i> , 8, | 30 | log: <i>tp2</i> , 12, 78 | log: <i>tp2</i> , 15, 384 | log: <i>tp2</i> , 21, 1056 | log: <i>tp2</i> , 24, 4800 | log: <i>tp2</i> , 28,11136 |
| | sqr: <i>tp2</i> , 18, 9 | 912 | sqr: <i>tp2</i> , 24, 2400 | sqr: <i>tp2</i> , 26,10368 | sqr: <i>tp2</i> , 30,23808 | sqr: <i>tp3</i> , 34,17920 | sqr: <i>tp4</i> , 39,12800 |
| 20 | sin: <i>po2</i> , 19, | 61 | sin: <i>tp2</i> , 24, 300 | sin: <i>tp2</i> , 28, 696 | sin: <i>tp2</i> , 32, 3168 | sin: <i>tp2</i> , 32, 6336 | sin: <i>tp3</i> , 38, 4992 |
| | log: <i>po2</i> , 7, | 27 | log: <i>tp2</i> , 12, 78 | log: <i>tp2</i> , 15, 384 | log: <i>tp2</i> , 21, 1056 | log: <i>tp2</i> , 24, 4800 | log: <i>tp2</i> , 27,10752 |
| | sqr: <i>tp2</i> , 18, 4 | 456 | sqr: <i>tp2</i> , 20, 2010 | sqr: <i>tp2</i> , 24, 4800 | sqr: <i>tp2</i> , 32,12288 | sqr: <i>tp3</i> , 33, 8704 | sqr: <i>tp3</i> , 37,19456 |
| 16 | sin: <i>tp2</i> , 16, | 54 | sin: <i>tp2</i> , 19, 240 | sin: <i>tp2</i> , 24, 600 | sin: <i>tp2</i> , 28, 2784 | sin: <i>tp2</i> , 32, 6336 | sin: <i>tp3</i> , 32, 4224 |
| | log: <i>po2</i> , 7, | 27 | log: <i>tp2</i> , 12, 78 | log: <i>tp2</i> , 15, 384 | log: <i>tp2</i> , 21, 1056 | log: <i>tp2</i> , 24, 4800 | log: <i>tp2</i> , 27,10752 |
| | sqr: <i>tp2</i> , 17, 3 | 324 | sqr: <i>tp2</i> , 18, 912 | sqr: <i>tp2</i> , 24, 2400 | sqr: <i>tp2</i> , 26,10368 | sqr: <i>tp2</i> , 30,23808 | sqr: <i>tp3</i> , 34,17920 |
| 12 | sin: <i>po2</i> , 9, | 31 | sin: <i>tp2</i> , 16, 204 | sin: <i>tp2</i> , 20, 504 | sin: <i>tp2</i> , 24, 2400 | sin: <i>tp2</i> , 28, 5568 | sin: <i>tp2</i> , 32,12672 |
| | log: <i>po2</i> , 7, | 27 | log: <i>tp2</i> , 12, 78 | log: <i>tp2</i> , 15, 384 | log: <i>tp2</i> , 21, 1056 | log: <i>tp2</i> , 24, 4800 | log: <i>tp2</i> , 27,10752 |
| | sqr: <i>tp2</i> , 12, | 156 | sqr: <i>tp2</i> , 18, 450 | sqr: <i>tp2</i> , 20, 2016 | sqr: <i>tp2</i> , 24, 4800 | sqr: <i>tp2</i> , 31,12288 | sqr: <i>tp3</i> , 33, 8704 |
| 8 | sin: <i>po2</i> , 6, | 22 | sin: <i>tp2</i> , 10, 132 | sin: <i>tp2</i> , 16, 408 | sin: <i>tp2</i> , 19, 1920 | sin: <i>tp2</i> , 24, 4800 | sin: <i>tp2</i> , 28,11136 |
| | log: <i>po2</i> , 7, | 27 | log: <i>tp2</i> , 11, 72 | log: <i>tp2</i> , 15, 384 | log: <i>tp2</i> , 21, 1056 | log: <i>tp2</i> , 24, 4800 | log: <i>tp2</i> , 27,10752 |
| | sqr: <i>tp2</i> , 7, | 27 | sqr: <i>tp2</i> , 17, 324 | sqr: <i>tp2</i> , 18, 912 | sqr: <i>tp2</i> , 24, 2400 | sqr: <i>tp2</i> , 26,10368 | sqr: <i>tp2</i> , 30,23808 |
| 4 | sin: <i>po2</i> , 6, | 22 | sin: <i>tp2</i> , 10, 132 | sin: <i>tp2</i> , 15, 384 | sin: <i>tp2</i> , 19, 1920 | sin: <i>tp2</i> , 23, 4608 | sin: <i>tp2</i> , 28,11136 |
| | log: <i>po2</i> , 7, | 27 | log: <i>tp2</i> , 11, 72 | log: <i>tp2</i> , 15, 384 | log: <i>tp2</i> , 17, 1056 | log: <i>tp2</i> , 24, 4800 | log: <i>tp2</i> , 27,10752 |
| | sqr: <i>po2</i> , 7, | 25 | sqr: <i>tp2</i> , 12, 150 | sqr: <i>tp2</i> , 18, 456 | sqr: <i>tp2</i> , 20, 2016 | sqr: <i>tp2</i> , 24, 4800 | sqr: <i>tp2</i> , 31,12288 |
| | 4 | | 8 | 12 | 16 | 20 | 24 |

