HUAWEI | MUNICH RESEARCH CENTER

AI-enabled science

— Challenges and opportunities arising from the convergence of artificial intelligence, high-performance computing and big data

Heiko Joerg Schick
Chief Architect | Advanced Computing

Presenting the work of many people at Huawei

Version 1 - DRAFT

HUAWEI TECHNOLOGIES DÜSSELDORF GmbH

HUAWEI

Agenda

Use cases

Challenges for system architecture (hardware, software, workflow and algorithms)

Artificial intelligence

Special computer architectures? How come, for what reason and why?

Examples

In ten years

Heiko Joerg Schick
Chief Architect | Advanced Computing
Munich Research Center

HUAWEI TECHNOLOGIES
Duesseldorf GmbH
Riesstrasse 25, C3
80992 Munich

Mobile +49-151-54682218
E-mail heiko.schick@huawei.com

LinkedIn
https://www.linkedin.com/in/heikojoergschick/

In symbols one observes an advantage in discovery which is greatest when they expressed the exact nature of a thing briefly and, as it were, picture it; then indeed the labor of thought is wonderfully diminished.

In Symbolen beobachtet man einen Entdeckungsvorteil, der am größten ist, wenn sie die genaue Natur einer Sache kurz zum Ausdruck bringen und gleichsam abbilden; dann wird in der Tat die Arbeit des Denkens wunderbar verringert.

— Gottfried Wilhelm Leibniz

In models one observes an advantage in discovery which is greatest when they expressed the exact nature of a thing briefly and, as it were, picture it; then indeed the labor of thought is wonderfully diminished.

In Symbolen beobachtet man einen Entdeckungsvorteil, der am größten ist, wenn sie die genaue Natur einer Sache kurz zum Ausdruck bringen und gleichsam abbilden; dann wird in der Tat die Arbeit des Denkens wunderbar verringert.

— Gottfried Wilhelm Leibniz

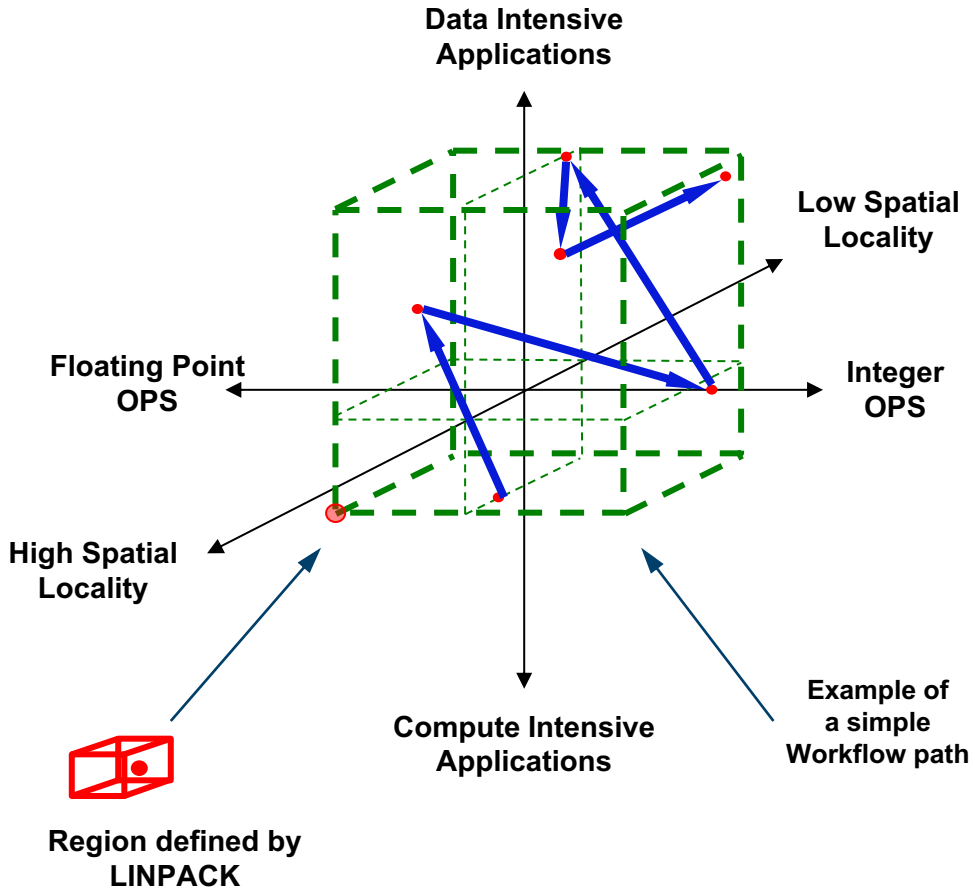
Integration of artificial intelligence and high performance computing

Use cases:

1. **Using HPC technologies to execute and enhance AI performance.**
2. **Using HPC simulations to train AI algorithms.**
3. **Using AI algorithms to configure and autotune AI workloads or HPC simulations.**
4. **Using AI algorithms to analyse results of HPC simulations.**
5. **Using AI algorithms to learn from HPC simulations and produce learned surrogates.**

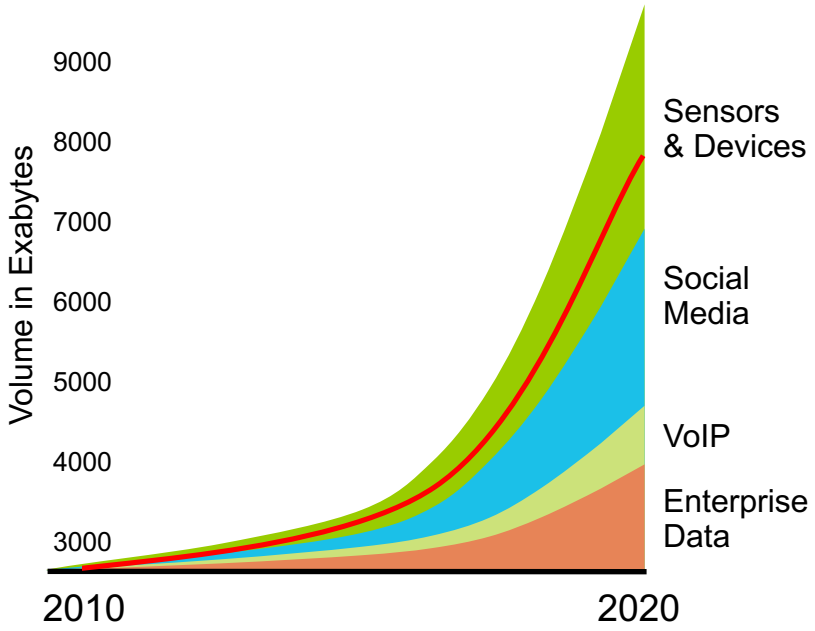
Workload-optimized Systems Requirements

- **Real-world workflows test a broad range of system design points:**
  - Challenge is to determine the best design region boundaries
  - Benchmarks are necessary but not sufficient …
  - Need to examine workflows to get the full picture of requirements
  - Need to understand how multiple workflows run simultaneously across system(s)

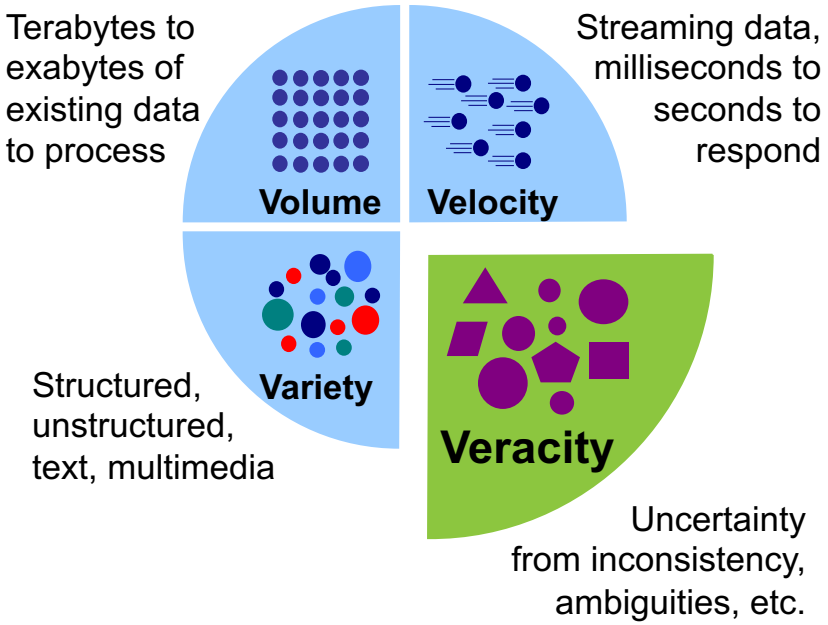


Powerful Information: AI-enabled Science — a new Area of Computing

Data volume is on the rise



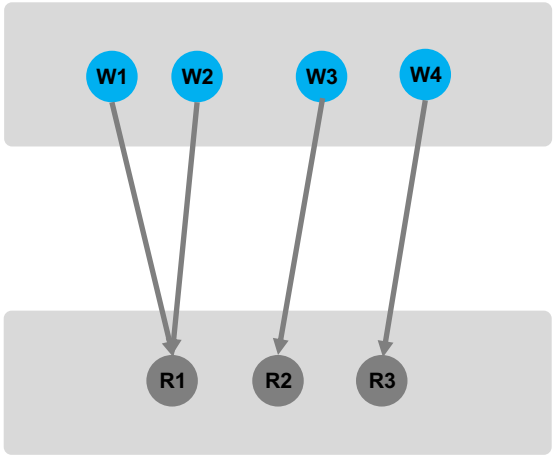
Dimensions of data growth



On the Roads to Software Defined Environments (SDE)

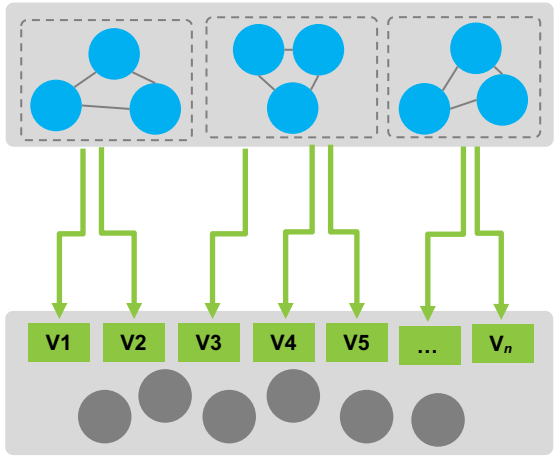
- Workload types are growing and becoming more flexible and diverse.
- Cloud infrastructure is becoming programmable to meet the requirements in efficiency and resiliency.
- Heterogeneity is increasingly present and important.

Traditional



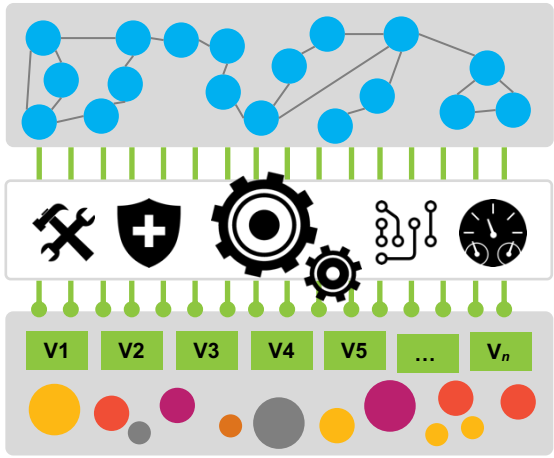
- Few, stable, and well known workloads
- Fixed system hardware, manual scaling
- Hard-wired workload, minimal configuration

Current



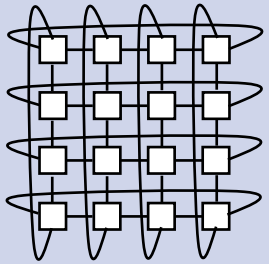
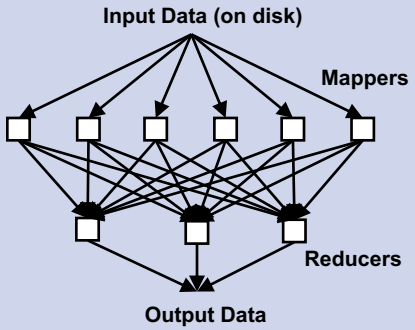
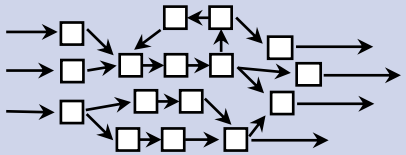
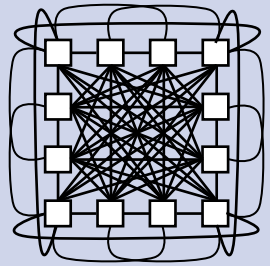
- Diverse workloads, limited patterns
- Homogenous resource pooling
- Expert configuration and mapping of workloads

Future



- Rapidly changing workloads, dynamic patterns
- Dynamic automatic composition of heterogeneous systems
- Autonomic and proactive management

Four Types of Parallel Architectures for Advanced Computing

	Compute Intensive	Data Intensive: Data at Rest	Data Intensive: Streaming Data	AI-enabled Science
Programming Language	C/C++, Fortran, MPI, OpenMP, CUDA	Java, JAQL, Python	SPL, C, Java	C/C++, Python, UPC, SHMEM, MPI, OpenMP
Characteristics	• Data is Generated • Long Running • Small Input • Massive Output	• Data at Rest • Low Velocity • Mixed Variety • High Volume	• Data in Motion • High Velocity • Mixed Variety • High Volume	• Data is Moving • Long Running • All Data View • Small Messages
Category	• Network Dependent • Structured Communication	• Embarrassingly Parallel • Structured Communication	• Embarrassingly Parallel • Random Communication	• Network Dependent • Random Communication
Applications	LINPACK	Apache Hadoop / Key-Value Databases	Apache Storm	
Network Topology				
Scaling	Strong	Weak		Both
Cluster Stack Management	Custom	Diverse control systems		N/A
Schedulers	Slurm	YARN, Mesos		N/A
File System	Cluster File-System	Distributed across nodes (e.g. HDFS)		POSIX or HDFS
Operating system	RedHat / CentOS	RedHat / CentOS		Ubuntu

