imperas

Avoiding Amdahl's Law: RISC-V Architecture Exploration for AI & ML Many-core Compute Arrays

Simon Davidmann, CEO

December 2019

Amdahl's Law (1967)

- IBM computer architect & entrepreneur
- Left IBM when his ideas were rejected
- Founded Amdahl computers:
  - Cheaper, faster, more reliable
    - IBM plug-compatible...
- Amdahl's law is used in parallel computing to predict the theoretical speedup when using multiple processors

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

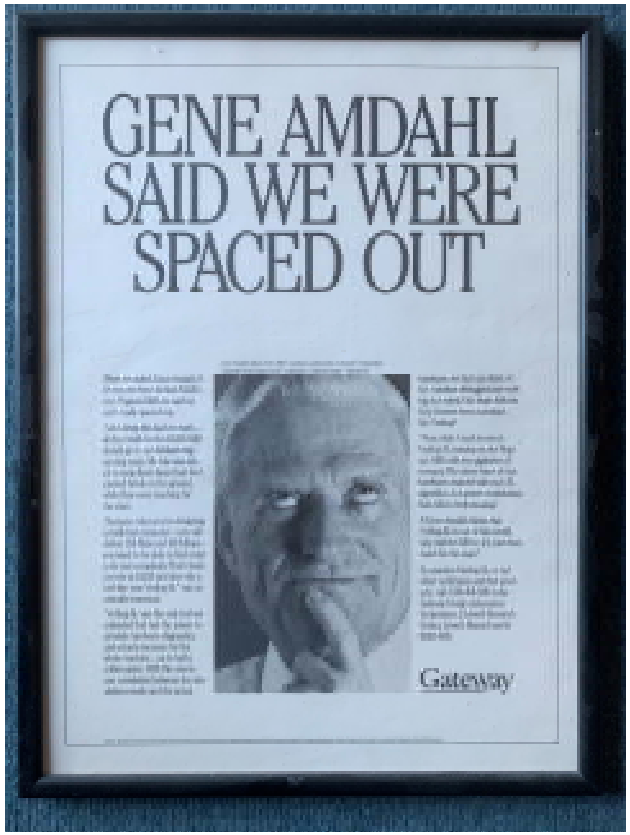
- S_{latency} is the theoretical speedup of the execution of the whole task;
- s is the speedup of the part of the task that benefits from improved system resources;
- p is the proportion of execution time that the part benefiting from improved resources originally occupied.

Amdahl's Law (1967)

- IBM computer architect & entrepreneur
- Left IBM when his ideas were rejected
- Founded Amdahl computers:
  - Cheaper, faster, more reliable
    - IBM plug-compatible…
- Amdahl's law is used in parallel computing to predict the theoretical speedup when using multiple processors

$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

- S_{latency} is the theoretical speedup of the execution of the whole task;
- s is the speedup of the part of the task that benefits from improved system resources;
- p is the proportion of execution time that the part benefiting from improved resources originally occupied.