Precision [bits]

Dong-U Lee, et.al., Optimizing Hardware Function Evaluation, *IEEE Transactions on Computers*. vol. 54, no. 12, pp. 1520-1531. Dec, 2005

Range [bits]

Munich, Jan 17 2024

Next: Minimize '1's => Sparse Coefficients







Nicolas Brisebarre, Jean-Michel Muller and Arnaud Tisserand Sparse Coefficient Polynomial Approximations for Hardware Implementation, Asilomar Conference, 2004.



Coefficients, Coefficients, FD Coefficients...

3D Finite Difference Coefficients



Time to compute is consequence of distance of coefficients in memory



Local temporal parallelism => Cascading timesteps

MUDelft Munich, Jan 17 2024

Motivation for Elementary Functions

- Used by compute intensive applications
- Evaluating on CPU
 - Many cycles in software or CPU microcode
- Evaluating on Reconfigurable HW (FPGA):
 - Evaluation stages add latency
 - Resources reduce to required precision and bit width
- Function composition at the cost of just one function
 - **Similar costs** in terms of hardware resources for approximating the value at a given x for expressions like $f(x) = log(1 + exp(-x^2/2))$ and g(x) = exp(x).



Resource utilisation: floating point¹

| | LUT | FF | BRAM | DSP | LUT | FF | BRAM | DSP |
|----------------|----------------|-------|------|-----|----------|---------|------|-----|
| | dfeFloat(8,24) | | | | dfeFloat | (11,53) | | |
| multiplication | 155 | 364 | 2 | 1 | 343 | 696 | 3 | 4 |
| addition | 582 | 657 | 4 | 0 | 1,025 | 1,307 | 2 | 0 |
| division | 3,302 | 3,188 | 10 | 0 | 9,713 | 7,881 | 24 | 0 |
| sqrt | 470 | 897 | 1 | 0 | 1,741 | 3,356 | 1 | 0 |
| sin | 679 | 1,082 | 7 | 4 | 2,053 | 3,928 | 28 | 16 |
| COS | 693 | 1,072 | 7 | 4 | 2,082 | 3,908 | 28 | 16 |
| ехр | 781 | 1,201 | 3 | 5 | 2,495 | 3,759 | 6 | 22 |
| pow2 | 684 | 961 | 3 | 3 | 2,097 | 3,131 | 6 | 14 |
| log2 | 541 | 916 | 5 | 4 | 1,533 | 3,163 | 26 | 16 |