Verticals & Applications

- **Academia**
  - Global Climate Research
  - Astrophysics
  - Material Science
- **Research Organizations**
  - High Energy Physics
  - Plasma Physics
- **Earth Sciences**
  - Forecasting
  - Meteorology
  - Solid Earth Sciences
  - Oceanography
  - Geophysics
- **Government**
  - All source Intelligence
  - Cybersecurity
  - Statutory Pension
- **Financial Services & Insurance**
  - Stress Testing
  - Risk Analysis
  - Algorithm Backtesting
- **Digital Content & Telecommunication**
- **Manufacturing**
  - Engine Development
  - Combustion Simulation
  - Vehicle & Aircraft Development
  - Electric Design Automation
- **Oil and Gas**
  - Upstream (e.g. Resource Discovery)
  - Downstream
- **Life Sciences**
  - Personalized Medicine
  - System Biology
  - Molecular Simulation
- **Energy & Transportation**
  - Distribution & Logistic
- **Cloud Providers**
  - Computer-aided Engineering as a Service

Traditional Computing

- Principal Component Analysis
- Collision Detection
- Random Numbers
- Encryption & Decryption
- Hashing
- Checksum
- Routing
- Error Detection and Correction
- Finite Elements
- Sequence Alignment
- Scheduling
- Pattern Matching

Computational Methods

- **Dense Linear Algebra** — Computationally limited
- **Sparse Linear Algebra** — 50% computation, 50 % memory bandwidth
- **Spectral Methods** — Memory latency limited
- **N-Body Methods** — Computationally limited
- **Structured Grids** — Currently more memory bandwidth limited
- **Unstructured Grids** — Memory latency limited
- **MapReduce** — Problem dependent
- **Combinational Logic** — Computationally limited
- **Graph Traversal** — Memory latency limited
- **Dynamic Programming** — Memory latency limited
- **Backtrack and Branch-and-Bound**
- **Construct Graphical Models**
- **Finite State Machine**

Hardware

Algorithms & Libraries

Computer Vison

Image Processing

Speech Recognition

Machine Learning

Clustering	Deep Learning Networks
Dimensionality Reduction	Convolutional Neural Networks
Classification	Recurrent Neural Network
Conventional Neural Networks	Deep Belief Networks

- Unsupervised Learning
- Instance-based Learning
- Deep Reinforcement Learning
- General-Purpose Machine Intelligence
- Contextual Computing

Analytics

Linear Regression	Compression
Search	Graph
Sorting	Optimization
Merging	Encryption
Monte Carlo	Statistic and Stochastic
Data Mining	Recommendation
Bayesian Methods	Hidden Markov Models
Decision Tree	

Natural Language Processing

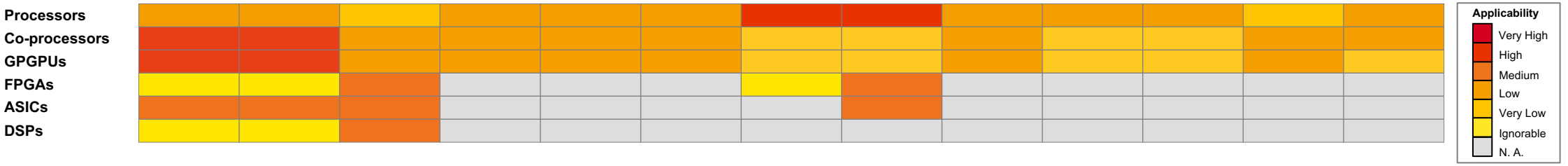
- Natural-Language Generation
- Real-time Interpretation and Translation
- Natural-Language Questioning Answering
- Translingual Collaborative Spaces

Computational Methods

Dense Linear Algebra — Computationally limited

Sparse Linear Algebra — 50% computation, 50 % memory bandwidth

Hardware



Implementation of end-to-end lifecycle in AI projects [Alake, 2020], [Sato et al., 2019]

- Problem statement
- Ideal problem solution
- Understanding and insight into the problem
- Technical requirements

- Data structure and source
- Solution form
- Model architecture
- Algorithm research
- Hardware requirements

- Data gathering (diverse, unbiased and abundant)

- Data reformatting
- Data cleaning
- Data normalization
- Data augmentation

- Usage of pre-trained models?
- Fine-tuning pre-trained models

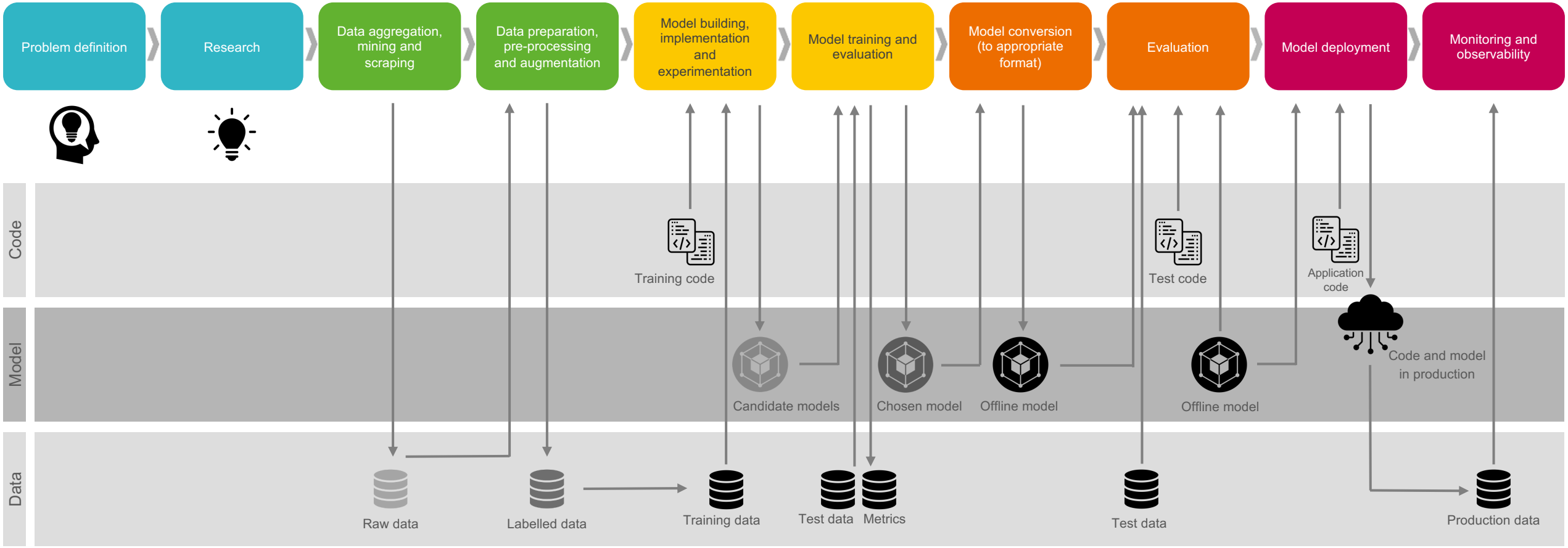
- Training accuracy
- Validation accuracy
- Training loss
- Validation loss
- Underfitting or overfitting?

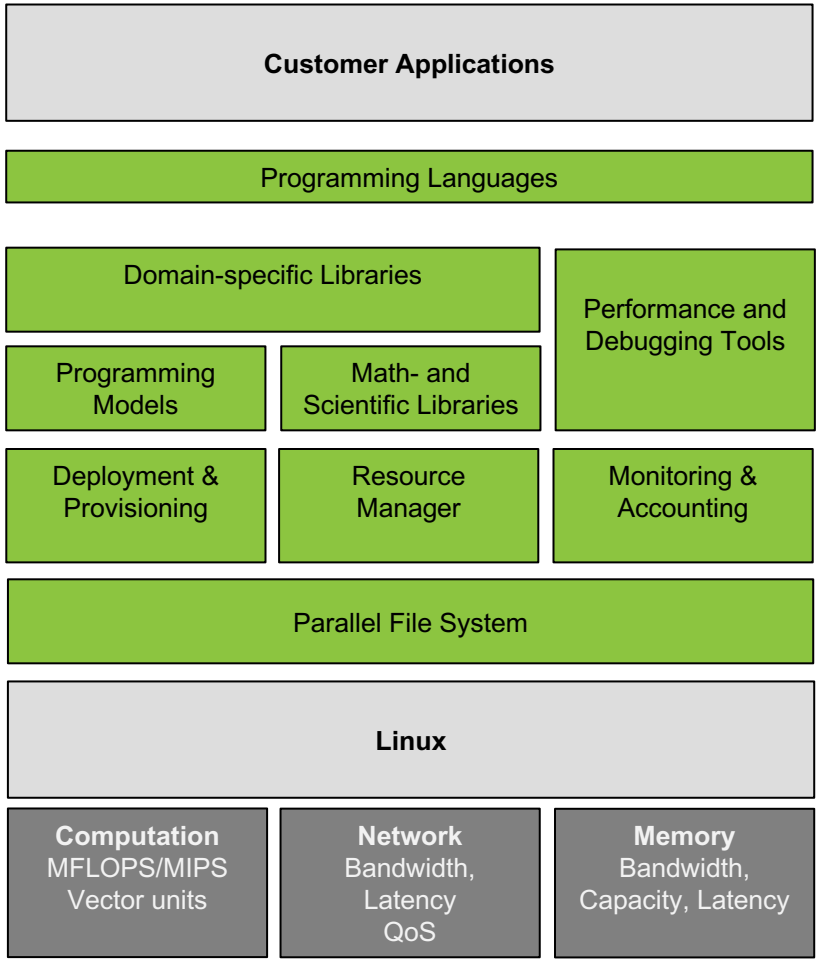
- Refine and optimise model
- Model conversion
- Mobile-optimised model

- Confusion matrix (error matrix)
- Precision-recall

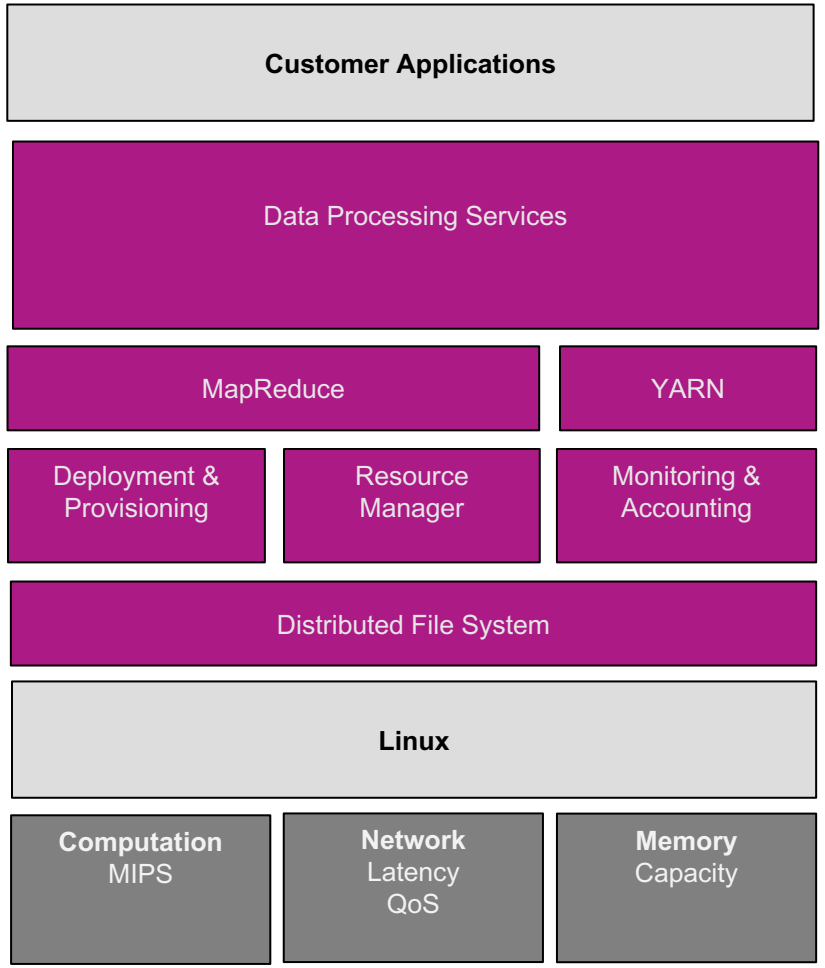
- UI interface to access model functionalities
- Continuous integration pipeline that enables model redeployment

