Arm **2018**

A Vision for Destruction of Post-Moore Disruption: How I Learned To Stop Worrying and Love Non-Von Neumann

Presented by: Jonathan Beard

UCAR SEA, Boulder, CO

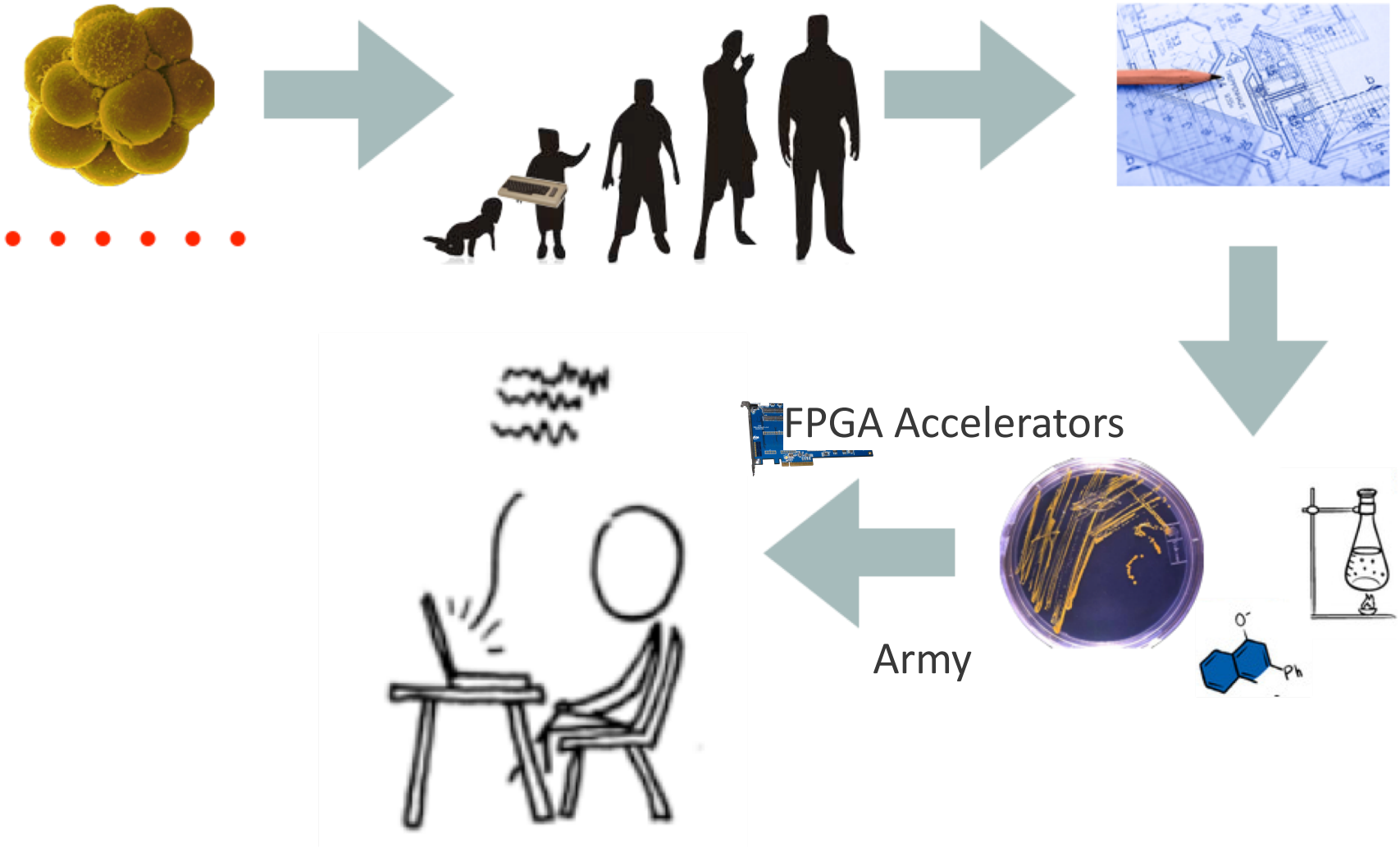
3 April

Twitter: @jonathan_beard

Disclaimer

- This is not a product presentation.
- Any changes to architecture must go through the standard architecture review board process.
- Nothing within this presentation should be taken as a direction of future architecture or product plans for Arm Inc/Ltd.
- Please ask questions related to the presentation, this should be a fun presentation!!