¹ Maia DFE, pipelining 1.0. Numbers may differ for each compilation. Sampled with MaxCompiler 2016.1.1





15

Resource utilisation: fixed point²

| | LUT | FF | BRAM | DSP | LUT | FF | BRAM | DSP |
|----------------|----------|------------|---------|--------|-----------|-----------|---------|-------|
| | dfeFix(1 | .6,16, TW0 | OSCOMPL | EMENT) | dfeFix(32 | ,32, TWOS | SCOMPLE | MENT) |
| multiplication | 36 | 33 | 0 | 2 | 193 | 536 | 0 | 8 |
| addition | 16 | 17 | 0 | 0 | 32 | 33 | 0 | 0 |
| division | 1210 | 2,378 | 0 | 0 | 4,786 | 8,509 | 19 | 0 |
| sqrt | 347 | 352 | 0 | 0 | 1,271 | 1,275 | 0 | 0 |
| sin | 349 | 431 | 4 | 8 | 1,255 | 2,909 | 26 | 26 |
| COS | 365 | 448 | 4 | 8 | 1,287 | 2,947 | 26 | 26 |
| exp | 1053 | 1,485 | 4 | 8 | 1,699 | 2,920 | 5 | 16 |
| pow2 | 904 | 1,198 | 1 | 4 | 1,322 | 1,868 | 1 | 7 |
| log | 248 | 317 | 2 | 5 | 659 | 1,193 | 4 | 14 |

² Maia DFE, pipelining 1.0. Numbers may differ for each compilation. Sampled with MaxCompiler 2016.1.1





16

What about custom functions

- Function value is *approximated* at a given point
 - vast literature on function approximations in hardware
 - libraries: CPU (e.g. fdlibm), FPGA (e.g. FloPoCo), ...
 - multi-precision approximations: Maple, Mathematica, etc.
- Options in hardware:
 - Simple lookup table
 - Iterative methods (e.g., Newton-Raphson)
 - Approximations on one interval [a,b]
 - Piece-wise approximations on many subintervals
 - Combination of lookup tables and shifts
 - Various combinations of all above



Small lookup table example

Problem: implement f(x) = sin(x) for 12 bit fixed point x

- There are 2¹² = 4,096 function values in total
- Each value is only 12 bit wide
- hardware implementation as a lookup into FMEM
 Cost: no more than 4 BRAMs

A bit of CPU work:

- tabulate the function on CPU
- define mapped ROM





Iterative methods

- E.g., Newton-Raphson $\varphi(x) = 0 \implies x_{n+1} = x_n \frac{\varphi(x_n)}{\varphi'(x_n)}$
- Works well when rhs simplifies to a polynomial
- Example: evaluate $f(x) = \frac{1}{\sqrt{x}}$ at x = a. Choose $\varphi(x) = \frac{1}{x^2} - a \implies x_{n+1} = \frac{x_n}{2} \left(3 - ax_n^2\right)$
- Notes
 - Needs differentiable $\varphi(x)$, converges to a local minimum
 - sensitive to initial guess
 - quadratic convergence: precision roughly doubles
 => you can start iterations in small bit width



Approximations on one interval

- 3 steps to compute f(x)
 - Step 1: <u>Argument Reduction</u> = g(x) (bare for the next slide)
 - Step 2: <u>Approximation of g(x)</u> over interval [a,b]
 - 1. Lookup Table for a small number of bits
 - 2. Lookup Table + Add/Sub => Bi-partite tables
 - 3. Lookup Table + Mult-Add => Piecewise Linear Approx
 - 4. Shift-and-Add Methods => e.g., CORDIC
 - 5. Polynomial and Rational Approximations
 - 6. Almost never use Taylor series: converges slowly!
 - Step 3: <u>Reconstruction</u> to original argument (if necessary)