- Model performance monitoring system

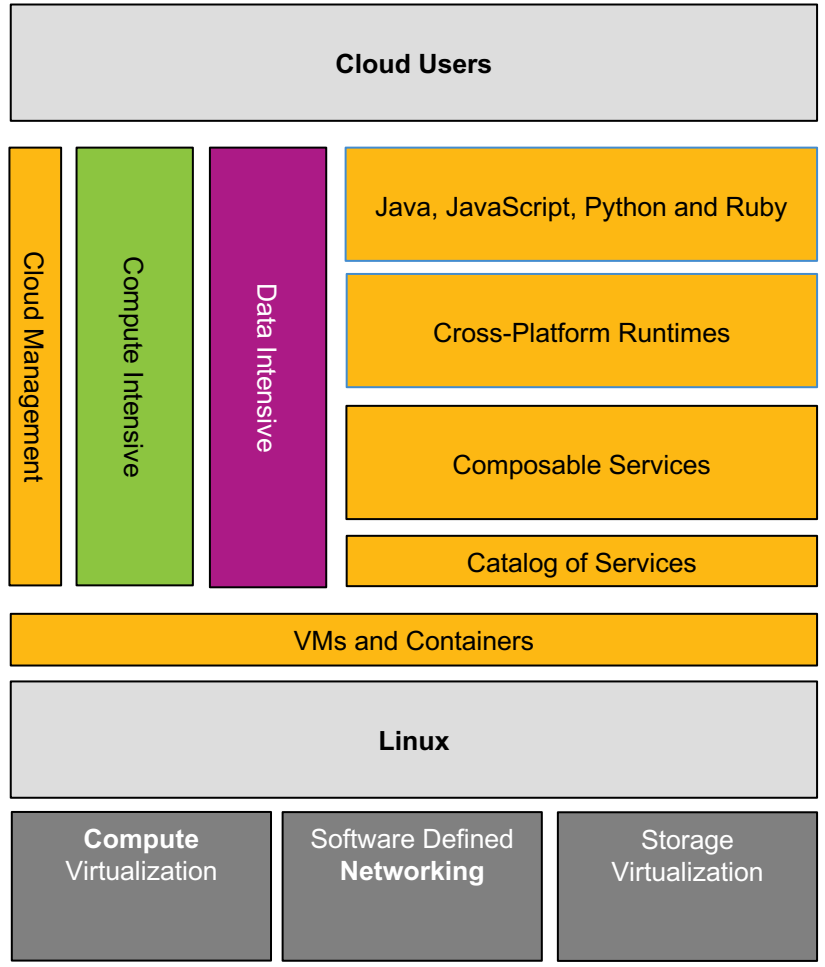




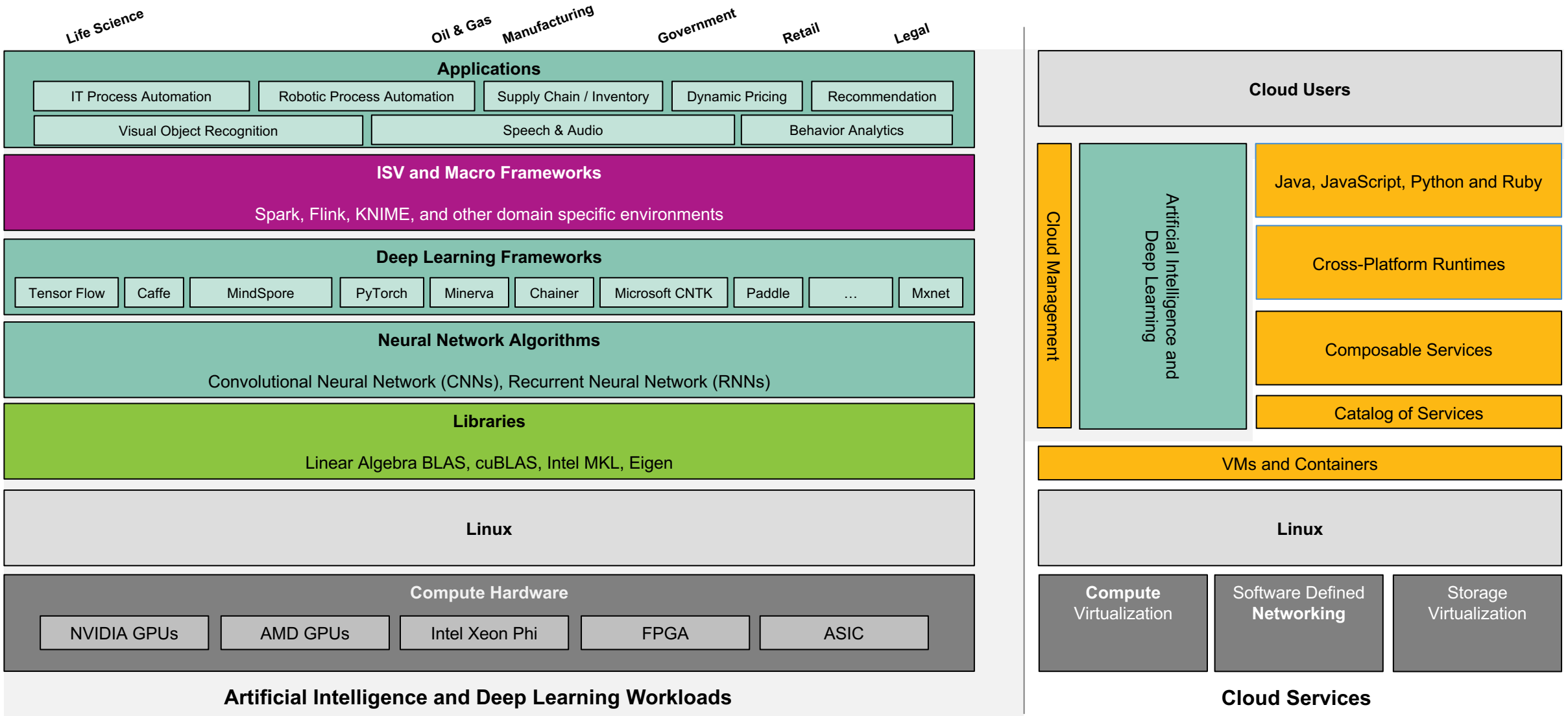
Compute Intensive Workloads



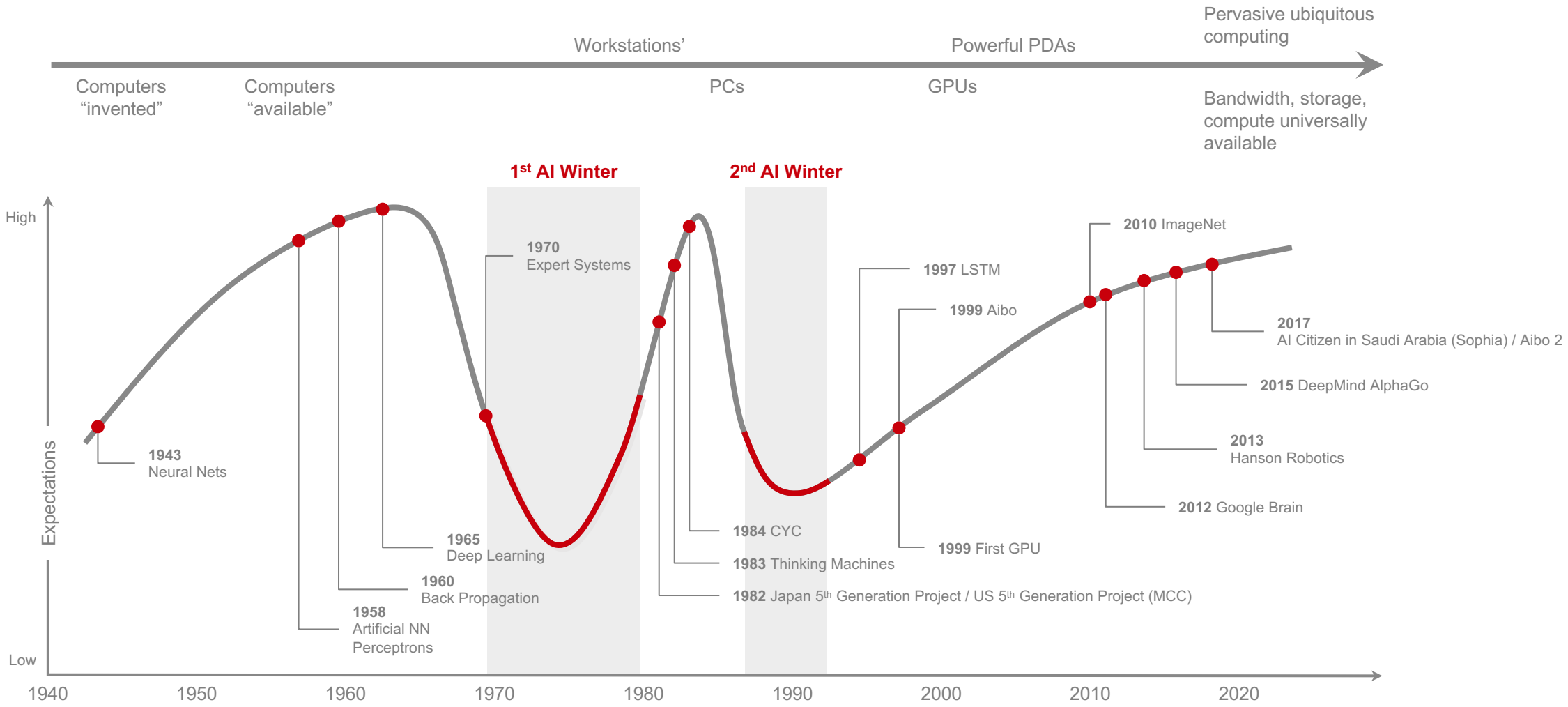
Data Intensive Workloads



Cloud Services


Life Science
Oil & Gas
Manufacturing
Government
Retail
Legal
Applications
IT Process Automation
Robotic Process Automation
Supply Chain / Inventory
Dynamic Pricing
Recommendation
Visual Object Recognition
Speech & Audio
Behavior Analytics
ISV and Macro Frameworks
Spark, Flink, KNIME, and other domain specific environments
Deep Learning Frameworks
Tensor Flow
Caffe
MindSpore
PyTorch
Minerva
Chainer
Microsoft CNTK
Paddle
...
Mxnet
Neural Network Algorithms
Convolutional Neural Network (CNNs), Recurrent Neural Network (RNNs)
Libraries
Linear Algebra BLAS, cuBLAS, Intel MKL, Eigen
Linux
Compute Hardware
NVIDIA GPUs
AMD GPUs
Intel Xeon Phi
FPGA
ASIC
Artificial Intelligence and Deep Learning Workloads
Cloud Users
Cloud Management
Artificial Intelligence and Deep Learning
Java, JavaScript, Python and Ruby
Cross-Platform Runtimes
Composable Services
Catalog of Services
VMs and Containers
Linux
Compute Virtualization
Software Defined Networking
Storage Virtualization
Cloud Services

The hype roller coaster of artificial intelligence [Villain, 2019]



Neural networks beat human performance /1 [Giró-i-Nieto, 2016], [Gershgorn, 2017]

— Example: Image classification on ImageNet

15 million images in dataset, 22,000 object classes (categories) and 1 million images with bounding boxes.

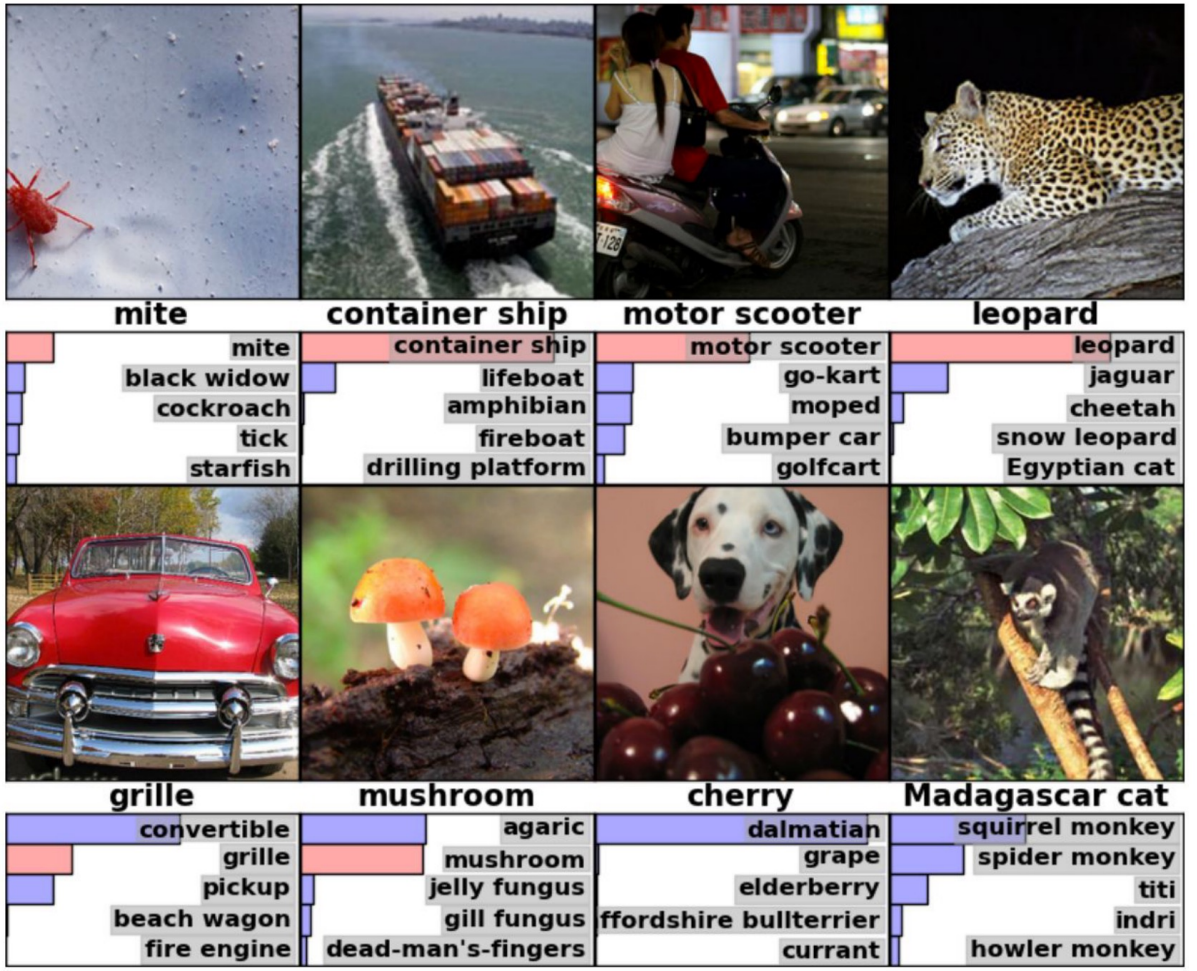
IMAGENET

Neural networks beat human performance /1 [Giró-i-Nieto, 2016], [Gershgorn, 2017]