About Me

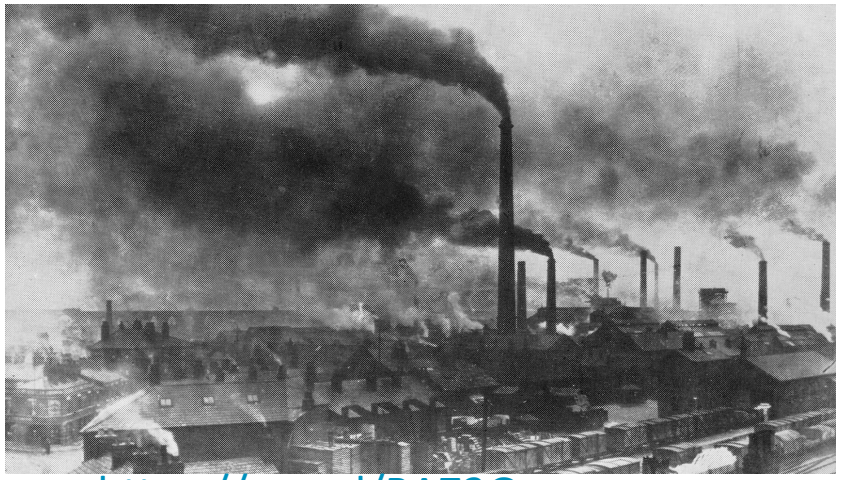


Natural Selection

Pale Peppered Moth

src: https://goo.gl/QJbLqG

19th Century London



src: https://goo.gl/BA72Gu

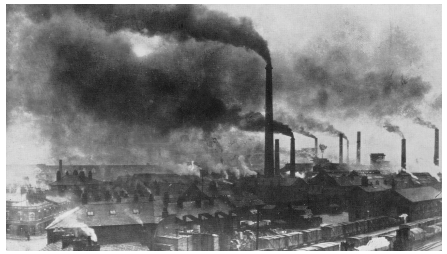
Bird Food

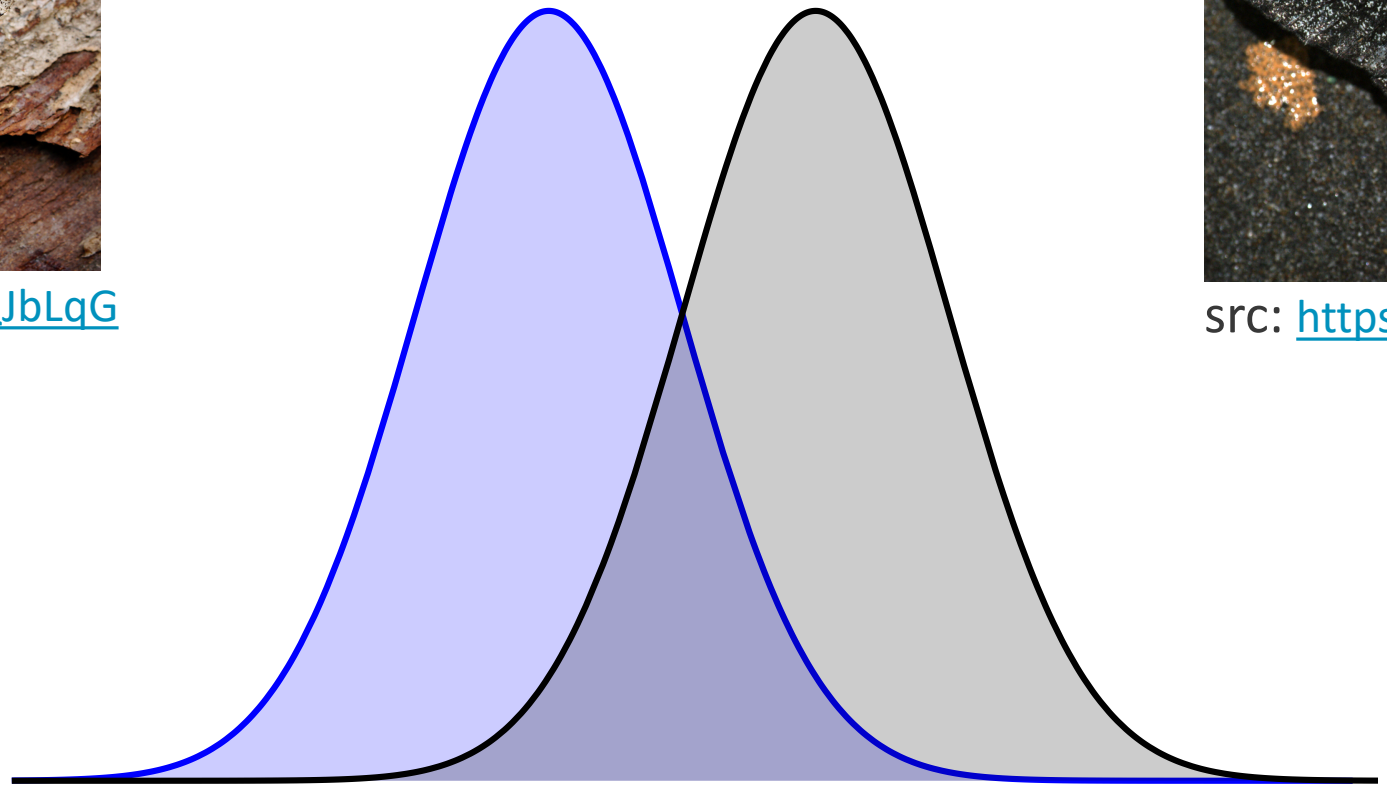
arm

Natural Selection

Pale Peppered Moth

src: https://goo.gl/QJbLqG





~~Pale~~ Peppered Moth