Simple argument reduction

- Function is periodic: can shift x towards the origin
- Example: sin(float x)

```
float sin(float x) {
   float y = x mod (π/2); // argument reduction
   float r1 = c0*y*y+c1*y+c2;
   float r2 = c3*y*y+c4*y+c5;
   return (r1/r2); // rational approximation
}
```

- c₀-c₅ are coefficients of a rational approximation of sin(x) in [0, π/2]
- How to generate coefficients c₀-c₅. Use computer algebra system: Wolfram alpha (Mathematica), Maple,...



More complicated argument reduction

- Function $y = \exp(x)$. Reducing x to r in [-ln(2)/2, +ln(2)/2]:
 - Find integer N such that $r := (x N^*ln(2))/2$ is in the interval
 - Equivalently, $x = N (0.5 \ln 2) + r$
 - Using identities: $exp(x) := 2^{0.5N} exp(r)$
- Step 1:
 - N := integer quotient of $x/(0.5 \ln 2)$. Adjust N to make it even!
 - calculate r as accurate as you can
- Step 2:
 - Compute exp(r) by approximation (e.g. polynomial)
 - Inaccurate r yields inaccurate exp(r)...
- Step 3:
 - Compute $exp(x) = 2^{0.5N} exp(r) = 2^k exp(r)$ -- just a shift! If N=2k



Evaluating Polynomials $f(x) \approx \dots + c_3 x^3 + c_2 x^2 + c_1 x + c_0$ $= (((\dots + c_3)x + c_2) \cdot x + c_1) \cdot x + c_0$

- Horner Rule transforms polynomial into a "Multiply-Add Structure"
- Multiply-Add is more numerically stable
- Multiply-Add takes less HW resources than multiply and add as 2 separate operations



Piece-wise approximations

- Many approximations locally defined on their subintervals [ai, bi].
- Approximations only differ by e.g., polynomial coefficients
- For every x find its interval
- Table lookup: get coefficients for this interval
- Evaluate e.g., polynomial
- Does not hurt to employ argument reduction: less intervals, higher convergence in each interval
- How to generate: use compute algebra system. Remez method (minimax polynomial), splines...



Further Reading on Function Evaluation

- J.M. Muller, "Elementary Functions," Birkhaeuser, Boston, 1997.
- Story, S. and Tang, P.T.P., "New algorithms for improved transcendental functions on IA-64," in Proceedings of 14th IEEE symposium on computer arithmetic, IEEE Computer Society Press, 1999.
- D.E. Knuth, "The Art of Computer Programming", Vol 2, Seminumerical Algorithms, Addison-Wesley, Reading, Mass., 1969.
- C.T. Fike, "Computer evaluation of mathematical functions," Englewood Cliffs, N.J., Prentice-Hall, 1968.
- L.A. Lyusternik, "Handbook for computing elementary functions", available in English translation.



Euclids Elements, Representing a²+b²=c² => optimal representation is important

and have a series of the serie Low of and and a second of degrade damagent ale agreed and allows. Again to read about much and the second provided by a second of the second s

top maria Diare marting to which a gray of and ages anvery here here any 1 the part more and the Laborguard and all a vy and the armondy Didgeon The wood for a diamon The hat ready a painty of and Brown of 1871 aperes and aligney of The in my program for any program a so happy for it in a in and and the for the second

the year and many and ing and that I the fun 2 No worward in mayor are swith the frike mane, I tore the gant ria

mould mean for a construction of the states year, minetery at method whohologs A which least is open dies - im record he monthly high's place off-to monthy and A to bay when I a oh of the love any I mp u as a your fip of a part of the vorpation apanaparties ho up They whop Joad by door Himp and any so in her on he and The and a welling - I what the way

pappheos hy oco hat le is both the to capagad zy hay hog op on Loobrach Decempertain bigradan J of tina leaser poor in proving his h

ugar - janiai duo brohaj ijay = a h p h hop-ra a or a poph led uphal - I was pre- po hahira main ha oponio jono co escual p to arog anapadily to var hist have air ashi hajkaar hi ard bi !!