— Example: Image classification on ImageNet

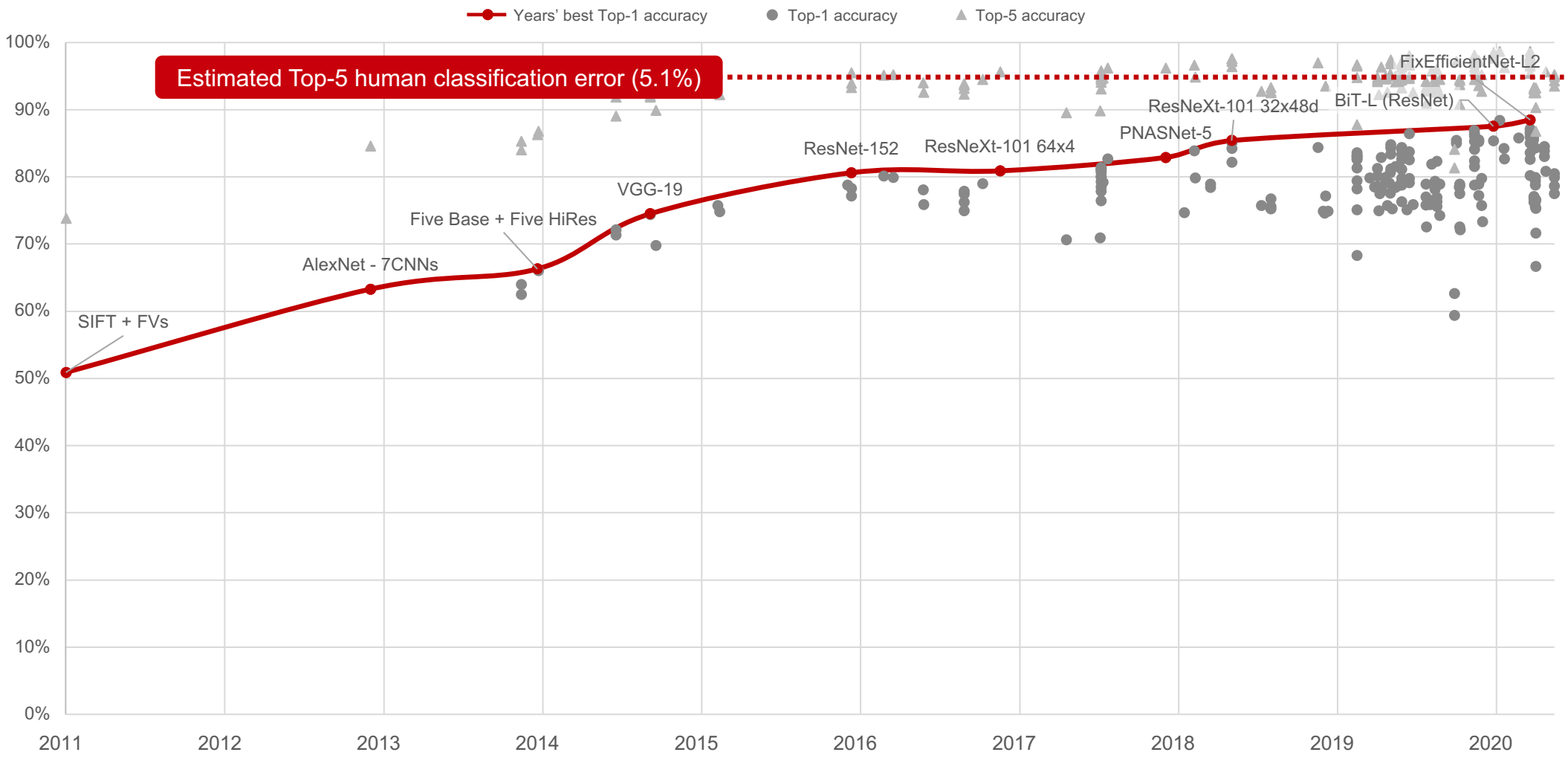
15 million images in dataset, 22,000 object classes (categories) and 1 million images with bounding boxes.

IMAGENET

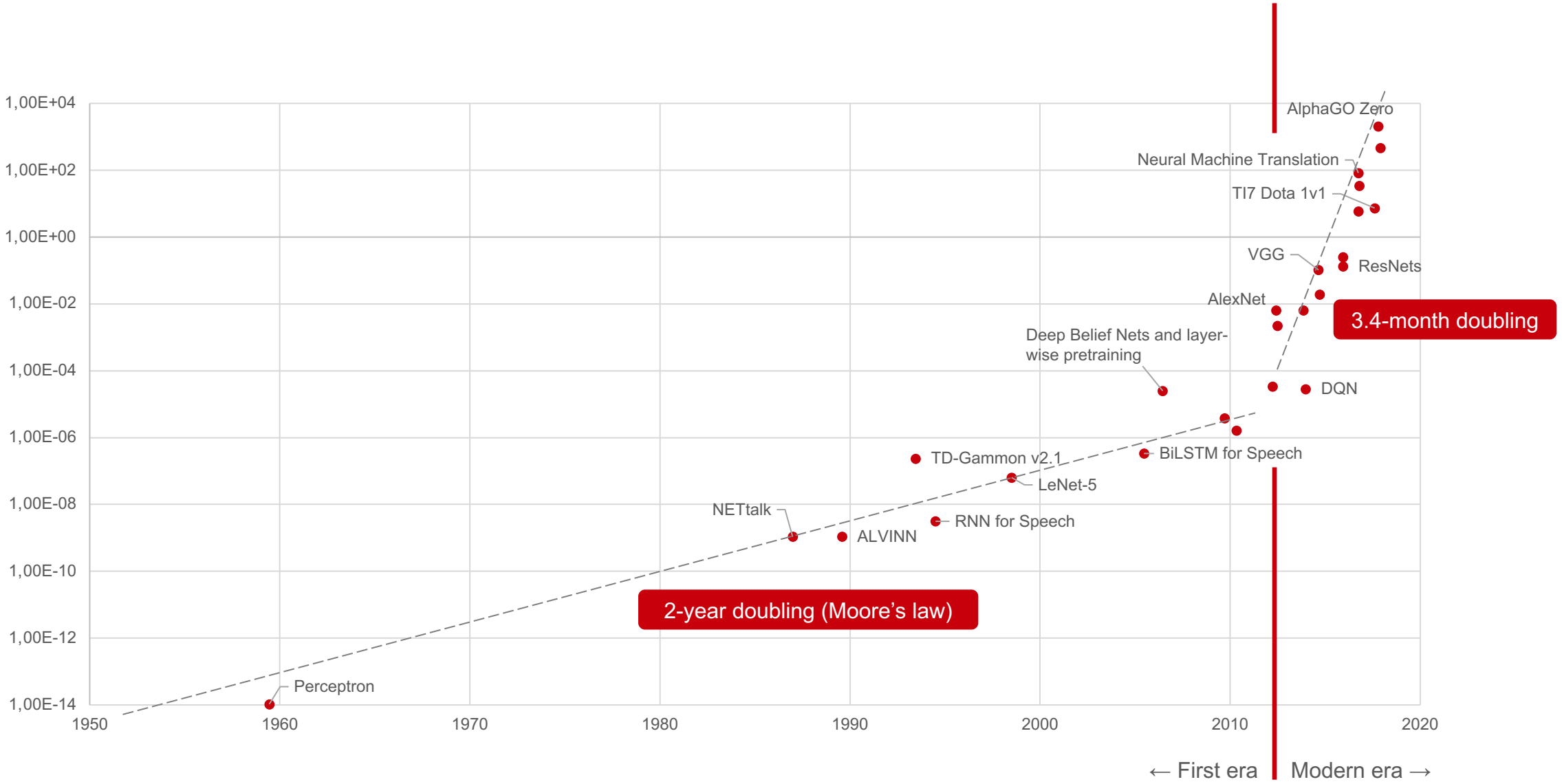


Neural networks beat human performance /2 [Russakovsky et al., 2015], [Papers With Code, 2020]

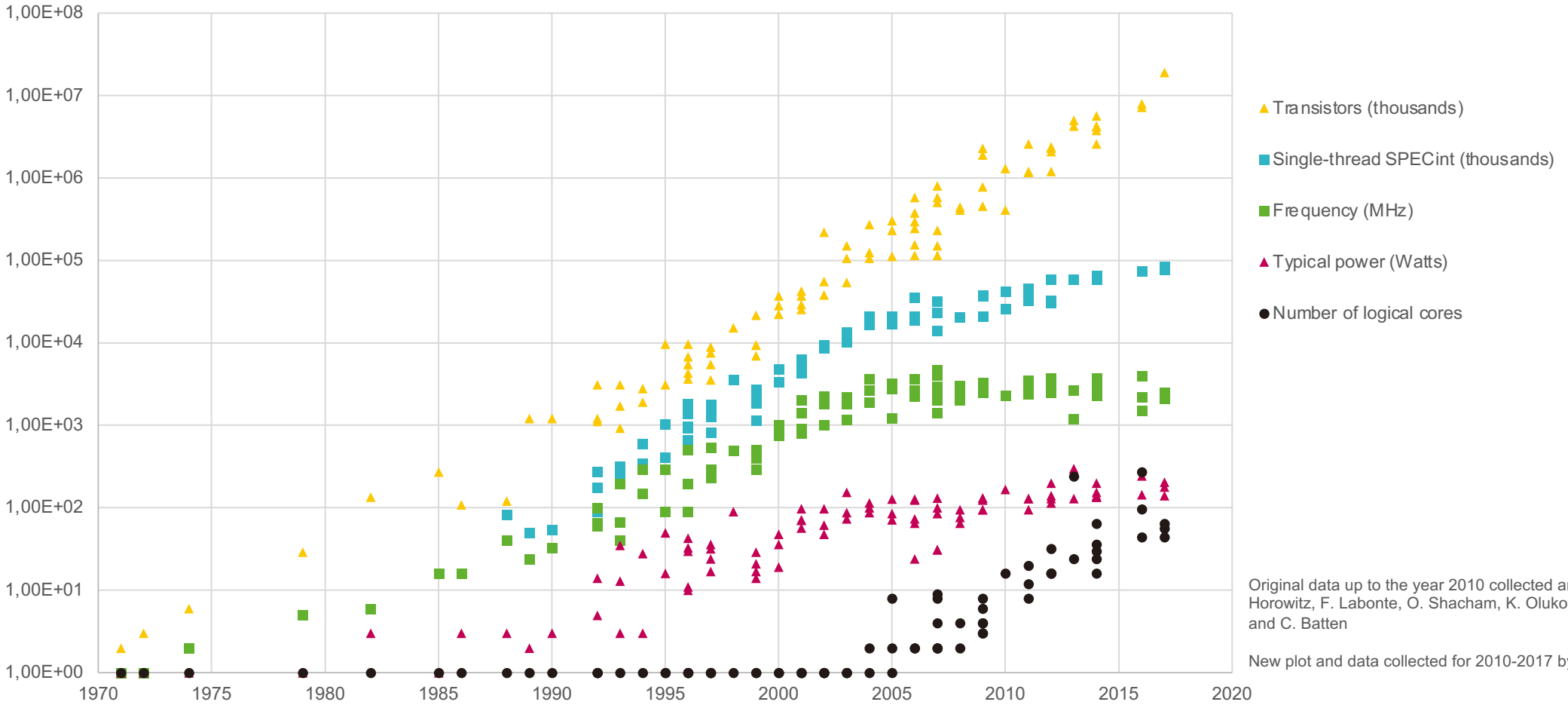
— **Example:** Image classification on ImageNet



Two distinct eras of compute usage in training AI systems [McCandlish et al., 2018], [Amodei et al., 2019]



Microprocessor trends [Brookes, 1986], [Sutter, 2005], [Rupp, 2015], [Rupp, 2018a], [Rupp, 2018b], [Hennessy et al., 2019]