src: https://goo.gl/1BBZVG

arm

Natural Selection

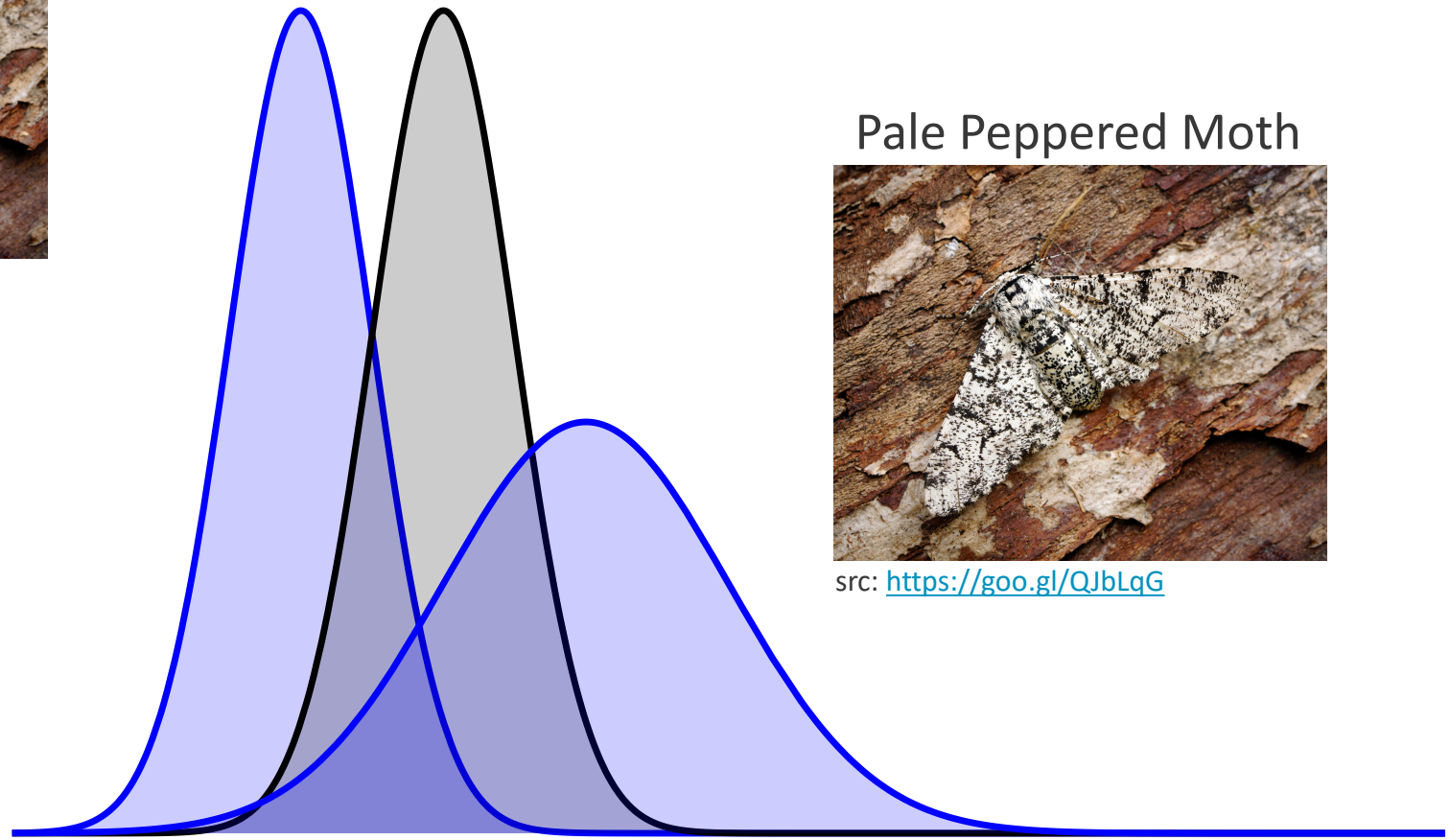
Pale Peppered Moth

src: https://goo.gl/QJbLqG

~~Pale~~ Peppered Moth

src: https://goo.gl/1BBZVG

src: https://goo.gl/yna3tH



Pale Peppered Moth

src: https://goo.gl/QJbLqG

arm

Selection on Computers

src: https://goo.gl/1dQoCZ

Speed

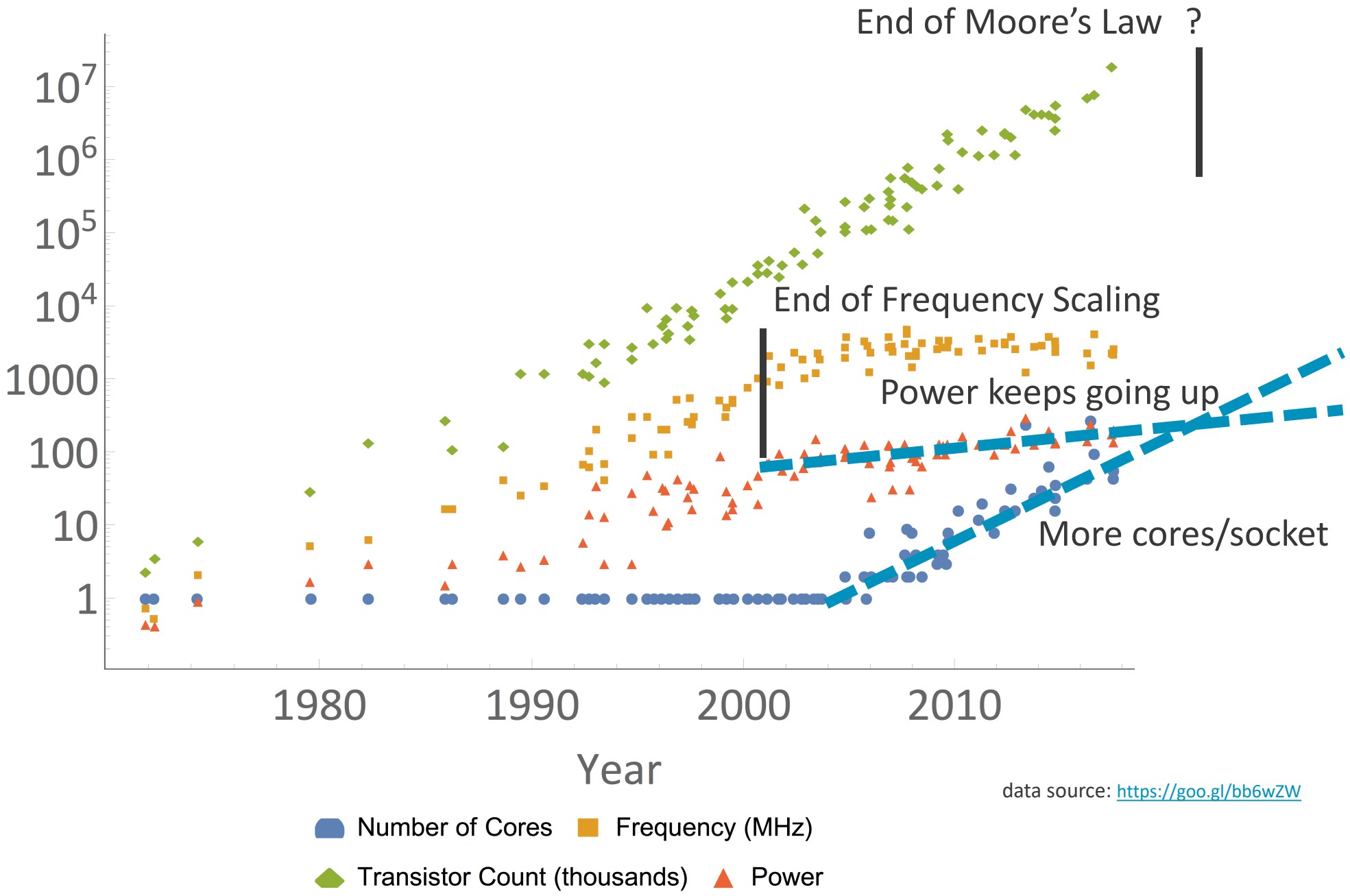