to a rogao hay tog jo htyp h maya Lichtarly Die Same + hy a s 6 has 6 parle or phope horoa hora av of haras in change by hi voo zav dy /4 John way too of joh by to had a har this

ageh die zw - this as his his his " as horrare - 210 my Joalde 14 6m A of 6 par 6 have 6 pan hay my ins h von have rapian - the from sur jake nao lo agartich ano frhizy johih real-lower the Kuchah - Low 3rd I bit when a the lash protection by ex and other hand hand a land and a land a about pie & hater ob a a a later a I present pres & anoth I pres of present

Tala milaio, dal a abat pt 010 - Lot = BY YY. 1879- 2ad. Ib. A when y har tan lah la an ille hand and hip had man as 106 hartones - ap in ovay ish the up the der he good to the good tot the good to the good to the go has designation and a solar type a - ala say the I a fil wh jouphy of a far jou all how of 1 joop aparty pleaser out ma batty who hat the on I as present to bake non open on the 22 A haby raching abrah by chopas leasto yt mabally to Labrob Jush I wi ar les apor of sharastery by they apop & valu- lors , was and phat where is a shall to got by the long on a por a first and ieworthou grapay per pairie to and the second for the second of the

I wind I bernonia Insterior 2 mar to A ביאדייד שד עצד ורי באומושיר אווודי לברים and and the second and the " The production a will now high his mating the start of the Bur , where any minister we have not any and at a strong the love when A is the ment Down A lo . Wi may us you the The the of D. Mayan, J' T. 3485 The To the par fall it in the that the rear rul the KG

Workand ATTO store the delter is me of her force it have him work Bainch non the ay after party groups, Loop of 10 - Toro - a too - loop and a d at y back - 1 th a maria . you be Loronob parabrara - 161 Ambera. To due Thorphice they a prap ismo - 1 quarohor o Japas Labor whoh' gob of h' - sie on ma-jopp hoop - the - a bpi & ou a what y ban the horas o who the had ag :- ,

The man frain - have a sta to to some I for the the former

insmilt -5-02 W KF. int F TIBALT . THE ALT a mostile rev.

Gand bod whood o wwenters - Lating to but I & bar when least place and Jackton mak Loal bit whong no at a pop of Gray whom in web. frender yours Transfar for = and lontor han your to beach ob-0 ppl /h. - 1 bi & whon & + biles and it agrooking it part have pair 6 pay mpan jo up by at join

an a lichter of a low low water a low low low have poro to y may applicately hims and handrey prom had un use as hugov - heav brogarmpas op-our that level le fort on - in un johhad bis toto hy on hilly shing joh app had of his in jo or the la retain the game of the party of non-low on o - have an it bad a por le aspos apoo lego por gorano Thoay of hear who has a se and white it is white when I day App - pointing apiariag 1 That whose a stand ore book and I why we was it was why have give may hours age to y is no fy in you A . V. Mabras Jury pares waray jo aphilip parameri he as noped tall a billing barrament

ing toutroy have in the in the oken is advint as the the the start of a pro-

Base ination & Ast mare use wood gail and a mouth if is tered wage him

y - 14 b & who h lash of the 10 time of her any - for my my more hay we have a grant in y as It joh ling ton h valagip it harry he aw bough Ab Warraush Lainy des Johns way hold main molo ha H Hy johr -piasapas play - the yes a way - jo h spot

is they be dere to reach and a set in the set of the se land leafer grib and about the set I as the section of any section of the section of t