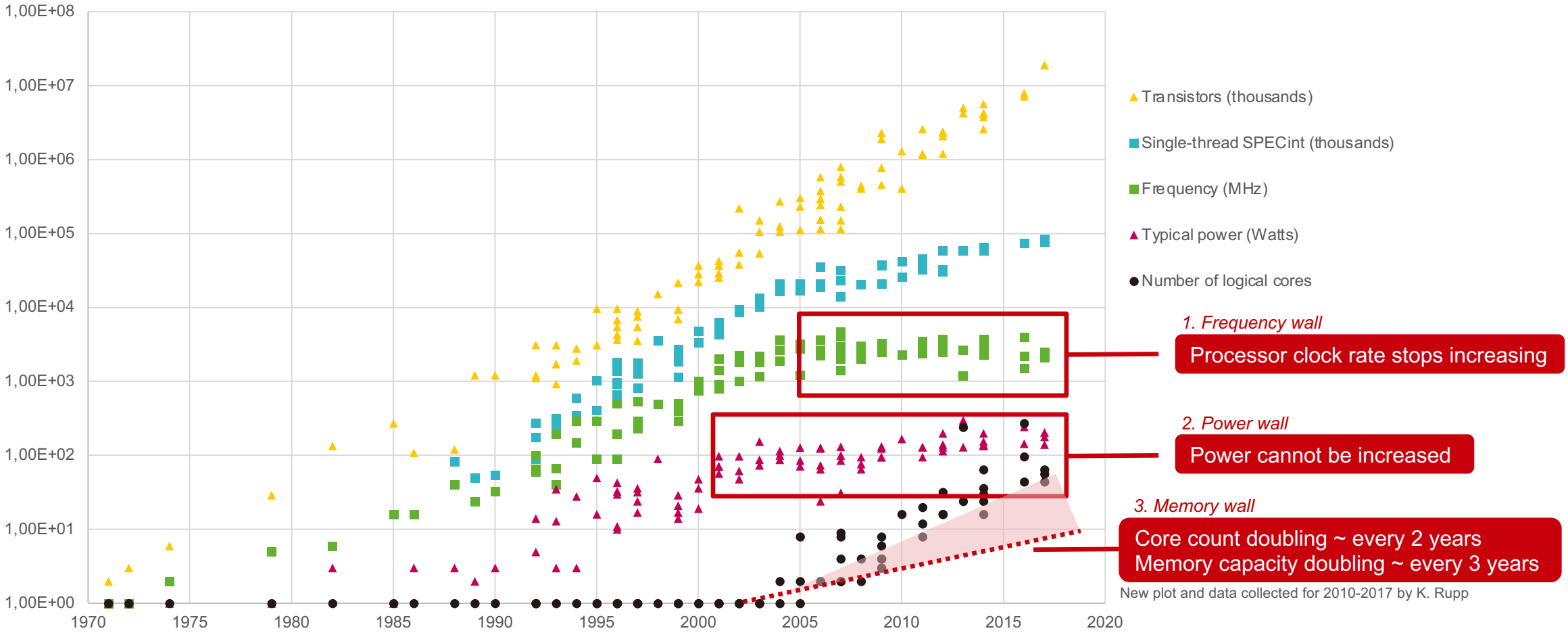


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten

New plot and data collected for 2010-2017 by K. Rupp

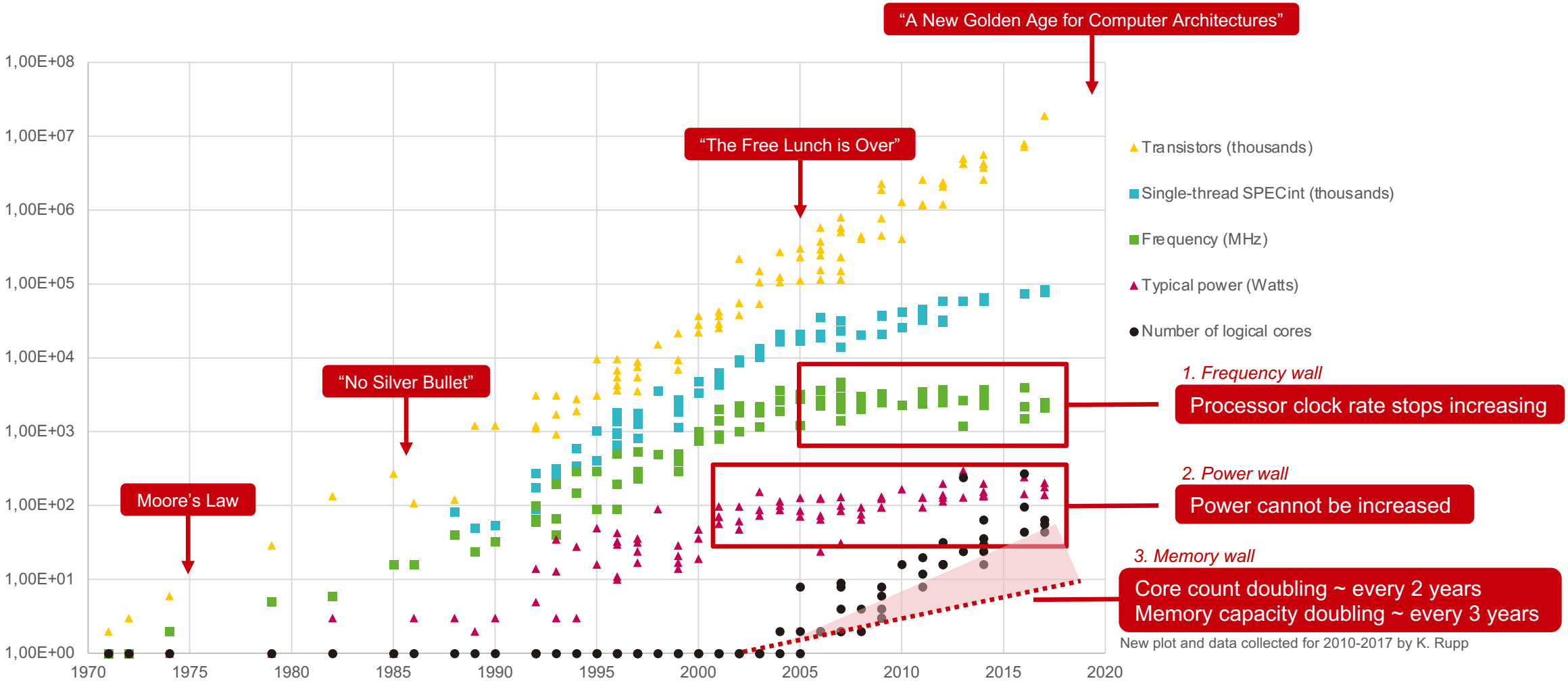
Microprocessor trends

[Brookes, 1986], [Sutter, 2005], [Rupp, 2015], [Rupp, 2018a], [Rupp, 2018b], [Hennessy et al., 2019]



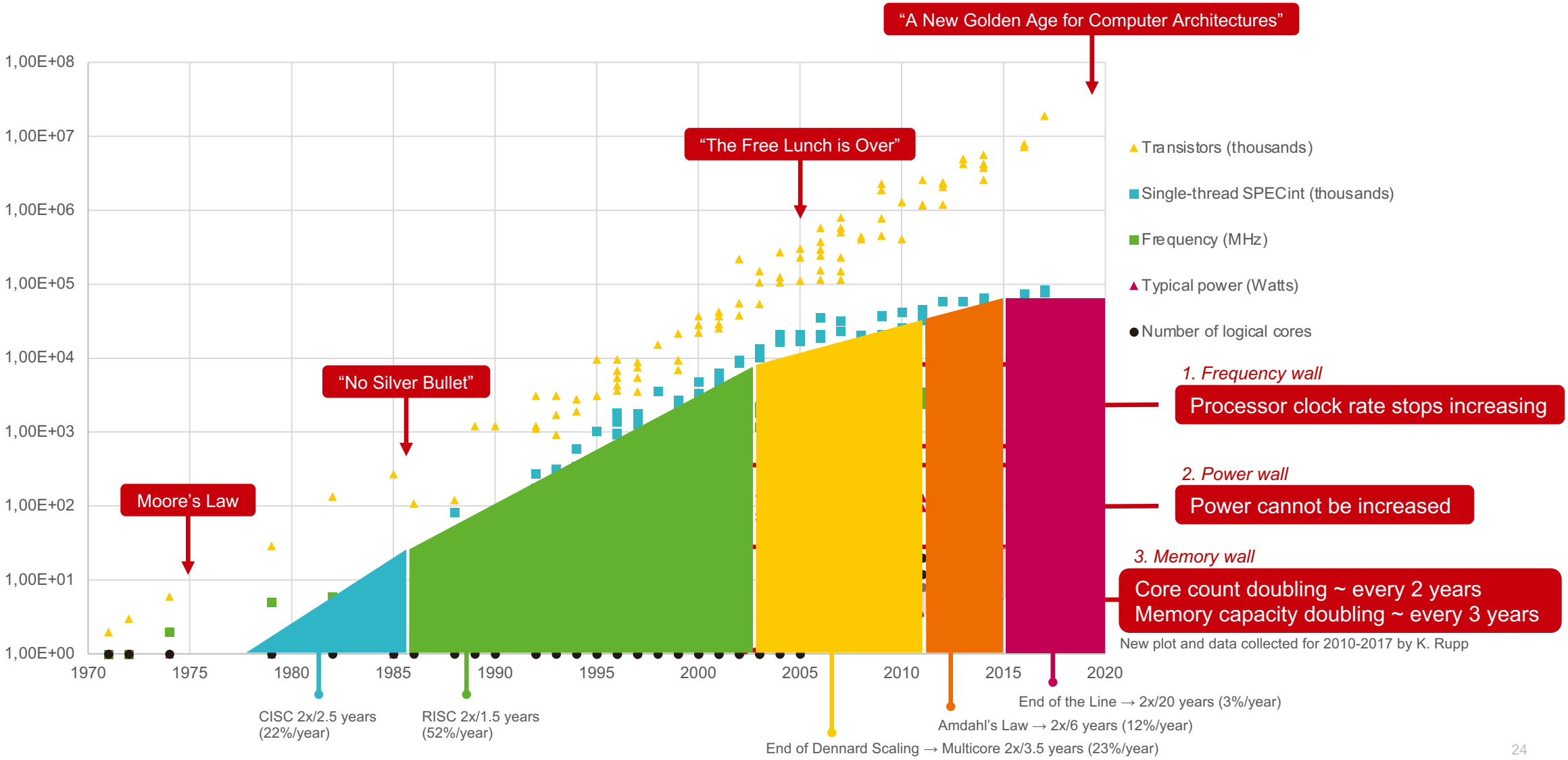
Microprocessor trends

[Brookes, 1986], [Sutter, 2005], [Rupp, 2015], [Rupp, 2018a], [Rupp, 2018b], [Hennessy et al., 2019]

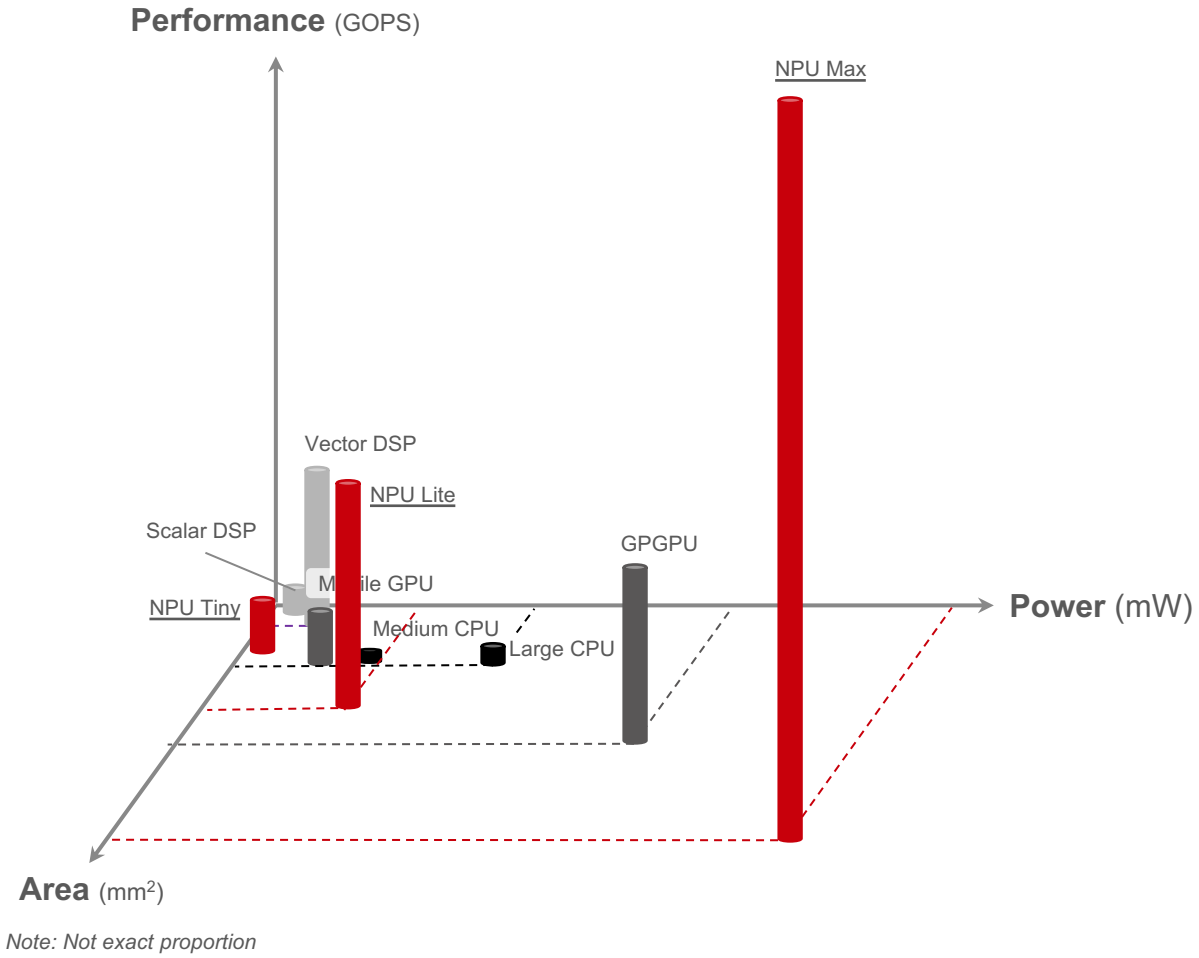


Microprocessor trends

[Brookes, 1986], [Sutter, 2005], [Rupp, 2015], [Rupp, 2018a], [Rupp, 2018b], [Hennessy et al., 2019]