$$

Energy Consumption

Ease of Use

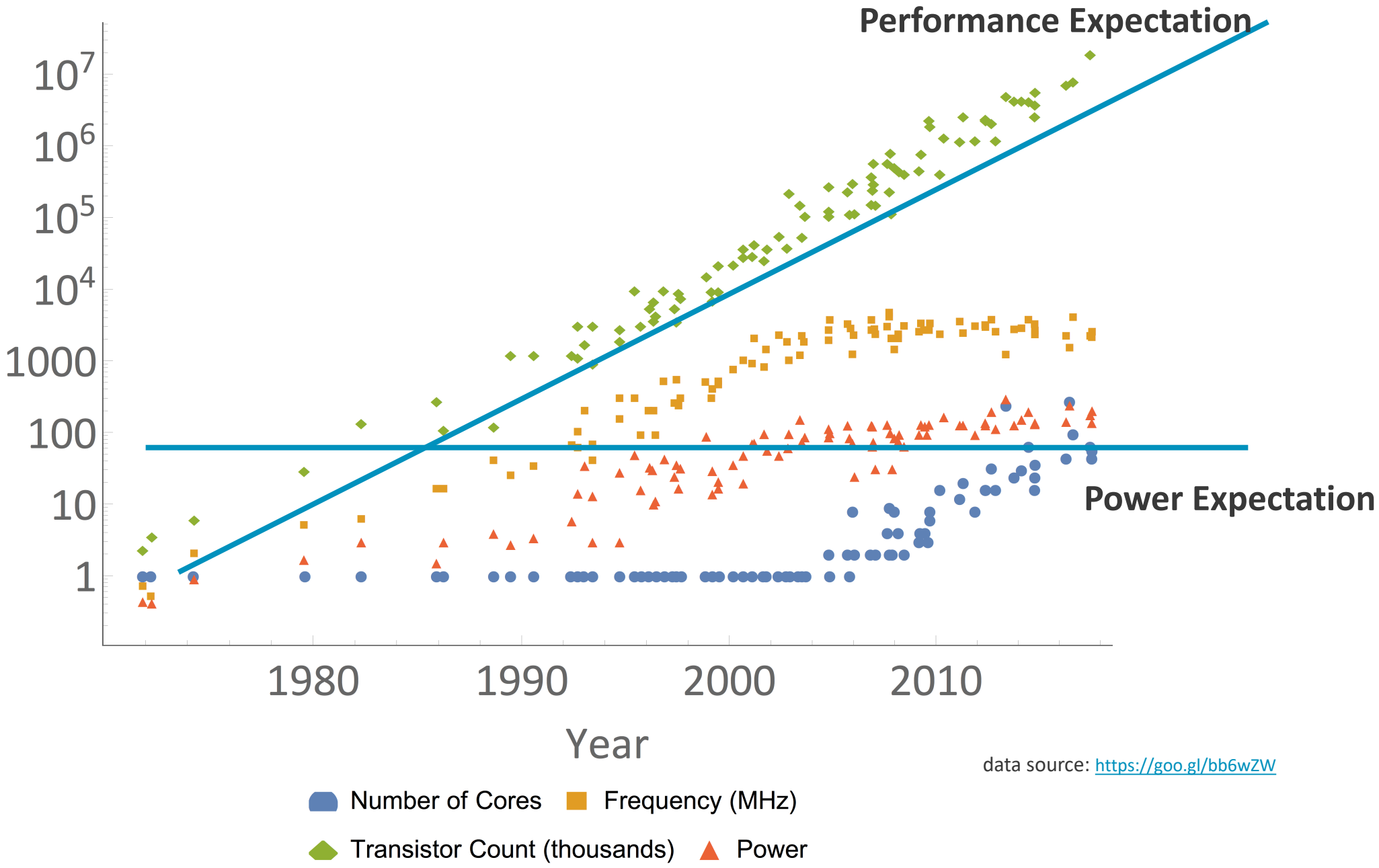
Software Availability

Trends



data source: https://goo.gl/bb6wZW

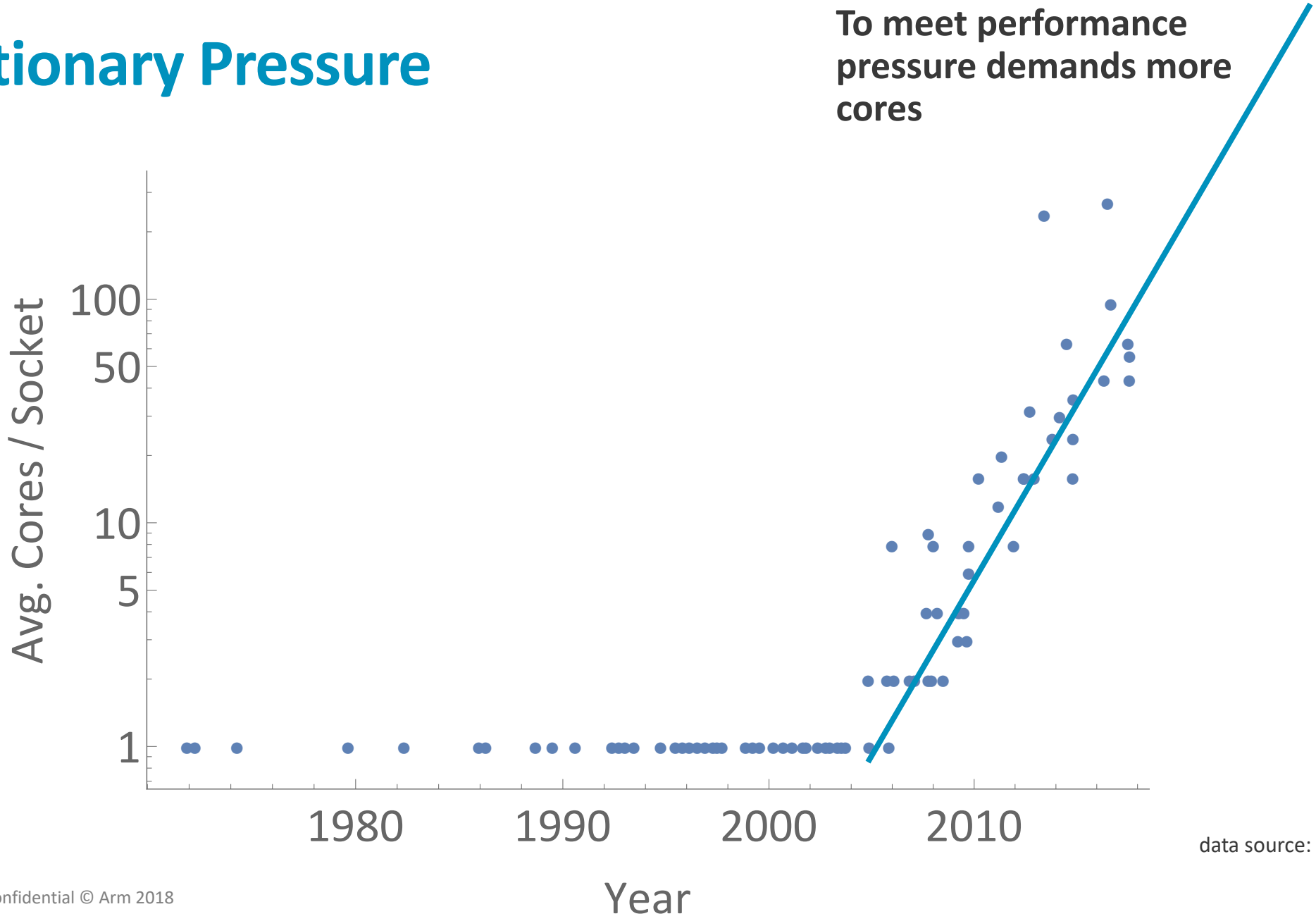
Trends



data source: https://goo.gl/bb6wZW

Evolutionary Pressure

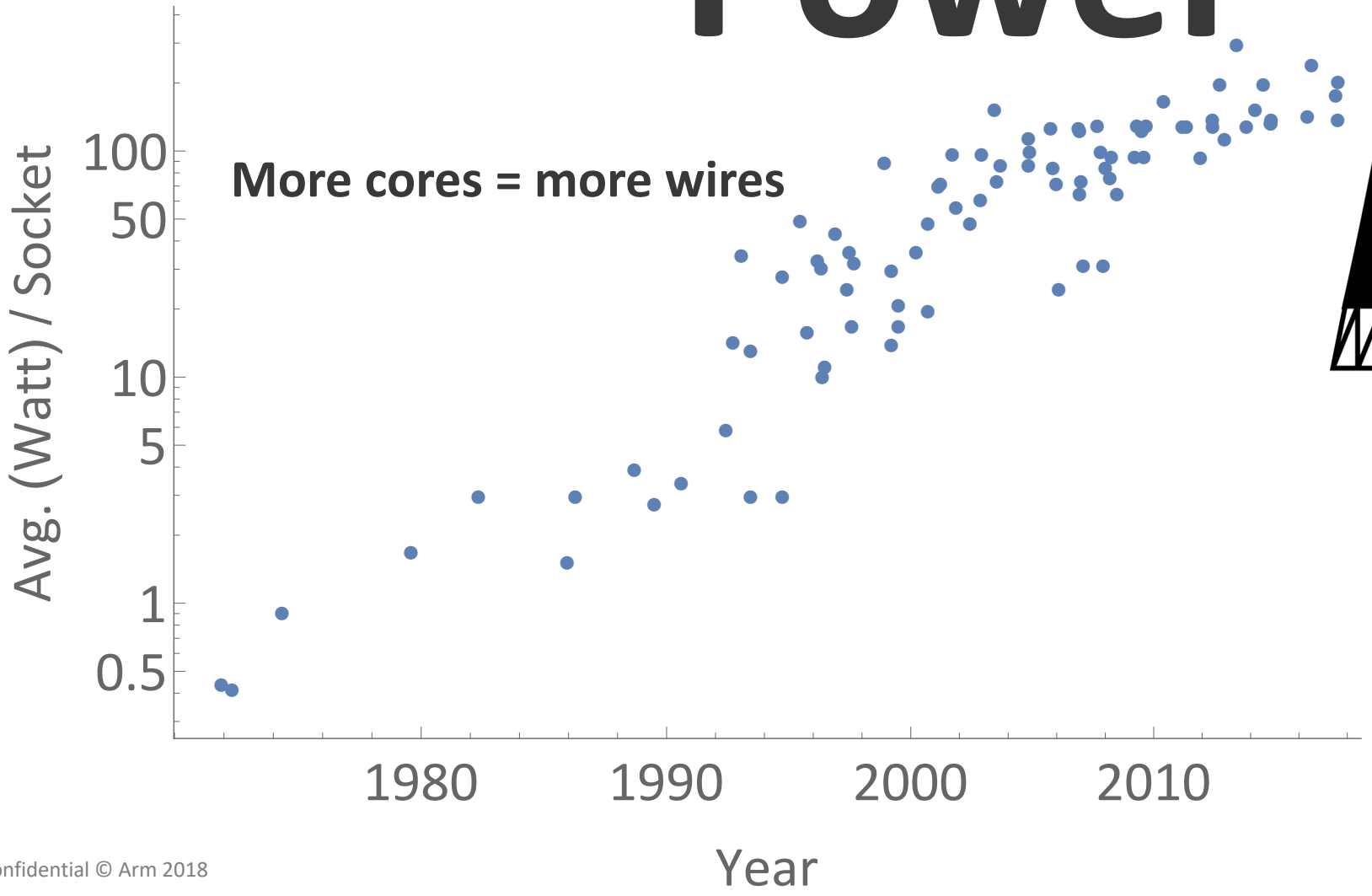
To meet performance pressure demands more cores