+ realized and tint Incit Talmen 1 icharder grodore at To Tolarro Take Tou Trong Solit + 2 - + with Trateriver ment mit はいいい ションシャンテレー ディン that ning the Total ATT WITH TRACTIVE No tread of the rest in rentwe and a wax of the xol mar mile x m - zache in the sel remain + A" EWO TO ARE THEY L UP Hastad at Tony the million is the me These meguine towers in ביבישואו ד שנולאיתי או inter all'a grant a'r in the with the the 2 mg al my b 17 - 15 - 5 11 The Fat at a Torres TT ICA TO SUR WIT GRAND SH Colgens Highman . Fairing le mainer hand sho ber wrage a The a war war and an

おんちょういんかうない あってん いち テ ちょうかん うち 7.5 6 g 7 2 7 6 a. . 11 agent Tumory 5 a

23-144-14-14-14

Maximum Performance Computing => 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)

Optimal Representation is a hard problem ontop of a hard problem.

*Kolmogorov, A.N. (1965). <u>"Three Approaches to the Quantitative Definition of Information"</u>. *Problems Inform. Transmission* **1** (1): 1–7. **TUDEIft** Munich, Jan 17 2024

Comparing an x86 based 1U machine with a Multiscale Dataflow based 1U machine with 8 DFEs



Modelling 25x



Finite Difference 60x



Data Correlation 22x

| Alignment Type | Query : UniRefSO_F2T2I7 Histone-lysine N-methyltransferase n=8 Tax=E | (1280) | |
|--|---|---|--|
| Query parter50.Fasta.opt Number of queries 174 Mix length 120 Mix length 120 | Best cores : 00100700 STL_0CCM Histone-lysine k-activit ransfranse, HS 00100700 STL_0CCM Histone-lysine k-activit ransfranse, HS 00100700 STL_DCCM Histone-lysine k-activit ransfranse, HS 00100700 STL_DCCM Histone-lysine k-activit ransfranse, HS 00100700 STL_DCCM Histone-lysine k-activit ransfranse, HS 0010070 STL_DCCM Histone-lysine k-activit ransfranse, HS 0010070 STL_DCCM Histone-lysine k-activit ransfranse, HS 0010070 STL_DCCM Histone-lysine k-activit ransfranse, HS 0010071 STL_DCM HIStone-lysine k-activit ransfranse, HS 0010071 STL_DCM HIStone-lysine k-activit ransfranse, HS | (Length) SV (1271) 407 (1229) 384 (1220) 384 (1220) 384 (1220) 384 (1220) 384 (1220) 384 (1231) 229 (1252) 215 (1076) 206 (1080) 998 (1040) 899 | 17 19 18 13 19 50 19 50 50 50 50 50 50 50 50 50 50 50 50 50 |
| Database | Best alignment : | | |
| Number of sequences : 532224 Number of residues : 188726448 | NSRAS KARADEEPTAPSALOKKIRS SKAKOREPKOLIKELOOPOSSIEARTAAT KAN TYTON NSRAPKAT ADITEPTAPSALOKKIRS - KAKOREHINA NT EKAKOPLENLOL SS - T | 779GAEEGGASONN PDIK-GGVG | ITNS TSA |
| Scoring matrix : | - HPWAVCE-R-SAE-T | SASA-P-POVAKP | NGI TYD |
| BLOSUM62 | LERKR. THEKREKPOYEVFOTTED-EARPADERIAIANVTRGA/CKOKTKVRPAPYILRP LORKP-SKAPBERDVETP/CVCCKDPRCDRPALANVTRGA/CKOKTKVRPIPYILRP | PYDPATSVGPOPP VAYDPTTSVGPOPP | TOT |
| Onen Can Banaltu | TOYDPLTPLAP ISALP SSFCDIAEIX NRIDPNTORFLOW/SIRVKOSRM/ROGOPLI AADA TOFDPLTPIAAISALFSSFCDIGEDNRIDPMTORFLOW/SIRVKOSRAFRODISLSASDA | ARAYLECKKEORI ARAYLECKKEORI | COVR COTR |
| open dap remary =3 | NELDRIWYSCHWARL ITADKAEFPSLEESKESVCDONRLPICOGAO DREDSKORL IPEGPRATHKOPAPSLEETPILD DIXROPYIFIAHCYPVLSTITPHLERRLXLFNHK | PSTAPKCPSCRSS | il HP |
| Stop Compute | Performance: 812.0759 GCUPS | WROCKIGYYIIFE | NSR. |