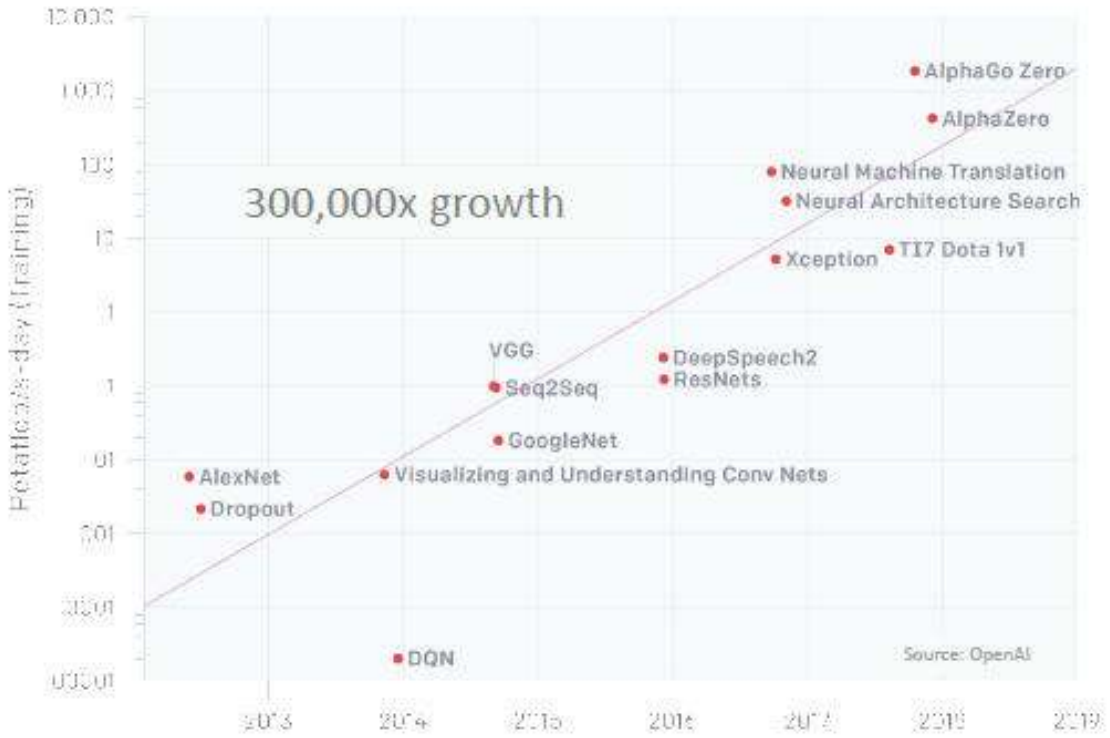


Gateway (developers of Verilog) advert c. 1989

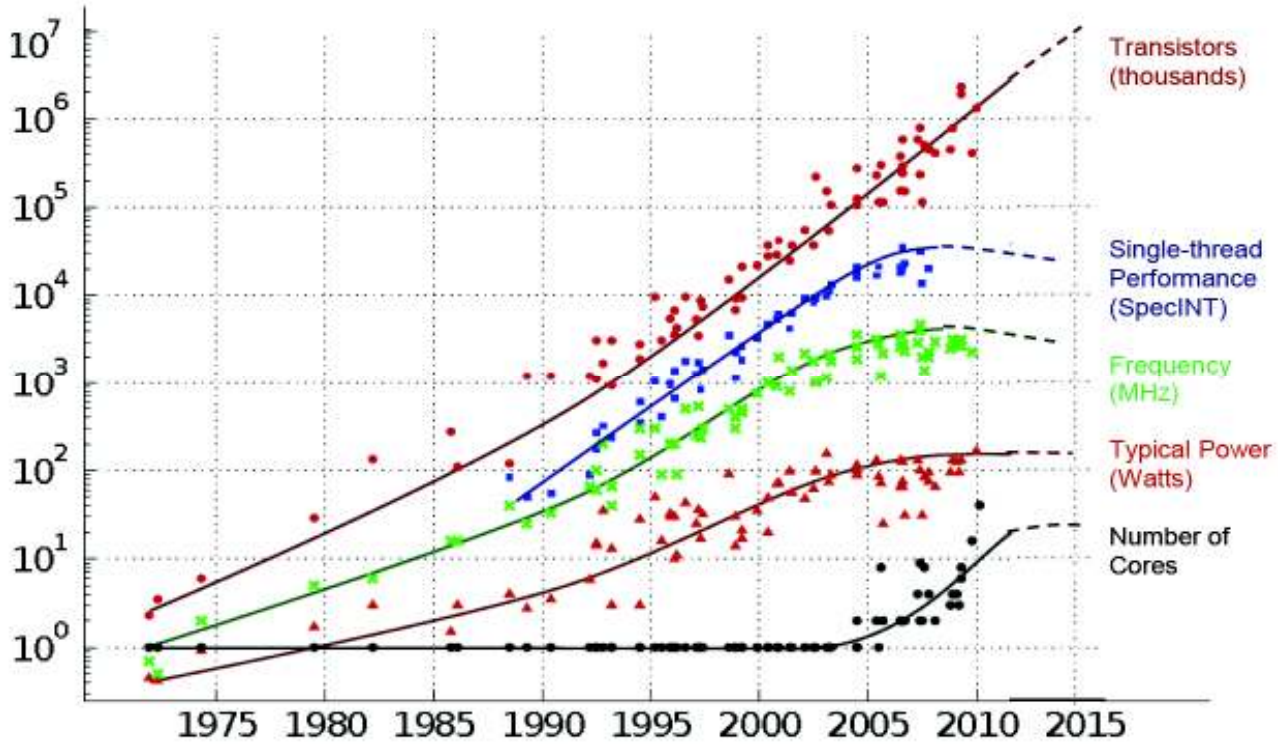
Agenda

- The challenge

MACHINE INTELLIGENCE COMPUTE IS FAST GROWING



35 YEARS OF MICROPROCESSOR TREND DATA



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

Computation needed for ML/AI

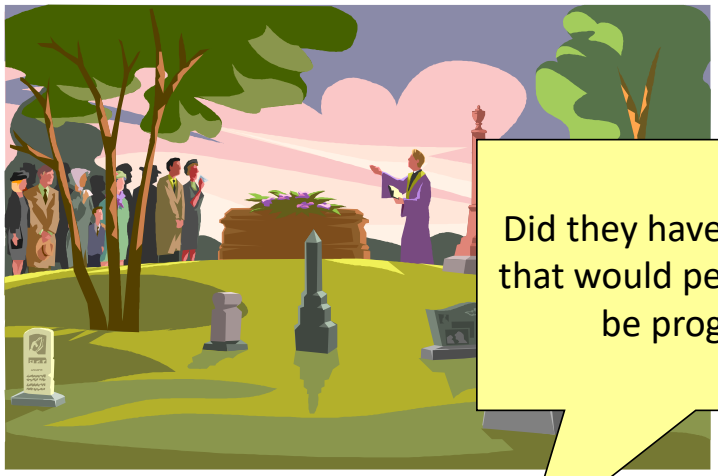
- e.g. 1 Billion MACs for AlexNet – image recognition… training
- X86 is not getting faster
- So we go have to have special processing and run in parallel
- And that's where Amdahl's law comes in…
  - Performance is hindered by the bottleneck(s) – the serial pinch points…

=>

- So it needs to be the correct parallel…
- Designers need to know that their algorithms run "well" on the configuration of hardware they select