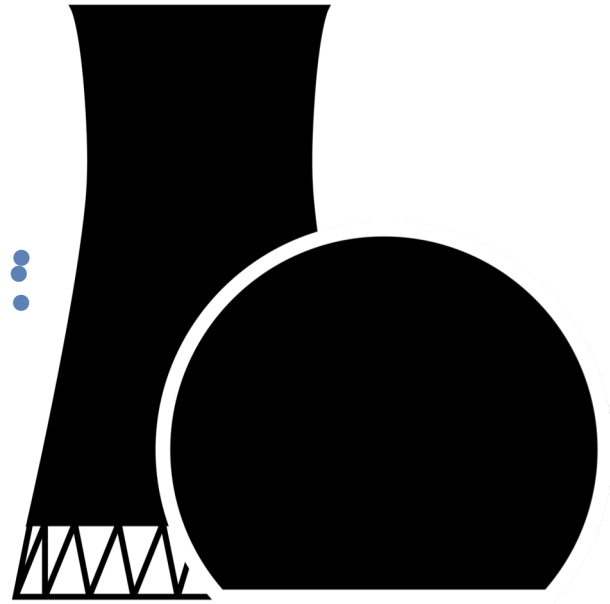


data source: https://goo.gl/bb6wZW

New Selective Pressure

Power



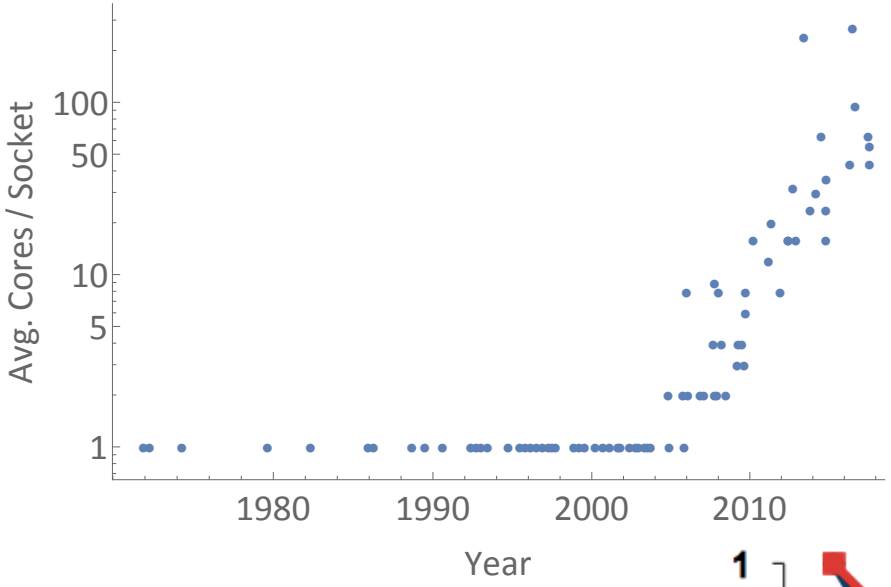


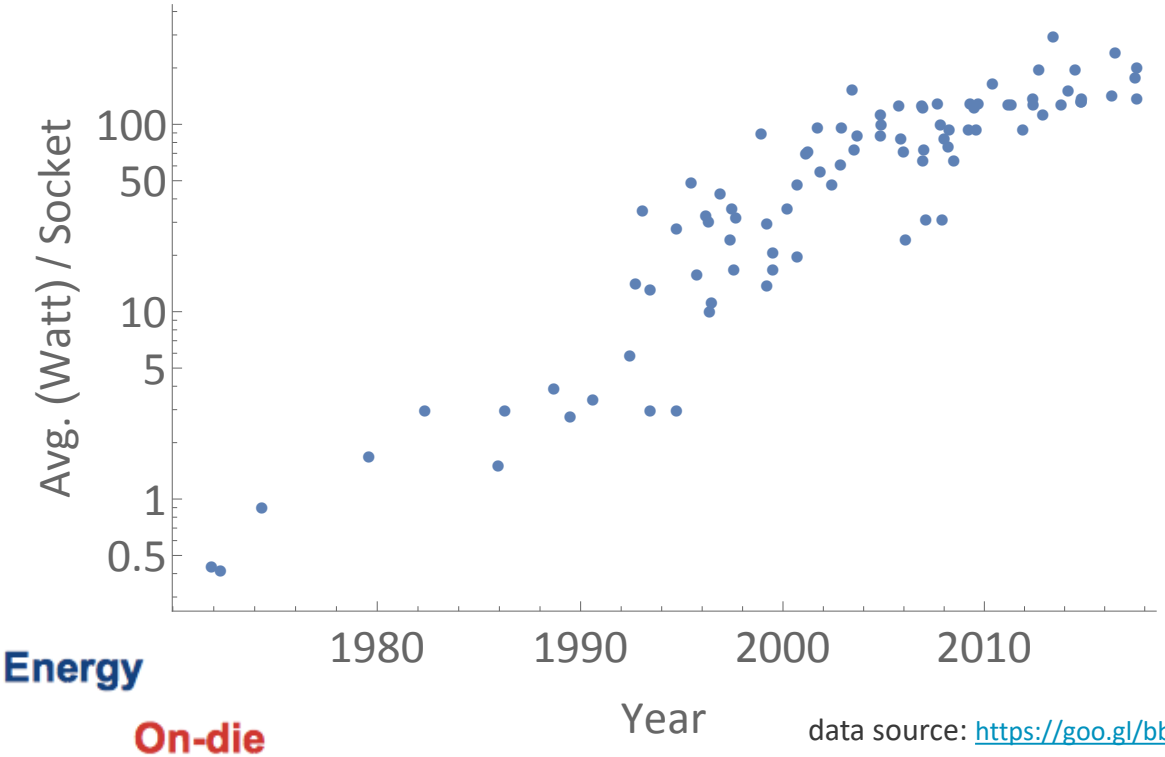
src: https://goo.gl/jUnqYZ

data source: https://goo.gl/bb6wZW

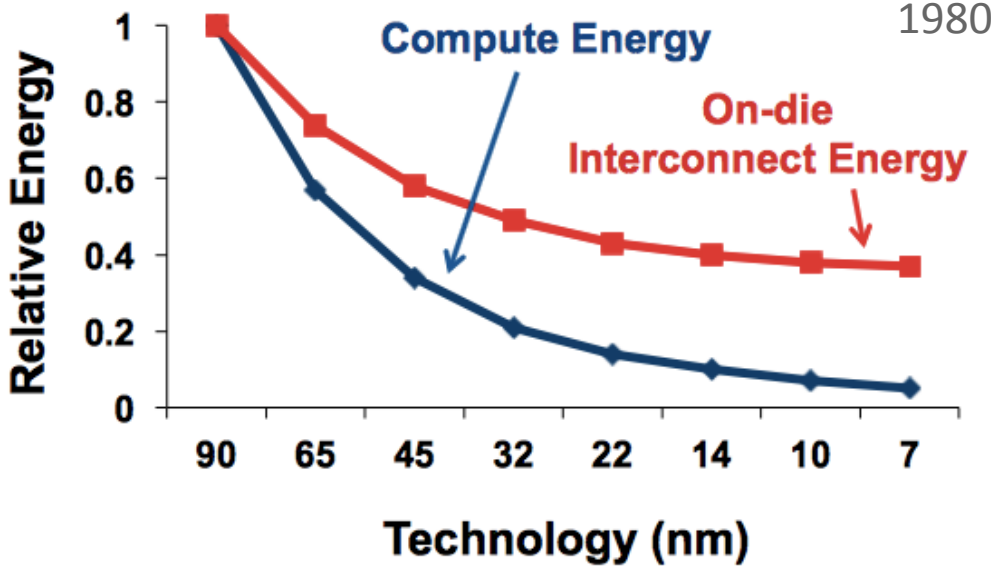
arm

More Power!!