Rich variety of computing architectures

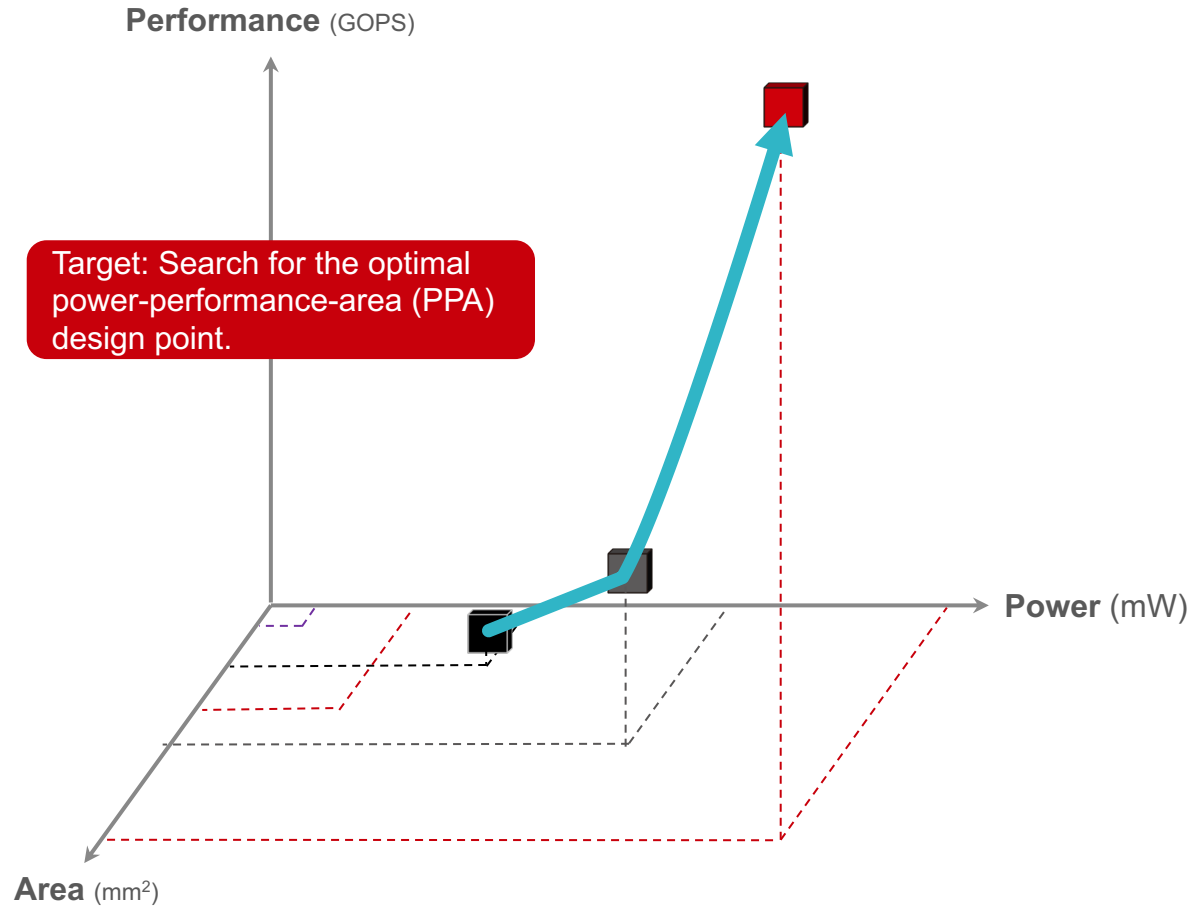


Note: Not exact proportion

- **Wide range of options to optimise for performance and efficiency:**
  - **Central processing unit (CPU)** executes general purpose applications (e.g. N-body methods, computational logic, map reduce, dynamic programming)
  - **General-purpose computing on graphics processing units (GPGPU)** accelerates compute intensive and time consuming applications for the CPU (e.g. dense linear algebra and sparse linear algebra)
  - **Digital signal processor (DSP)** accelerates signal processing for post camera operations (e.g. spectral methods)
  - **Image signal processor (ISP)** executes processing for camera sensor pipeline
  - **Vision processing unit (VPU)** accelerates machine vision tasks
  - **Network processor (NP)** accelerates packet processing
  - **Neural processing unit (NPU)** accelerates artificial intelligence applications (e.g. matrix-matrix multiplication, dot-products, scalar a times x plus y)

Each of these options represents different power, performance, and area trade-offs, which should be considered for specific application scenarios.

Rich variety of computing architectures

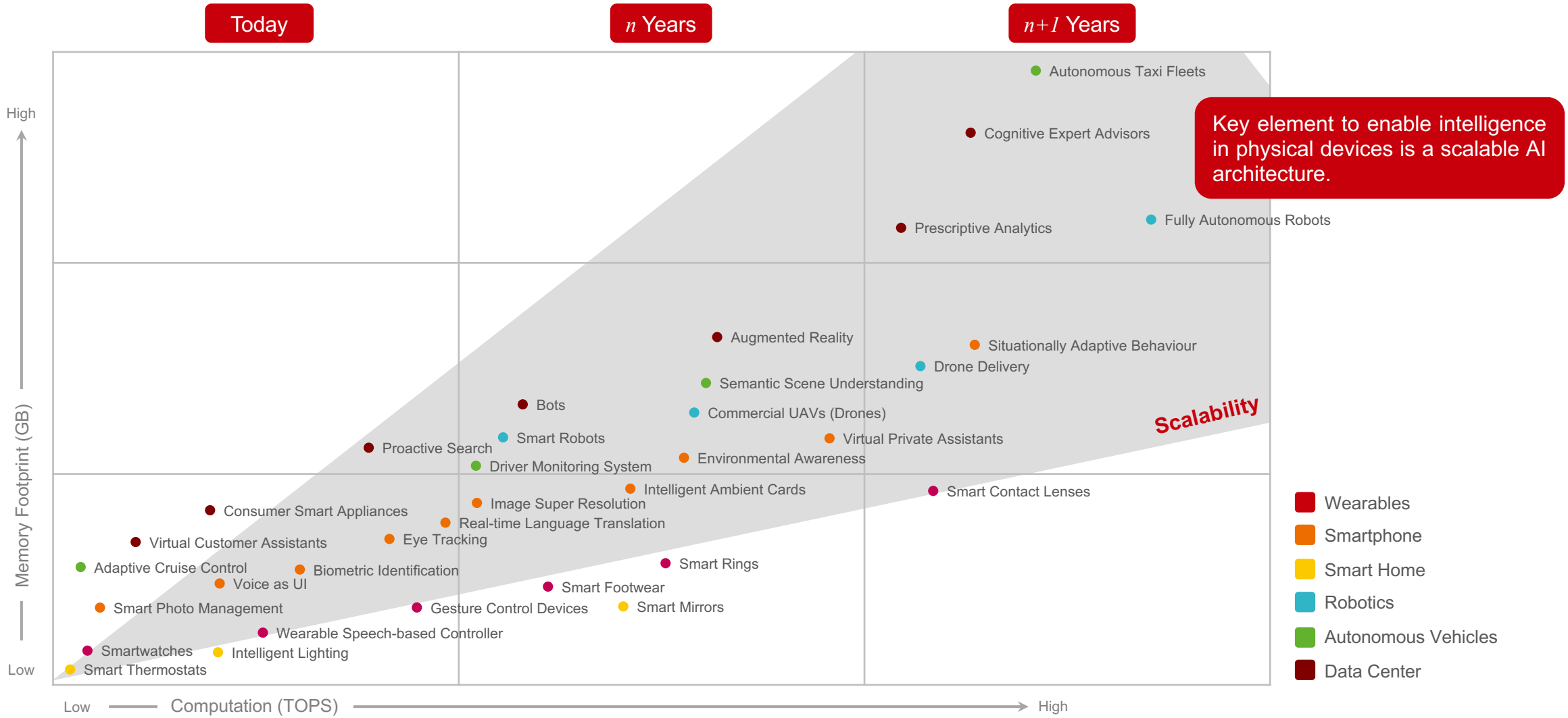


Note: Not exact proportion

- **Wide range of options to optimise for performance and efficiency:**
  - **Central processing unit (CPU)** executes general purpose applications (e.g. N-body methods, computational logic, map reduce, dynamic programming)
  - **General-purpose computing on graphics processing units (GPGPU)** accelerates compute intensive and time consuming applications for the CPU (e.g. dense linear algebra and sparse linear algebra)
  - **Digital signal processor (DSP)** accelerates signal processing for post camera operations (e.g. spectral methods)
  - **Image signal processor (ISP)** executes processing for camera sensor pipeline
  - **Vision processing unit (VPU)** accelerates machine vision tasks
  - **Network processor (NP)** accelerates packet processing
  - **Neural processing unit (NPU)** accelerates artificial intelligence applications (e.g. matrix-matrix multiplication, dot-products, scalar a times x plus y)

Each of these options represents different power, performance, and area trade-offs, which should be considered for specific application scenarios.

Ubiquitous and future AI computation requirements



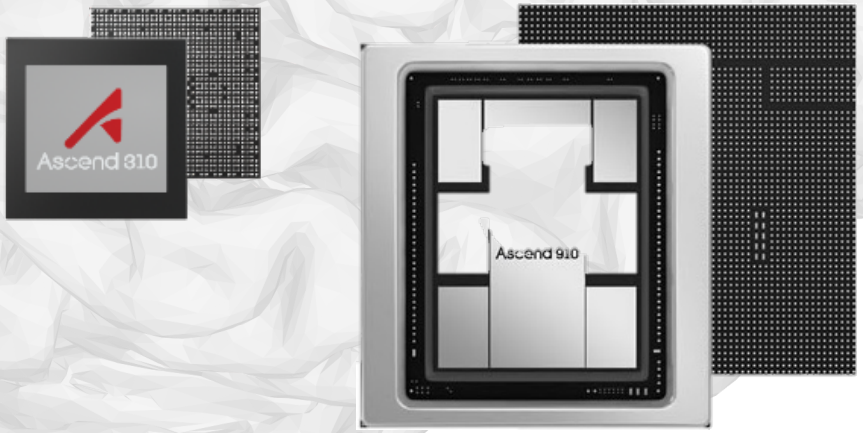
Focus on innovation, continuous dedication and backward compatibility

Inference **Training**

Ascend 310

AI SoC with ultimate efficiency

- Half precision (FP16): 8 TFLOPS
- Integer precision (INT8): 16 TOPS
- 16-channel full-HD video decoder: H.264/265
- 1-channel full-HD video encoder: H.264/265
- Max. power consumption: 8 W
- 12nm



Ascend 910

Highest compute density on a single chip

- Half precision (FP16): 256 TFLOPS
- Integer precision (INT8): 512 TOPS
- 128-channel full-HD video decoder: H.264/265
- Max. power consumption: 350 W
- 7nm

"Once a technology becomes digital—that is, once it can be programmed in the ones and zeros of computer code—it hops on the back of Moore's law and begins accelerating exponentially."

– Peter H. Diamandis & Steven Kotler, The Future Is Faster Than You Think

Focus on innovation, continuous dedication and backward compatibility

Atlas 200

AI Accelerator Module

- 16 TOPS of INT8
- 16-channel HD video real-time analytics, JPEG decoding
- 4 GB/8 GB memory, PCIe 3.0 x4 interface
- Operating temperature: -25°C to +80°C

52 mm x 38 mm x 10 mm

Atlas 300

AI Accelerator Card

- 64 TOPS of INT8 @ 67 W
- 32 GB memory
- 64-channel HD video real-time analytics
- Standard half-height half-length PCIe card form factor, applicable to general-purpose servers

reddot award 2019 winner

Atlas 500

AI Edge Stations

- 16 TOPS of INT8
- 25–40 W
- Wi-Fi & LTE
- 16-channel HD video real-time analytics
- Fanless design, -40°C to +70°C environments

reddot award 2019 winner

Atlas 800

Deep Learning System

- Plug-and-play installation
- Ultimate Performance
- Integrated Management

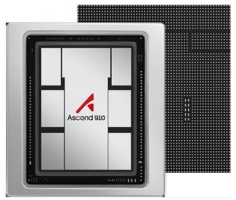
Ascend 310

AI SoC with ultimate efficiency

- Half precision (FP16): 8 TFLOPS
- Integer precision (INT8): 16 TOPS
- 16-channel full-HD video decoder: H.264/265
- 1-channel full-HD video encoder: H.264/265
- Max. power consumption: 8 W
- 12nm

Ascend 910

Highest compute density on a single chip



- Half precision (FP16): 256 TFLOPS
- Integer precision (INT8): 512 TOPS
- 128-channel full-HD video decoder: H.264/265
- Max. power consumption: 350 W
- 7nm

Atlas 900 AI Cluster

The pinnacle of computing power

- Thousands of Ascend 910 AI processors
- High-speed interconnection
- Delivers up to 256 to 1024 PetaFLOPS at FP16
- Can complete model training based on ResNet-50 within 59.8 seconds
- 15% faster than the second-ranking product
- Faster AI model training with images and speech

Atlas 200 DK

Quickly build development environments in 30 minutes

- 16 TOPS of INT8 @ 24 W
- 1 USB type-C, 2 camera interfaces, 1 GE port, 1 SD card slot
- 4 GB/8 GB memory

Kunpeng 920

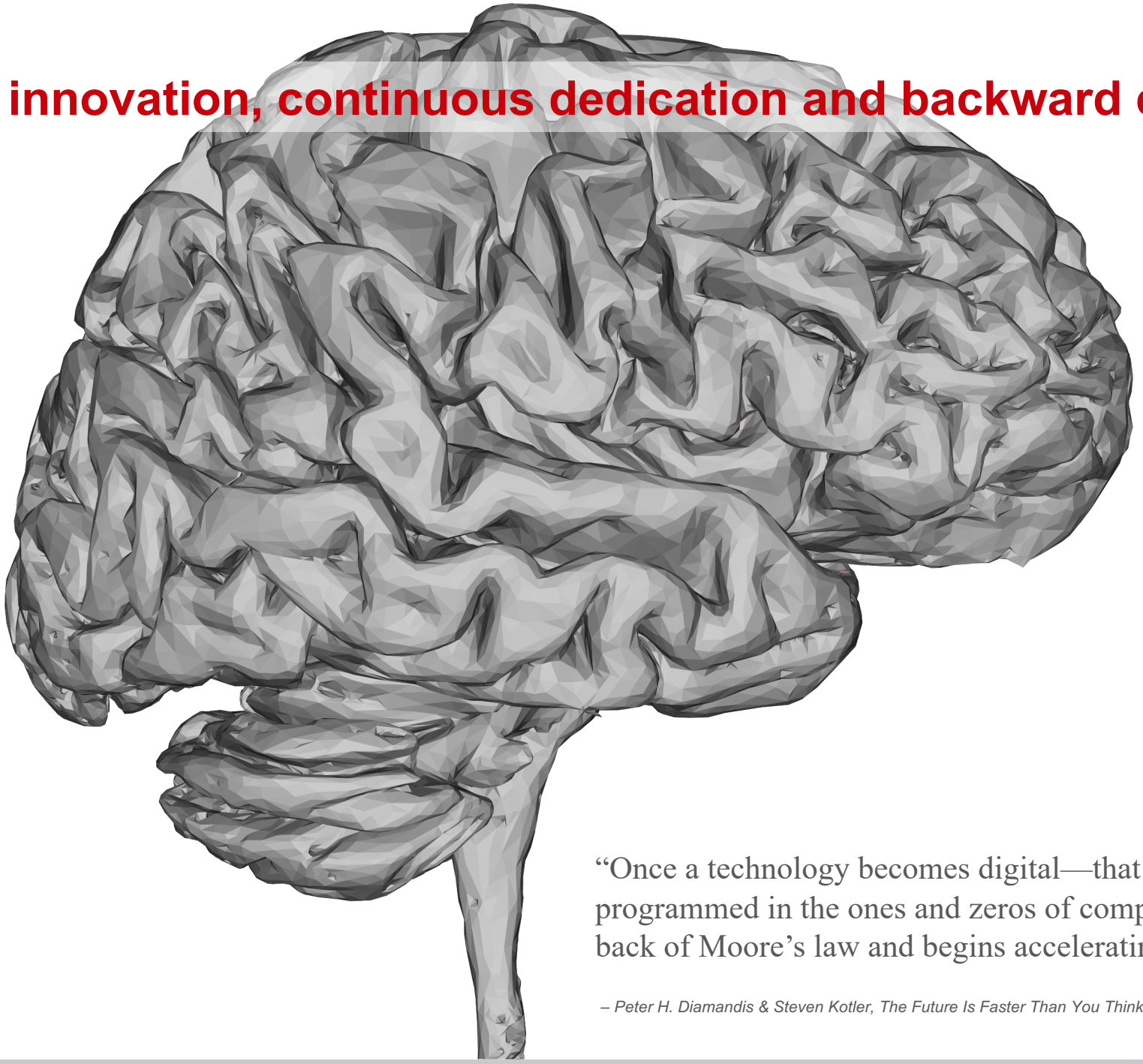
The industry's highest-performance ARM-based server CPU