Many different approaches to parallelism

Many failed
(a 2007 Imperas slide…)



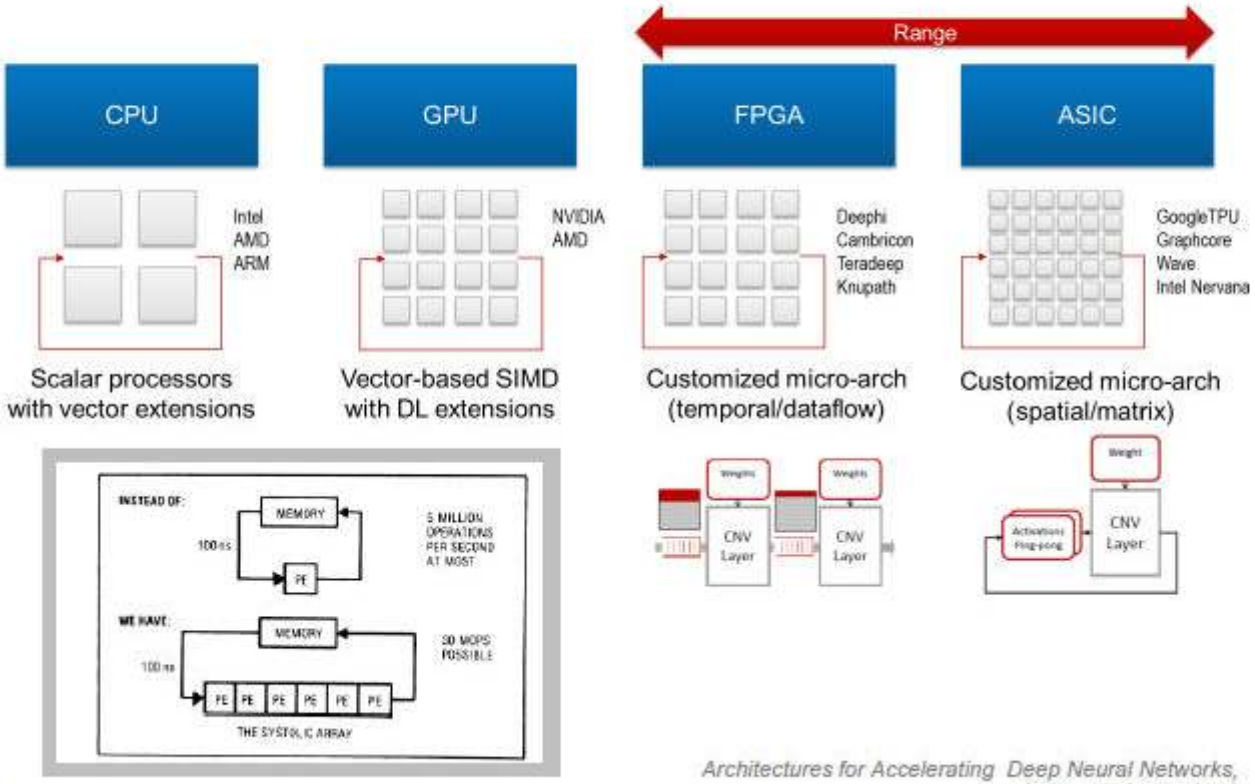
Did they have an architecture that would perform and could be programmed?

Quicksilver ?
Chameleon ?
Morphics ?
Chromatic Research ?
Triscend ?
BOPS ?
Equator ?
PACT ?
Systolix ?
Intrinsity ?
Adelante ?
….?

But now there is established SW

- That has many requirements on hardware platforms to run efficiently

Recent AI Advancements



Architectures for Accelerating Deep Neural Networks, Xilinx, Hot Chips 2018

- Data center chips for deep learning training & inference
  - 20B+, 800mm² designs
  - Thousands of processing elements @ 100+ TOP/s
- Edge IP (primarily) for deep learning inference
  - Mixed scalar/vector/spatial compute
  - Ultra energy efficient: Several TOP/s/W

SYNOPSYS

 8

BULK SYNCHRONOUS PARALLEL (BSP)

Software bridging model for parallel computing

Compute

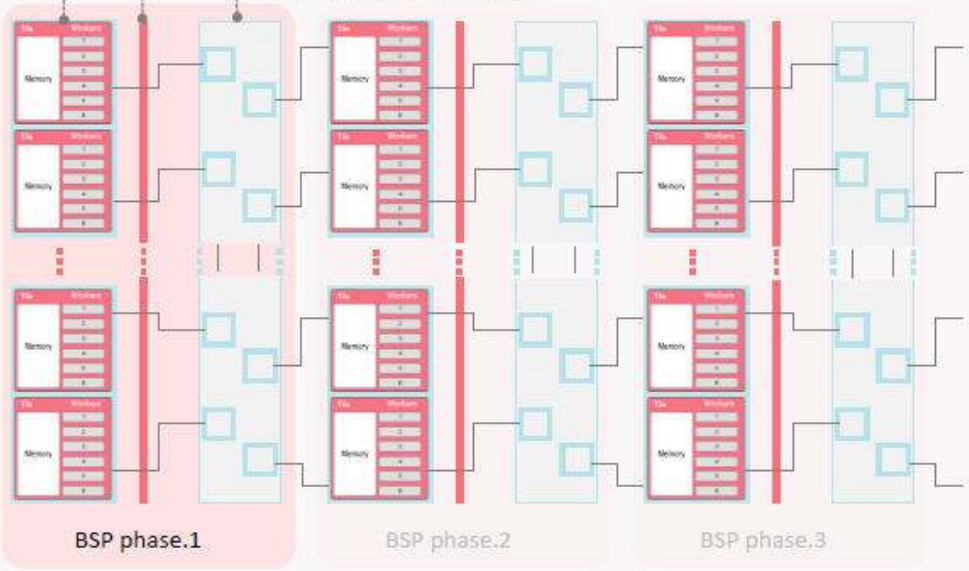
10,000s of compute threads all operating in parallel each with all the data that they need, held locally