data source: https://goo.gl/bb6wZW



Source: Shekhar Borkar, Journal of Lightwave Technology, 2013

Selection on Computers

src: https://goo.gl/1dQoCZ

$$

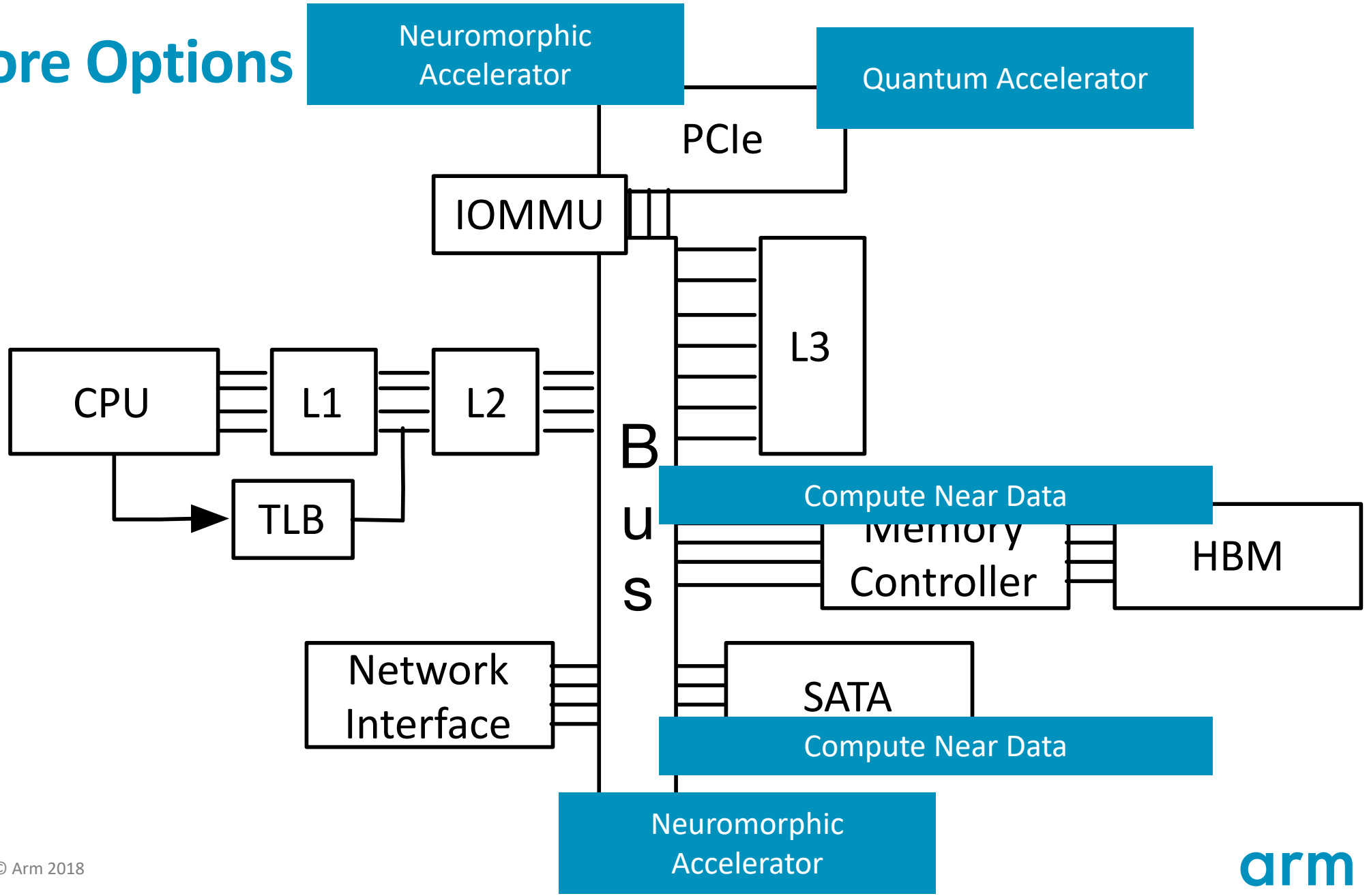
Speed

Energy Consumption

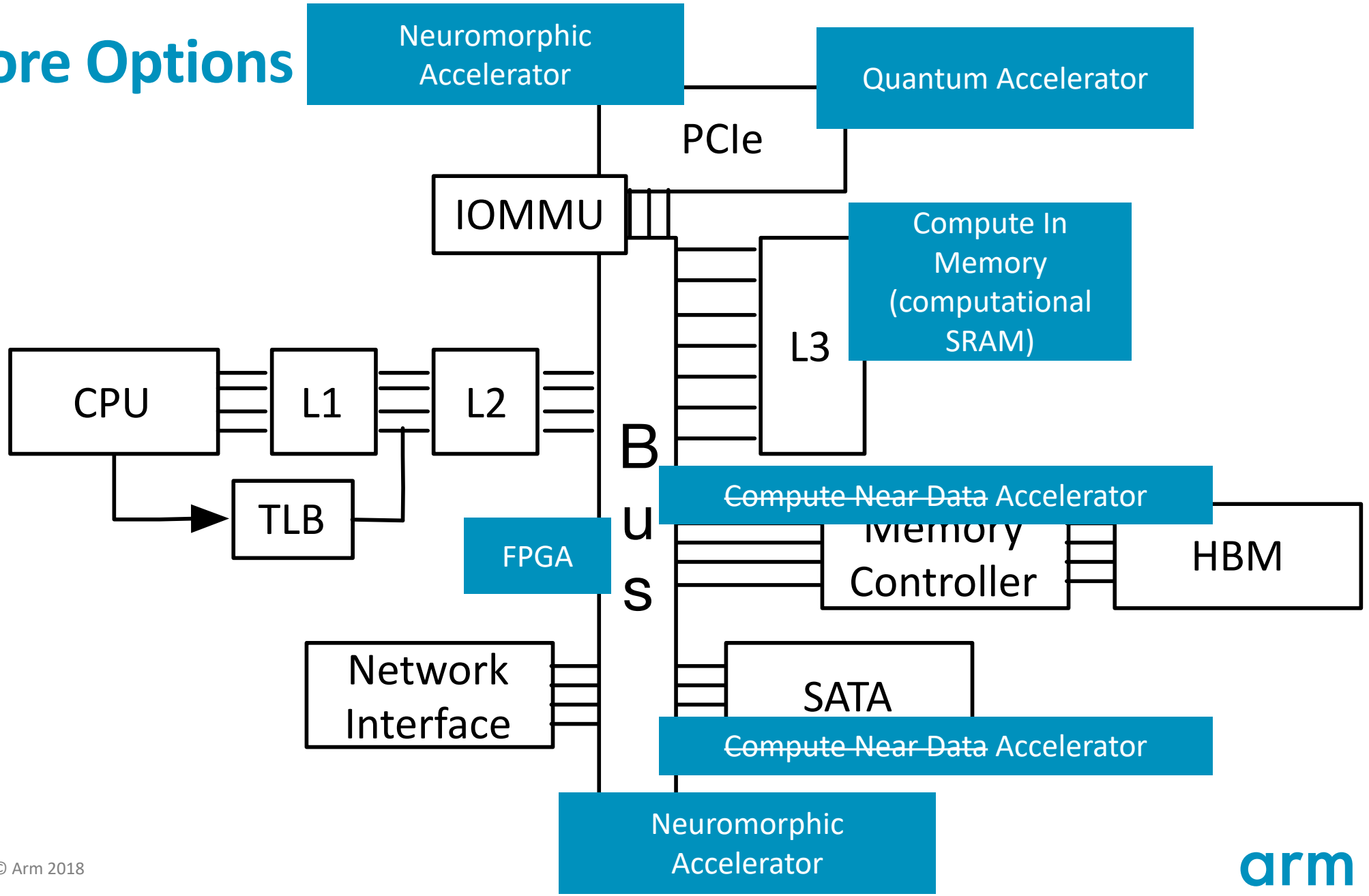
Ease of Use

Software Availability

Post-Moore Options

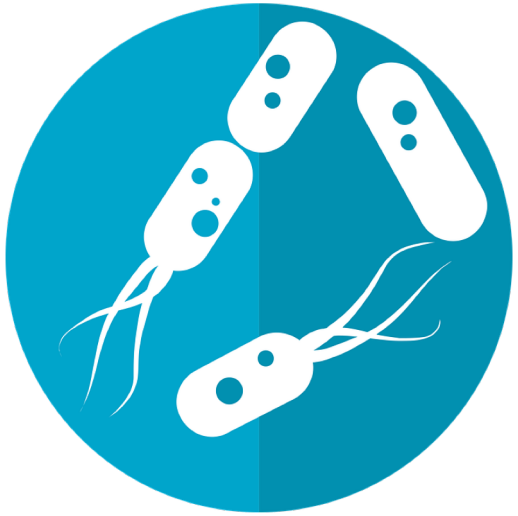


Post-Moore Options

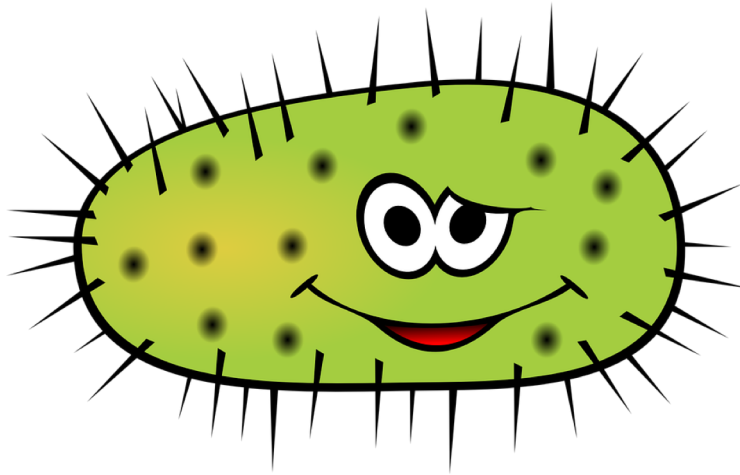


Evolutionary Pressure

Antibiotics



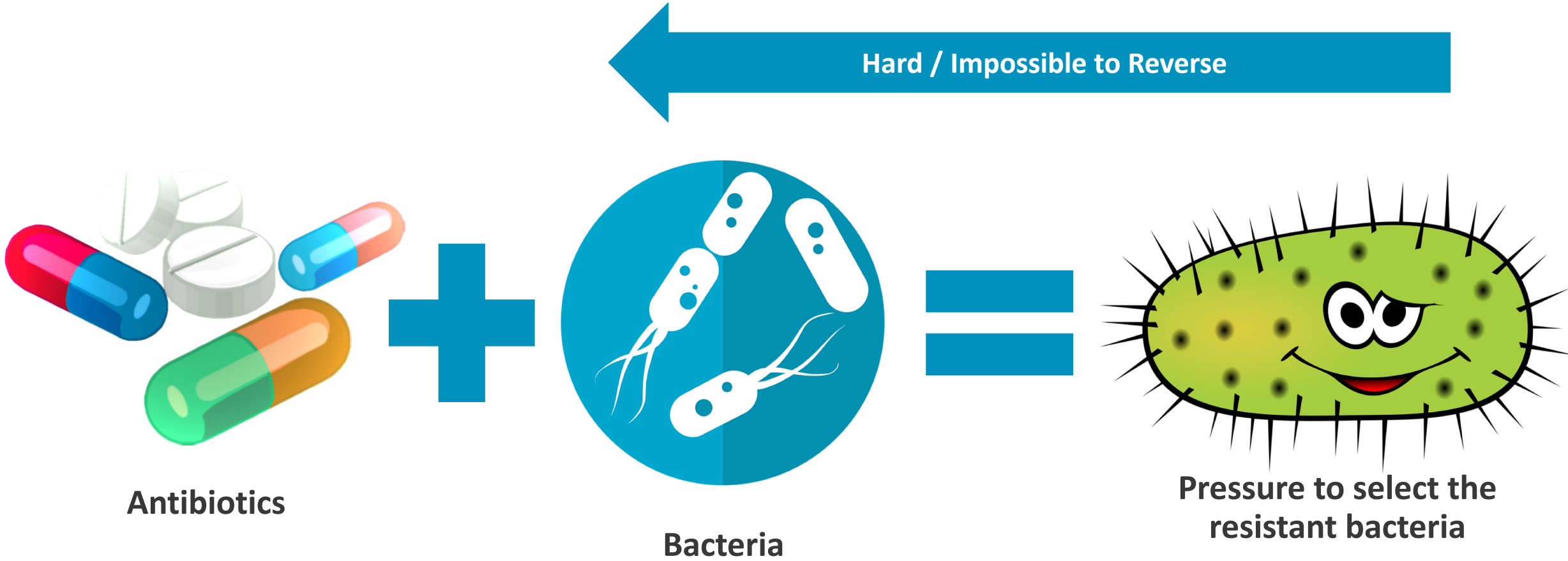
Bacteria



Pressure to select the resistant bacteria

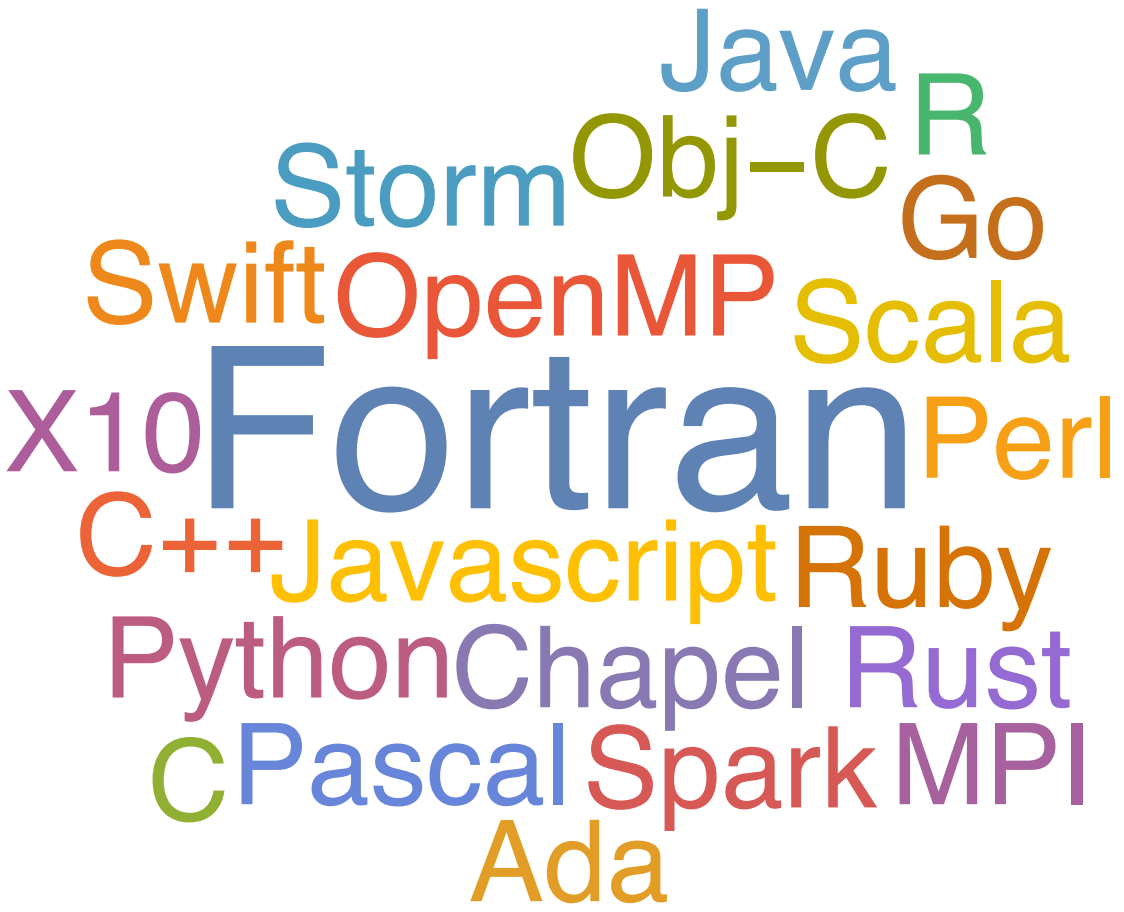
arm