- ARM v8.2-architecture
- up to 64 cores, 2.6 GHz
- 8 DDR4 memory channels
- PCIe 4.0 and CCIX
- Integrated 100GE LOM and encryption and compression engines
- Supports 2- or 4-socket interconnects

Storage-intensive ——— **Computing-intensive**

5290	5280	2280	1280	2480	X6000
4U 72-drive storage model	4U 40-drive storage model	2U 2S balanced model	1U 2S high-density model	2U 4S high-performance model	2U 4-node high-density model

Focus on innovation, continuous dedication and backward compatibility

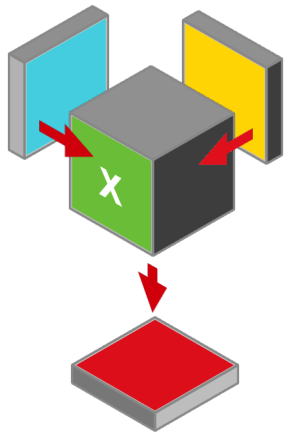


"Once a technology becomes digital—that is, once it can be programmed in the ones and zeros of computer code—it hops on the back of Moore's law and begins accelerating exponentially."

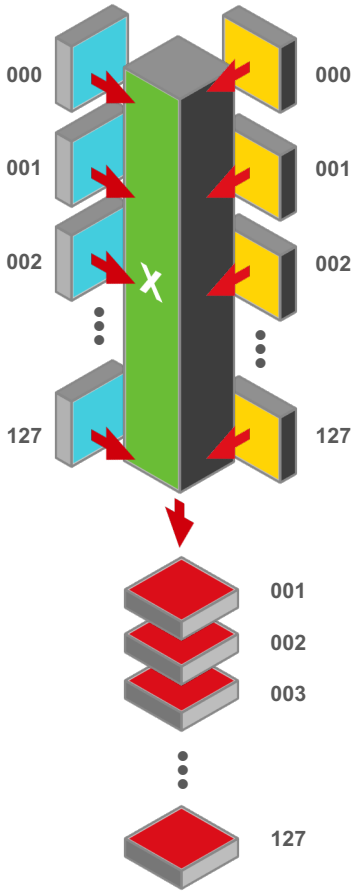
– Peter H. Diamandis & Steven Kotler, The Future Is Faster Than You Think

Building blocks and compute intensity

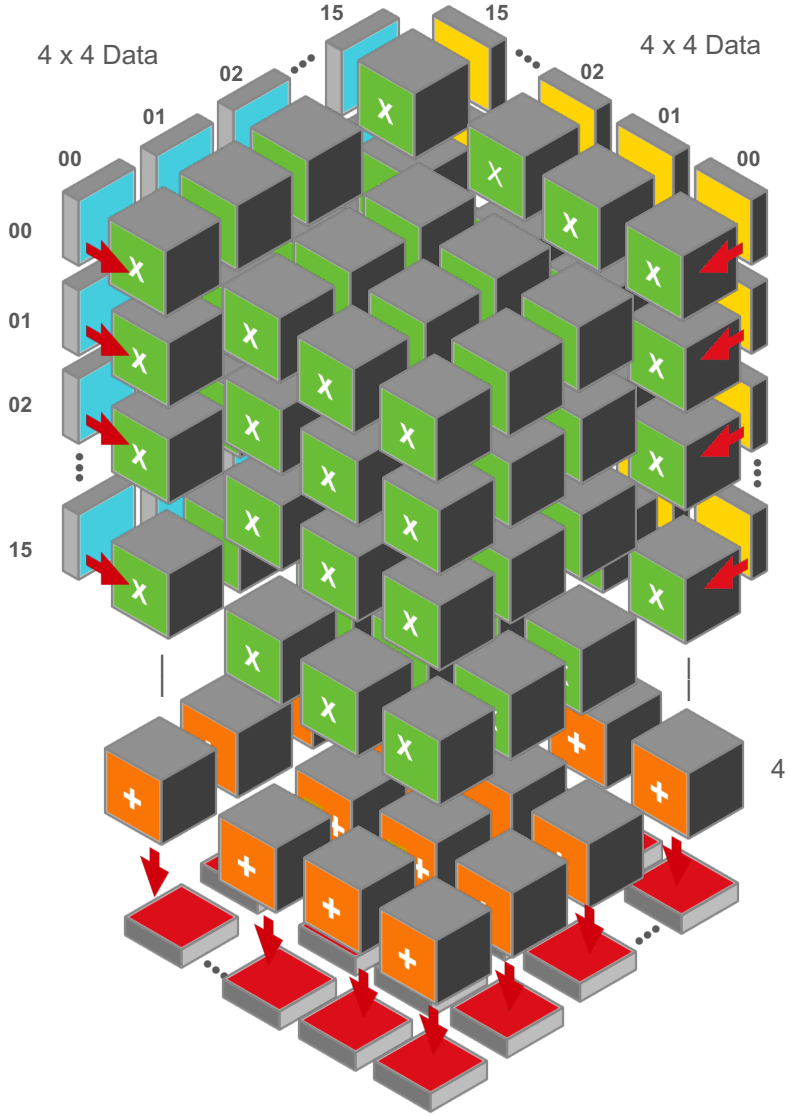
Scalar Unit
Full flexibility in computation



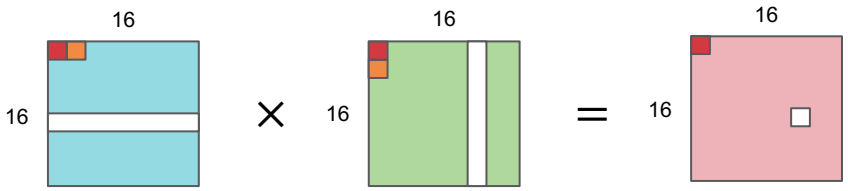
Vector Unit
Rich and efficient operations



Cube Unit
High intensity computation



Advantages of special compute units




```
float a[16][16], b[16][16], c[16][16];
```

Scalar:

```
for (int i=0; i<16; i++) {
    for (int j=0; j<16; j++) {
        for (int k=0; k<16; k++) {
            c[i][j] += a[i][k] * b[k][j];
        }
    }
}
```

$Cycles = 16 * 16 * 16 * 2 = 8192$
$Data\ per\ cycle = Rd\ 2;\ Wr\ 1$

Vector:

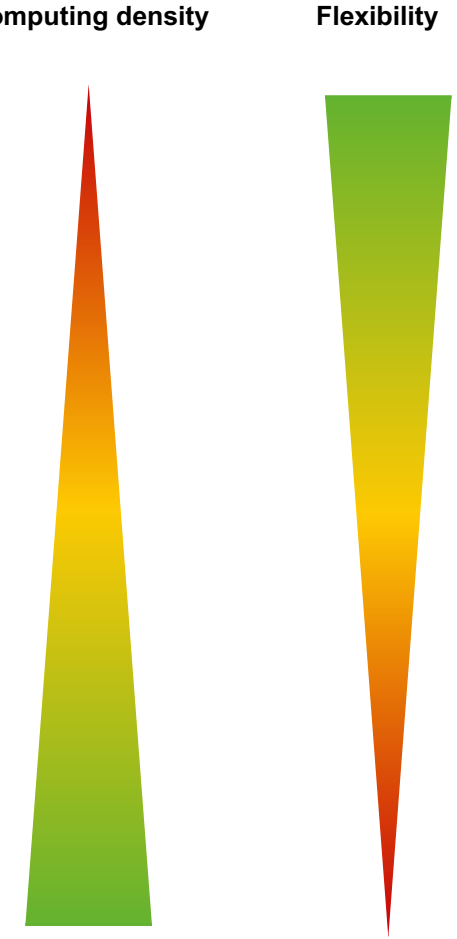
```
for (int i=0; i<16; i++) {
    for (int j=0; j<16; j++) {
        c[i][j] = a[i][:] *+ b[:][j];
    }
}
```

$Cycles = 16 * 16 = 256$
$Data\ per\ cycle = Rd\ 2 * 16;\ Wr\ 16$

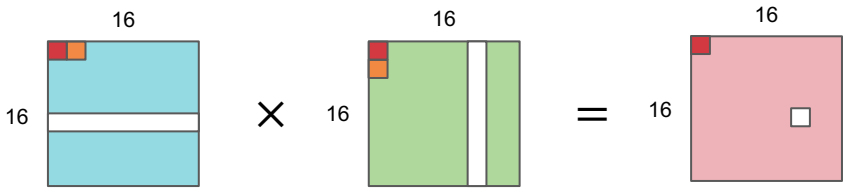
Cube:

```
c[:][:] = a[:][:] X b[:][:]
```

$Cycles = 1$
$Data\ per\ cycle = Rd\ 2 * 16 * 16;\ Wr\ 16*16$



Advantages of special compute units




```
float a[16][16], b[16][16], c[16][16];
```

Scalar:

```
for (int i=0; i<16; i++) {
    for (int j=0; j<16; j++) {
        for (int k=0; k<16; k++) {
            c[i][j] += a[i][k] * b[k][j];
        }
    }
}
```

$Cycles = 16 * 16 * 16 * 2 = 8192$
$Data\ per\ cycle = Rd\ 2;\ Wr\ 1$

Vector:

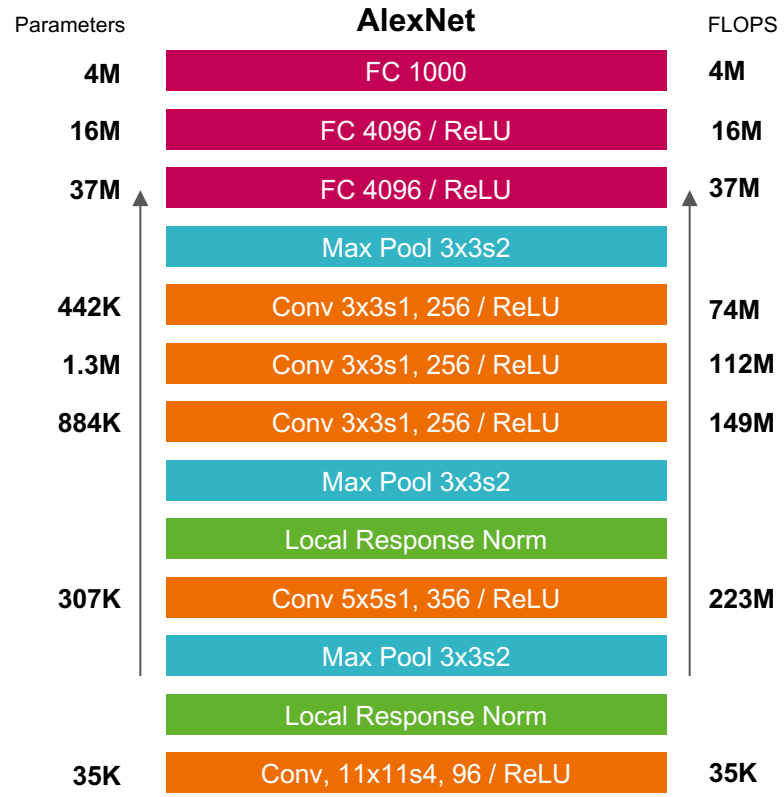
```
for (int i=0; i<16; i++) {
    for (int j=0; j<16; j++) {
        c[i][j] = a[i][:] *+ b[:][j];
    }
}
```

$Cycles = 16 * 16 = 256$
$Data\ per\ cycle = Rd\ 2 * 16;\ Wr\ 16$

Cube:

```
c[:][:] = a[:][:] X b[:][:]
```

$Cycles = 1$
$Data\ per\ cycle = Rd\ 2 * 16 * 16;\ Wr\ 16*16$



Number of parameters and floating point operations per second (FLOPS) for each layer of the AlexNet artificial intelligence model.