BSP Sync

All threads are synchronized

Exchange

Data is exchanged so that every thread has all the data that it needs for the next phase of Compute

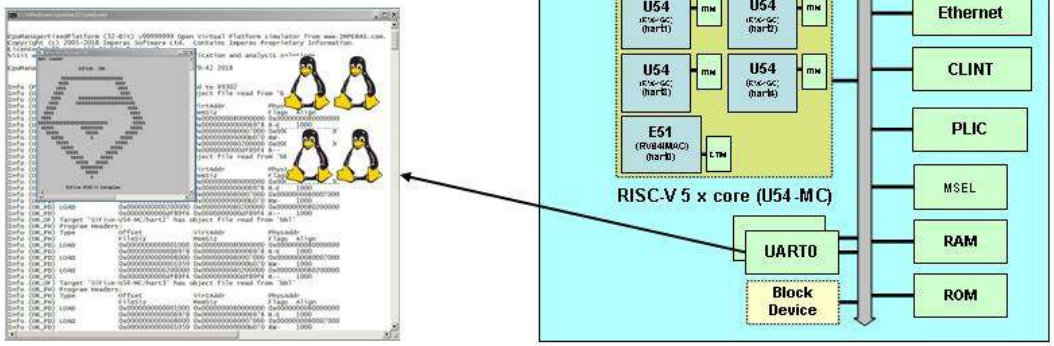


How we help

Imperas

- We model the processors (240+ in library)
  - You can create your own, with own ISA, or you can add your own extensions / smarts to ours
- We have a library of the behavioural components (300+ in library)
  - You can add your own, or configure ours
- Our technology allows you to build/model the hierarchical platform (40+ in library)
  - Your configuration of processors, behavioral components and hierarchies
- Our simulators can simulate from core to system
  - We have industry leading simulation performance, 300MIPS -2 Billion instructions sec
  - Single core, multi-core, AMP, SMP
  - From bare metal to SMP Linux boot in under 10 secs

OVPsim RISC-V (U54-MC) Linux Kernel



Imperas FU540 Virtual Platform

- Virtual Platforms built using Open Virtual Platforms (OVP) APIs
  - Using OVP processor and peripheral models
  - Models are open source
- Simulated unmodified production binaries
- Software does not know it is not on hardware
- Runs very fast, 100-2,000MIPS

[SMP Linux boot in under 10 secs]

Imperas

- We model the processors (240+ in library)
  - You can create your own, with own ISA, or you can add your own extensions / smarts to ours
- We have a library of the behavioural components (300+ in library)
  - You can add your own, or configure ours
- Our technology allows you to build/model the hierarchical platform (40+ in library)
  - Your configuration of processors, behavioral components and hierarchies
- Our simulators can simulate from core to system
  - We have industry leading simulation performance, 300MIPS -2 Billion instructions sec
  - Single core, multi-core, AMP, SMP
  - From bare metal to SMP Linux boot in under 10 secs
- Full holistic multi-core platform debug (Eclipse based)
- Advanced tools for analysis and profiling
- Full capability for hardware verification
  - Works with SystemVerilog UVM simulators from Cadence, Mentor, Synopsys, Metrics

Example: RISC-V Vector engine

- Imperas OVP model of RISC-V
  - Full specification, all user, privilege, vector instructions and functionality
  - Single core, single thread performance 300-2,000MIPS speed
    - 24 cores, single thread performance dhrystones 1,000 MIPS overall (40MIPS per core)
  - Parallel simulation: host core utilization ~2x overall for 4 cores (your mileage may differ)
- Uses very little RAM for simulation and uses sparse memory - so scales well
  - Seen simulations of platforms with up to 1024 cores
- Vector engine configurable for HW options: VLEN, SLEN, ELEN