Fluid Flow 30x



Imaging 29x



Weather and climate models on DFEs



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

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

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







Weather models precision comparison



30

What about 15 days of simulation?



Surface pressure after **15 days** of simulation for the double precision and the reduced precision simulations (quality of the simulation hardly reduced)







Concluding remarks

- Reconfigurable acceleration success points to the weakness of evolutionary approaches to parallel processing: hardware (multi core) and software (C++, etc.), at least for HPC applications
- The automation of acceleration is still early on; still required: tools, methodology for writing apps., analysis methodologies, and
- a new hardware substrate (coarser grain, higher speed, shorter P&R times) maybe DFRA (way too many letters)



Concluding remarks (2)

- Reconfigurable HPC can become a reality if underlying software problems can be solved
- In HPC the parallel approach demands rethinking algorithms, data representation, programming approach, models of computation and environment (and hardware)
- There's a lot of research ahead to effectively create parallel translation and array based technology
 - How much automation?
- Tools, Tools, Tools + dedicated methodologies



Example: data flow graph generated by MaxCompiler 4,866 static dataflow cores in 1 chip

ŤUDelft



Thank you very much for your attention





Fixed-point bit-width exploration

8-bit fixed-point



Different parts are explored separately, i.e., when we investigate one part, we keep the bit-widths in other parts a constant high value

Similarly, we observe a significant drop of the error when the SQRT bit-width increases from 8 to 10

Similar precision thresholds observed in both synthetic and field results. This behavior enables an automatic tool to determine the minimum precision that still keeps the result good enough

10-bit fixed-point

5000 6000 7000 8000





Imperial College London

Floating-point bit-width exploration



floating-point: 6-bit exponent



We use the Marmousi synthetic data set as the test data, and explore different combinations of exponent and mantissa bit-width

Reduced Precision Seismic Image

'true' image: single-precision floating-point

2000 3000 4000 5000 6000 7000 8000 0 1000 2000 3000

Full Precision Seismic Image

A precision threshold at exponent width of 6 bits:

- The error drops significantly when we increase the exponent width from 5 bits to 6 bits
- The image also turns from nearly random noise at 5 bits, to almost identical to the 32-bit image at 6 bits





Imperial College London

Global Weather Simulation (10 years ago!)

Atmospheric equations



Equations: Shallow Water Equations (SWEs)
 a0 1 a(AF1) 1 a(AF1)

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

[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]

Imperial College

London

Always double-precision needed?

Range analysis to track the absolute values of all variables



What about error vs area tradeoffs











Accuracy validation



Figure 15. Surface level distribution of the atmosphere at day 15 in the isolated mountain test. Results are obtained on a 10,240 10,240 6 cubed-sphere mesh using 1,536 nodes of the Tianhe-1A. The conical mountain is outlined by the dotted circle in the figure.



Figure 16. Surface level distribution of the atmosphere at day 15 in the real-topography test. We compare results at a 40-km resolution (upper panel) and a 1-km resolution (lower panel).

[Chao Yang, Wei Xue, Haohuan Fu, Lin Gan, et al. 'A Peta-scalable CPU-GPU Algorithm for Global Atmospheric Simulations', PPoPP'2013]









And there is also 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 |

Meshsize: 1,024×1,024×6 14x MaxNode speedup over Tianhe node: 14 times









And power efficiency too

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

Meshsize: 1,024×1,024×6 9 x MaxNode is 9 times more power efficient