99% of the computations are matrix-matrix multiplications	Typical CNN networks		
	AlexNet	VGG16	Inception-v3
Model memory (MB)	> 200	> 500	90-100
Parameter count (Million)	60	138	23.2
Computation amount (Million)	720	15300	5000

Upscaling and colourisation of video footage

Source: https://www.reddit.com/r/interestingasfuck/comments/idbtrg/i_upscaled_and_colorized_the_footage_about_the/

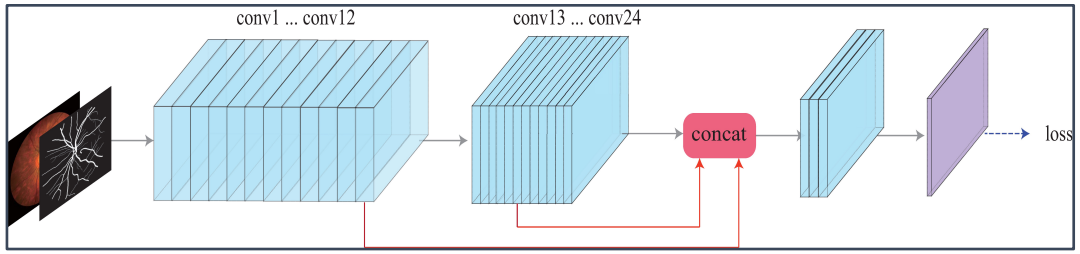
Despina Manaki is the earliest-born person on film. In 1905, when she was 114 years old (born 1791), she was filmed by her grandsons, Yanaki and Milton Manaki, cinema pioneers in the Balkans and the Ottoman Empire. This video shows the full force of artificial intelligence restoring old footages.

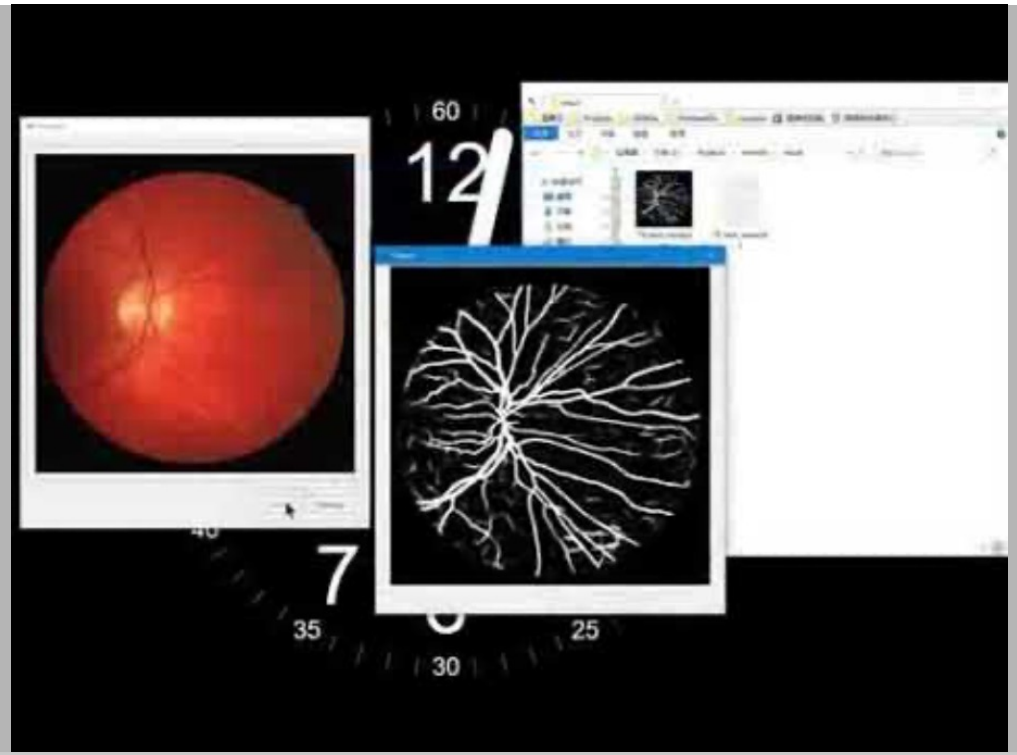
Retinal blood vessel segmentation in the eyeground

南開大學 Nankai University

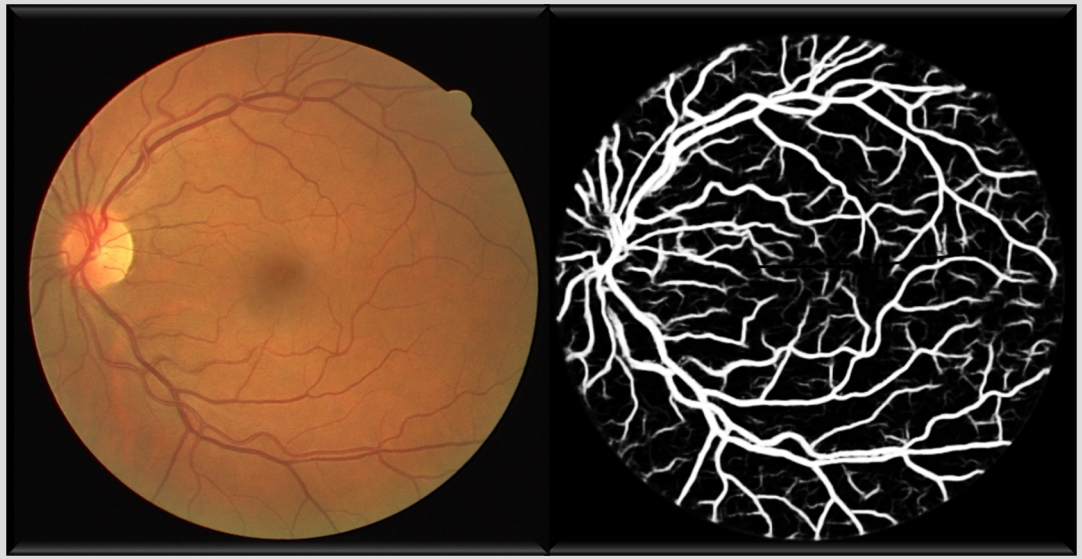
- The fundus retinal blood vessel segmentation application was developed for the Atlas 200 DK inference system, in partnership with the Nankai University, led by Professor Li Tao of Intelligent Computing System Research Office .
- This project makes full use of the neural network computing power of the Atlas 200 DK system to segment the fundus vessels in real-time.
- The total inference time of **20 pictures is 761.8 milliseconds**, and the average inference time of one image is 38 milliseconds.

An overview of the vascular segmentation model

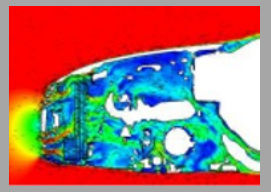
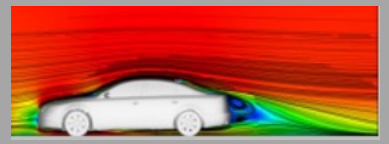
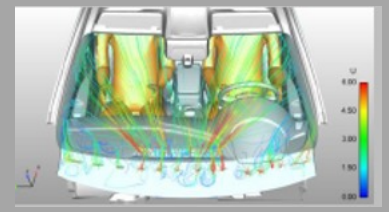




Before and after



Things we can do in science with AI technologies now /1

	Underhood air flow	External aerodynamics	HVAC
			
Standard case	3 stationary points - vmax, 250 km/h - Idle - Mountain 30 km/h	- 140 km/h - With and without underhood flow	- Defrost - Pull-down - Heater mode
Target values	- Cooling air mass flow - p distribution	- Drag - Lift - cp distribution	- Air distribution - Velocity distribution - Δp
Turnaround times	5-8 h/over-night	2-3 days	Few hours/over-night

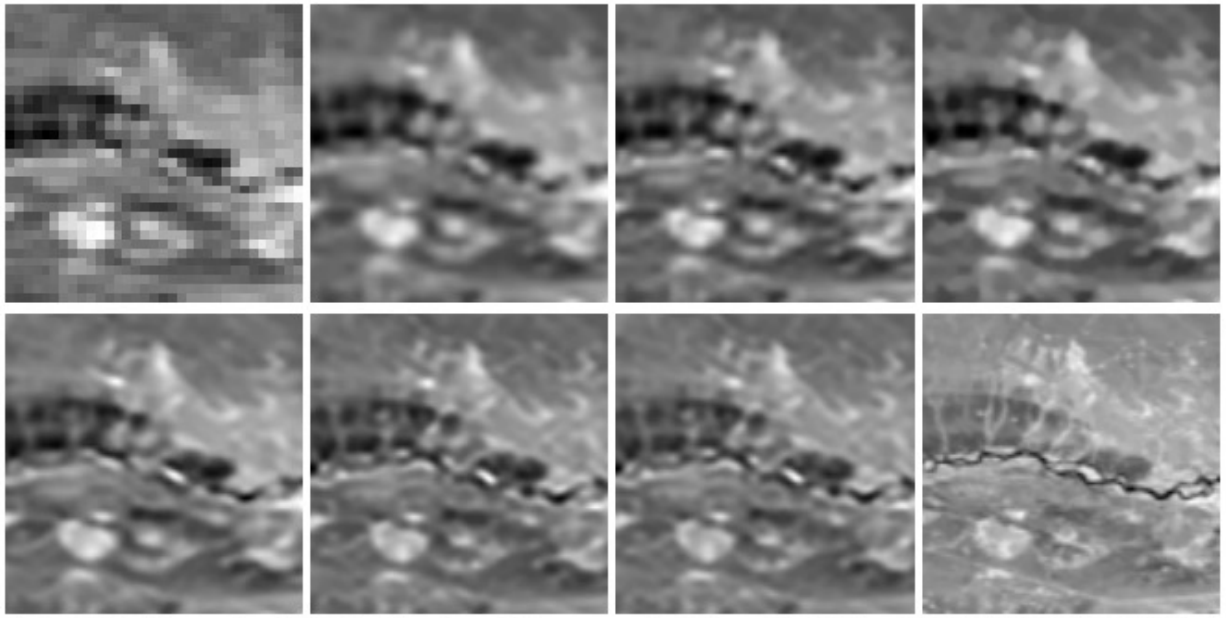
Things we can do in science with AI technologies now /2

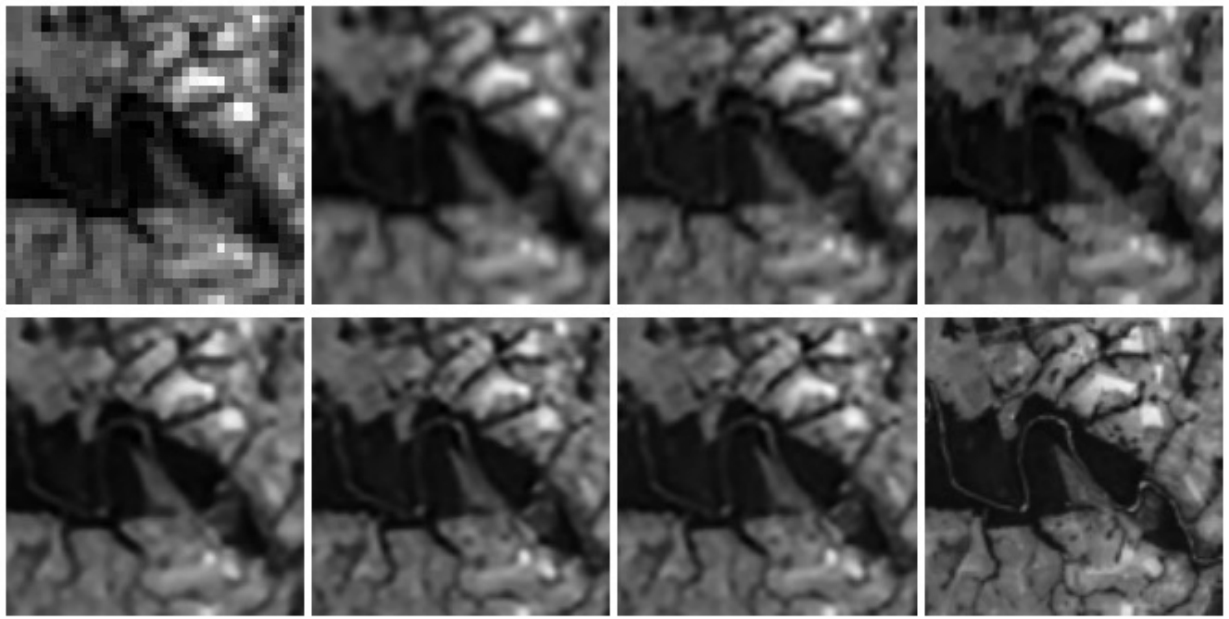
DeepSUM: Deep neural network for Super-resolution of Unregistered Multitemporal images

Andrea Bordone Molini, Diego Valsesia, Giulia Fracastoro, and Enrico Magli

***Abstract*—Recently, convolutional neural networks (CNN) have been successfully applied to many remote sensing problems. However, deep learning techniques for multi-image super-resolution from multitemporal unregistered imagery have received little**

versions of the same scene to be combined by means of MISR techniques, where the reconstruction of high spatial-frequency details takes full advantage of the complementary information coming from different observations of the same scene.





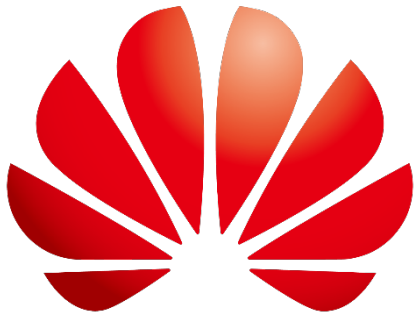
In ten years …

1. Learned AI model begin to replace data
2. The discovery process of experiments is dramatically refactored
3. We will persue many questions semi-autonomously
4. HPC simulations and AI technologies will merge
5. AI algorithms will contribute to advancing theories
6. AI technologies are becoming a common part of scientific laboratories activities

→ AI technologies will help to solve new challenges

→ AI technologies will become a new partner in simulation and data analytics

→ AI technologies will generate new computing architectures, new software environments, new policies, new user communities and new ways of dissemination.



HUAWEI

Stay safe — stay healthy