Related experience

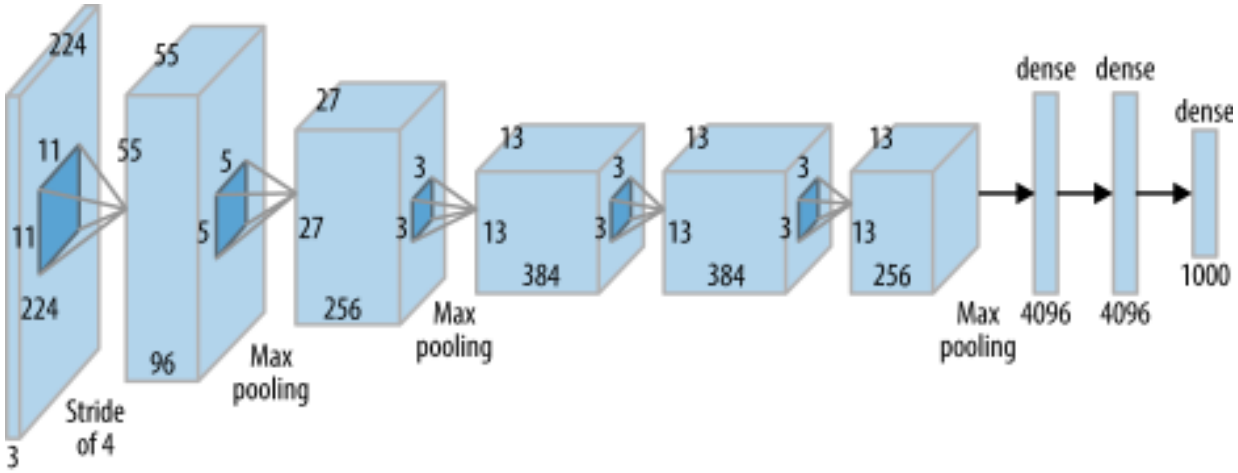
- Customer project
  - Full ML/AI engine
  - 150+cores
    - Many with RISC-V Vector engine
  - Runs simulation in 2hrs @ 500MIPS
    - Cross compiled software running on simulated CPUs
  - Allows software stack development
  - Allows hardware platform config, re-config, architectural changes
    - Explore performance options
    - Runs real software (production binaries) – can see how it interacts with HW config
  - Running in Imperas a year before RTL commit
    - Customer has SW and is looking to design HW to make it work the way they want…
  - Also a by-product: kick-start SoC process by feeding models into HW DV at start

Another: Japanese partner

- Summary
  - Platform : ARM Cortex-A57 x 1 + RISC-V RV64GCV x 17
  - Application1 : AlexNet image recognition deep neural network
- Keypoints
  - “Imperas simulator can simulate heterogeneous virtual platform”
  - “Imperas also provides dedicated debugger which can debug hetero-system (ex. ARM and RISC-V) using one debugger at same time”
  - “Very fast. This example runs (at most) 10 times slower than native x64 execution on host PC”

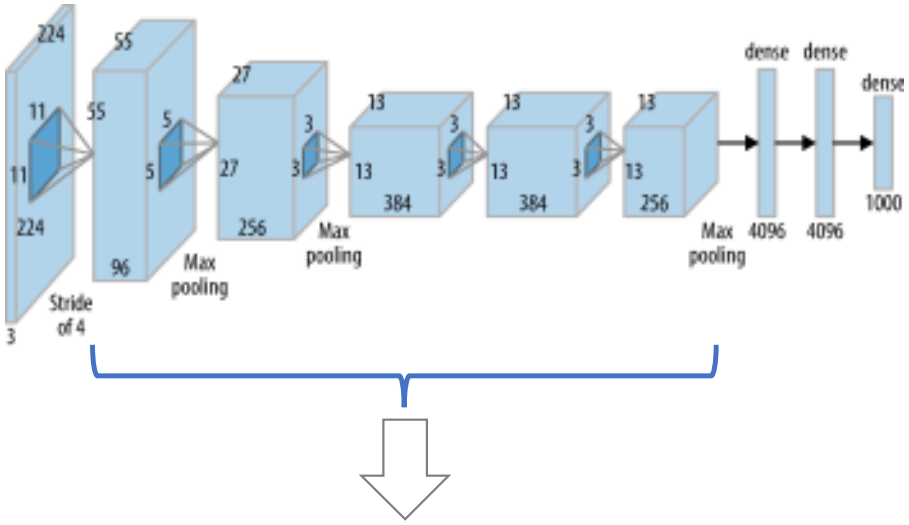
Imagenet with AlexNet deep neural network

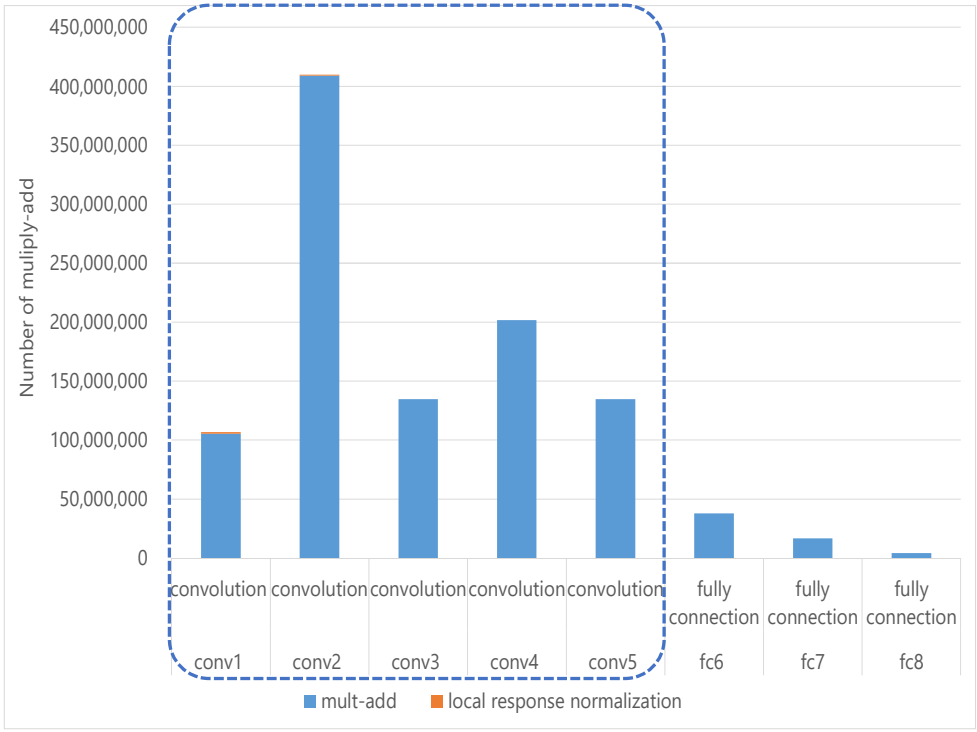
- AlexNet (University of Toronto, 2012)
  - https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96



- Hyper parameters
  - Number of Parameter : 58 M (float32)
  - Computation cost : 1000 M (Number of multiply-add)

Parallelization for multiple core

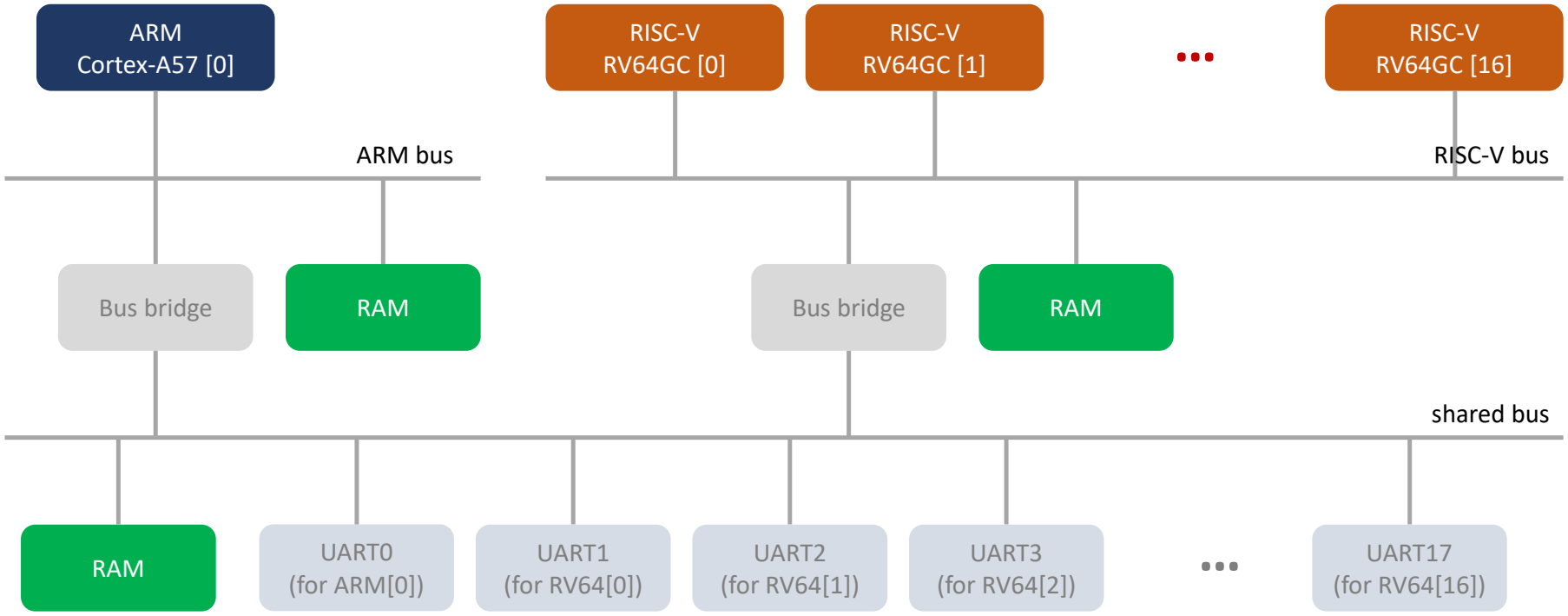




Convolution layers have a lot of calculation

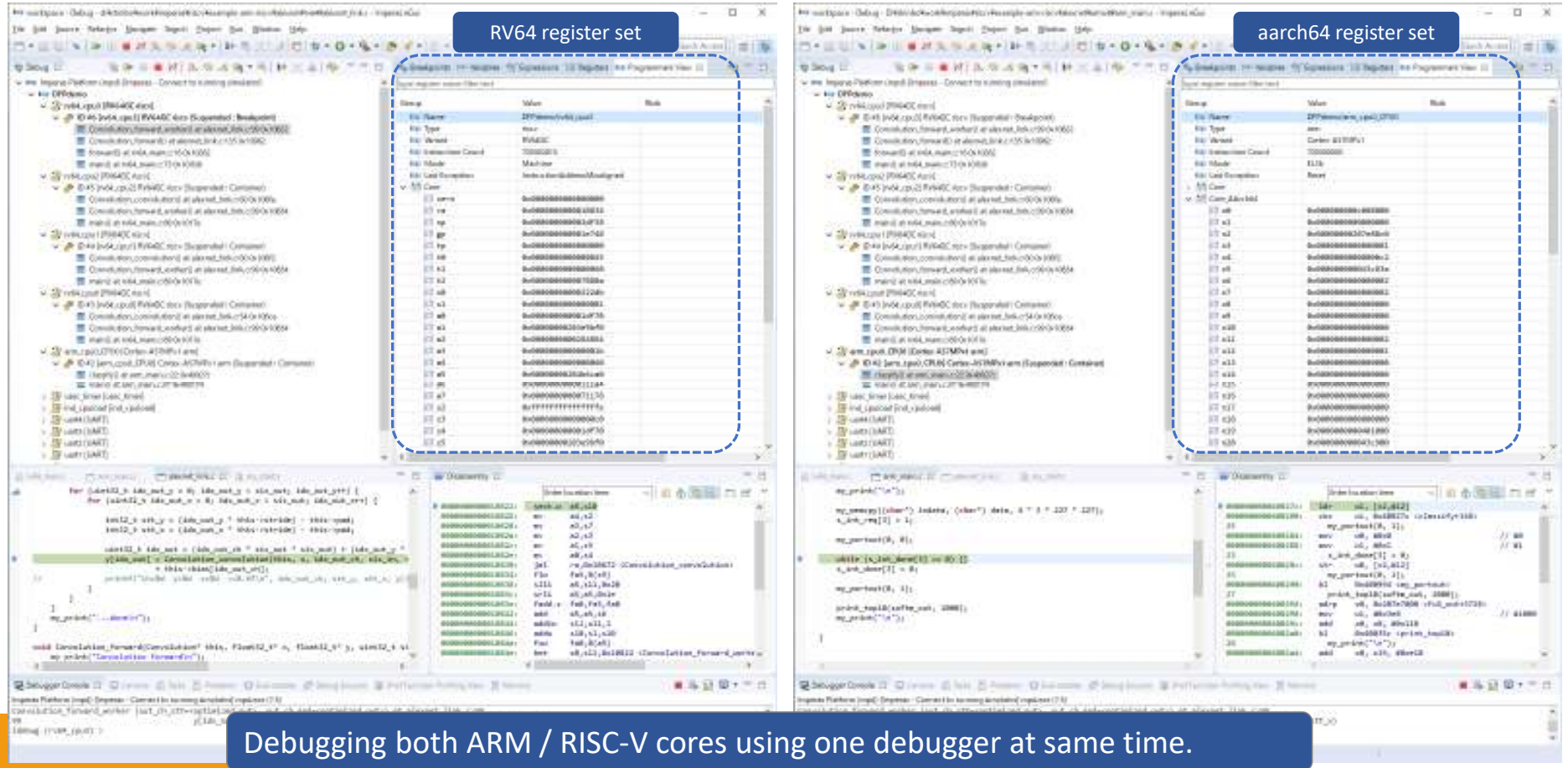
Parallelized these layers to use 16 CPU cores

Platform


ARM
Cortex-A57 [0]
RISC-V
RV64GC [0]
RISC-V
RV64GC [1]
RISC-V
RV64GC [16]
ARM bus
RISC-V bus
Bus bridge
RAM
Bus bridge
RAM
shared bus
RAM
UART0
(for ARM[0])
UART1
(for RV64[0])
UART2
(for RV64[1])
UART3
(for RV64[2])
UART17
(for RV64[16])

Executing simulation – different consoles

Single Multi-Processor Debug



Debugging both ARM / RISC-V cores using one debugger at same time.

Some classification result

imperas

tree_227x227.bin

0 : 937 : 0.024571 n07714990 broccoli
1 : 738 : 0.020480 n03991062 pot, flowerpot
2 : 990 : 0.014112 n12768682 buckeye, horse chestnut, conke
3 : 853 : 0.012724 n04417672 thatch, thatched roof
4 : 483 : 0.010819 n02980441 castle
5 : 595 : 0.010186 n03496892 harvester, reaper
6 : 703 : 0.009717 n03891251 park bench
7 : 984 : 0.009478 n11879895 rapeseed
8 : 410 : 0.008571 n02727426 apiary, bee house
9 : 958 : 0.008400 n07802026 hay

pasta_227x227.bin

0 : 533 : 0.014579 n03207743 dishrag, dishcloth
1 : 770 : 0.013624 n04120489 running shoe
2 : 842 : 0.012774 n04371430 swimming trunks, bathing trun
3 : 911 : 0.011252 n04599235 wool, woolen, woollen
4 : 415 : 0.011122 n02776631 bakery, bakeshop, bakehouse
5 : 658 : 0.010166 n03775071 mitten
6 : 748 : 0.010100 n04026417 purse
7 : 840 : 0.010091 n04367480 swab, swob, mop
8 : 883 : 0.009004 n04522168 vase
9 : 659 : 0.008804 n03775546 mixing bowl

cup_227x227.bin

0 : 504 : 0.157205 n03063599 coffee mug
1 : 968 : 0.105636 n07930864 cup
2 : 967 : 0.049826 n07920052 espresso
3 : 809 : 0.046456 n04263257 soup bowl
4 : 725 : 0.045708 n03950228 pitcher, ewer
5 : 441 : 0.016535 n02823750 beer glass
6 : 738 : 0.015744 n03991062 pot, flowerpot
7 : 901 : 0.015085 n04579145 whiskey jug
8 : 463 : 0.014723 n02909870 bucket, pail
9 : 969 : 0.013509 n07932039 eggnog

fujisan_227x227.bin

0 : 980 : 0.056212 n09472597 volcano
1 : 977 : 0.023318 n09421951 sandbar, sand bar
2 : 976 : 0.011876 n09399592 promontory, headland, head, foreland
3 : 978 : 0.011804 n09428293 seashore, coast, seacoast, sea-coast
4 : 975 : 0.011525 n09332890 lakeside, lakeshore
5 : 112 : 0.008982 n01943899 conch
6 : 979 : 0.008915 n09468604 valley, vale
7 : 972 : 0.008791 n09246464 cliff, drop, drop-off
8 : 974 : 0.008665 n09288635 geyser
9 : 930 : 0.008184 n07684084 French loaf

rabbit_227x227.bin

0 : 331 : 0.134504 n02326432 hare
1 : 333 : 0.114495 n02342885 hamster
2 : 332 : 0.112388 n02328150 Angora, Angora rabbit
3 : 330 : 0.074634 n02325366 wood rabbit, cottontail, cotte
4 : 279 : 0.030452 n02120079 Arctic fox, white fox, Alopex
5 : 356 : 0.017656 n02441942 weasel
6 : 338 : 0.015547 n02364673 guinea pig, Cavia cobaya
7 : 362 : 0.011832 n02447366 badger
8 : 361 : 0.010876 n02445715 skunk, polecat, wood pussy
9 : 357 : 0.010846 n02442845 mink

cat_227x227.bin

0 : 281 : 0.115859 n02123045 tabby, tabby cat
1 : 282 : 0.054218 n02123159 tiger cat
2 : 287 : 0.047298 n02127052 lynx, catamount
3 : 285 : 0.043925 n02124075 Egyptian cat
4 : 290 : 0.011876 n02128925 jaguar, panther, Panthera onca, F
5 : 292 : 0.011247 n02129604 tiger, Panthera tigris
6 : 286 : 0.011120 n02125311 cougar, puma, catamount, mount
7 : 289 : 0.009661 n02128757 snow leopard, ounce, Panthera u
8 : 280 : 0.009410 n02120505 grey fox, gray fox, Urocyon cinere
9 : 288 : 0.008847 n02128385 leopard, Panthera pardus

lion_227x227.bin

0 : 291 : 0.103345 n02129165 lion, king of beasts, Panthera le
1 : 366 : 0.042367 n02480855 gorilla, Gorilla gorilla
2 : 367 : 0.031897 n02481823 chimpanzee, chimp, Pan troglo
3 : 261 : 0.030083 n02112350 keeshond
4 : 372 : 0.028271 n02486410 baboon
5 : 297 : 0.028121 n02134418 sloth bear, Melursus ursinus, U
6 : 369 : 0.027656 n02483708 siamang, Hylobates syndactylu
7 : 373 : 0.026115 n02487347 macaque
8 : 365 : 0.018029 n02480495 orangutan, orang, orangutang,
9 : 260 : 0.016685 n02112137 chow, chow chow

panda_227x227.bin

0 : 388 : 0.781814 n02510455 giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca
1 : 850 : 0.008743 n04399382 teddy, teddy bear
2 : 295 : 0.007253 n02133161 American black bear, black bear, Ursus americanus, Euarctos americanus
3 : 270 : 0.007223 n02114548 white wolf, Arctic wolf, Canis lupus tundrarum
4 : 232 : 0.006188 n02106166 Border collie
5 : 296 : 0.005935 n02134084 ice bear, polar bear, Ursus Maritimus, Thalarctos maritimus
6 : 222 : 0.005294 n02104029 kuvasz
7 : 279 : 0.005271 n02120079 Arctic fox, white fox, Alopex lagopus
8 : 805 : 0.005230 n04254680 soccer ball
9 : 294 : 0.004849 n02132136 brown bear, bruin, Ursus arctos

Another: Japanese partner

- Summary
  - Platform : ARM Cortex-A57 x 1 + RISC-V RV64GCV x 17
  - Application1 : AlexNet image recognition deep neural network
- Keypoints
  - “Imperas simulator can simulate heterogeneous virtual platform”
  - “Imperas also provides dedicated debugger which can debug hetero-system (ex. ARM and RISC-V) using one debugger at same time”
  - “Very fast. This example runs (at most) 10 times slower than native x64 execution on host PC”

Summary

- Fast Imperas simulation allows software to run on virtual platform many months before RTL commit
- Allows analysis of performance on different hardware configuration choices
- Imperas allows heterogeneous platforms with full OS running
- Provides detailed analysis, profiling, performance and debug tooling
- Has all the current RISC-V specification features and enables you to develop custom instructions
- Imperas is fast, very fast,…

More Information:

- Stop by the Imperas booth #416 at the Dec 2019 RISC-V Summit