Evolutionary Pressure



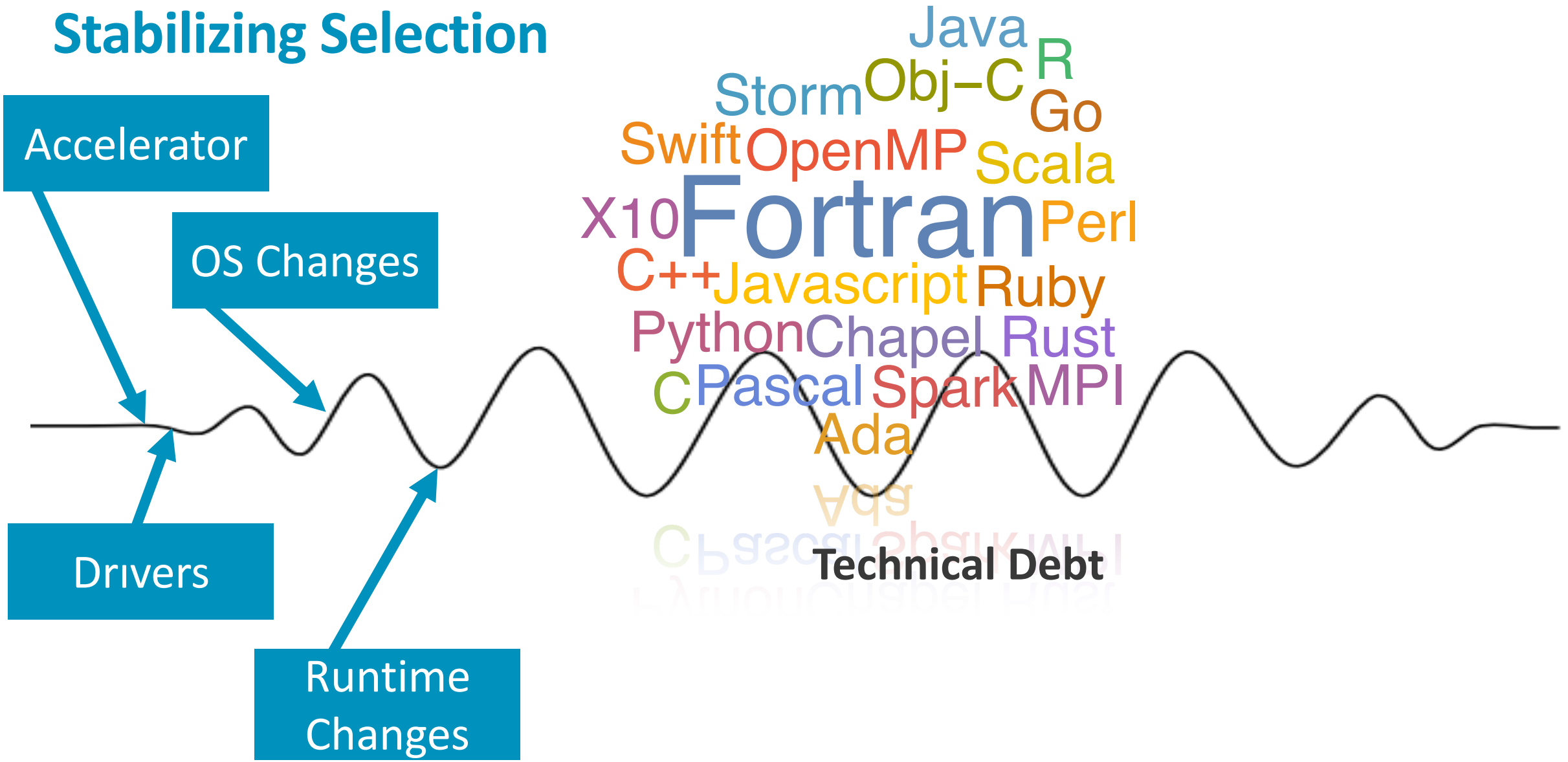
arm

Many Language Options



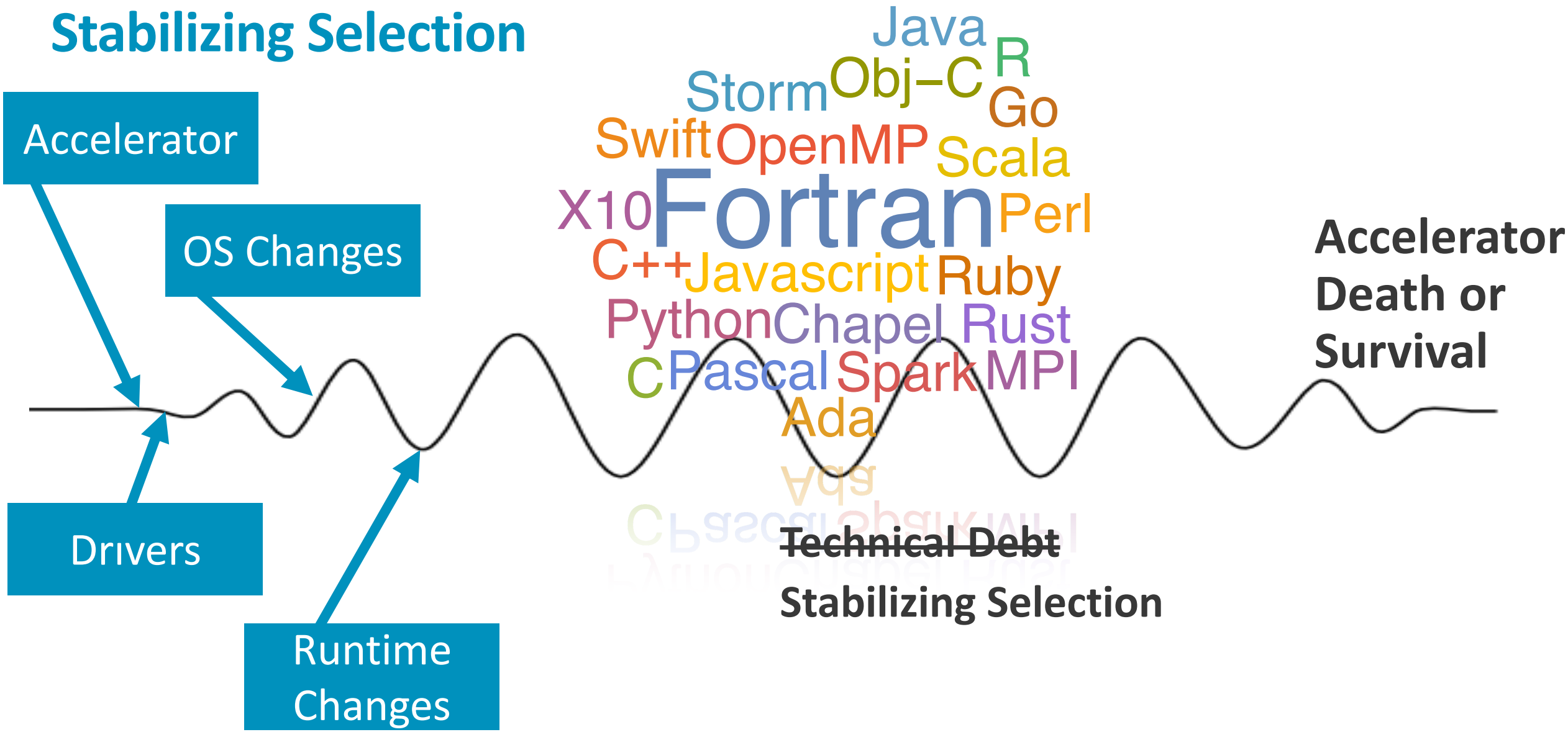
arm

Stabilizing Selection



arm

Stabilizing Selection



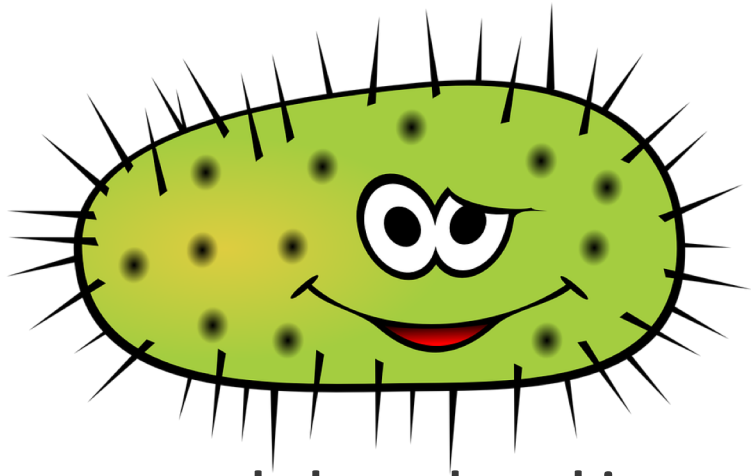
arm

Why language interfaces stabilize

Evolutionary Pressure

src: https://goo.gl/1dQoCZ

Accelerator



e.g., unbalanced machine

arm

Hyper-optimized pressure

src: https://goo.gl/1dQoCZ

Accelerator

No memory system changes

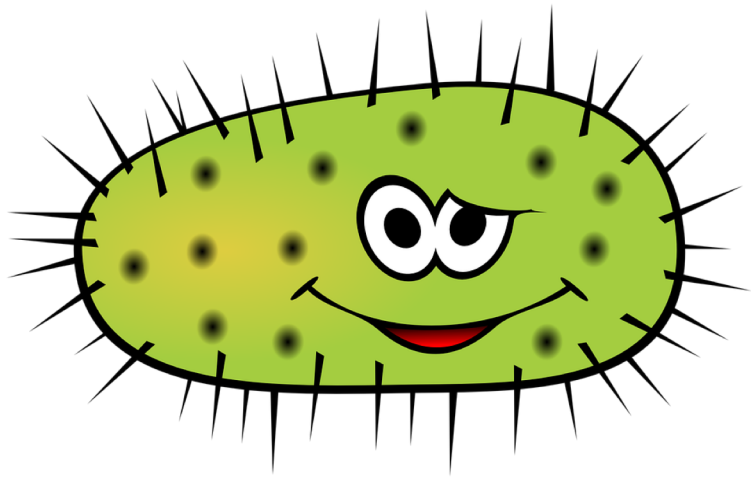
Only fully utilized at 20 FLOPS/B

Hyper-optimized pressure

src: https://goo.gl/1dQoCZ

Accelerator

No memory system changes



arm

Coevolution

src: https://goo.gl/1dQoCZ

Tune Code

```
_MM_TRANSPOSE4_PS(*((__v4sf*)b[4]) , *((__v4sf*)b[5]) , *((__v4sf*)b[6]) , *((__v4sf*)b[7]));
_MM_TRANSPOSE4_PS(*((__v4sf*)(b[4]+4)) , *((__v4sf*)(b[5]+4)) , *((__v4sf*)(b[6]+4)) , *((__v4sf*)(b[7]+4)));
row_swap((__v4sf*)(b[0]+4),(__v4sf*)(b[4]));
row_swap((__v4sf*)(b[1]+4),(__v4sf*)(b[5]));
row_swap((__v4sf*)(b[2]+4),(__v4sf*)(b[6]));
row_swap((__v4sf*)(b[3]+4),(__v4sf*)(b[7]));
```

Coevolution

```
_MM_TRANSPOSE4_PS(*((__v4sf*)b[4]) , *((__v4sf*)b[5]) , *((__v4sf*)b[6]) , *((__v4sf*)b[7]));
_MM_TRANSPOSE4_PS(*((__v4sf*)(b[4]+4)) , *((__v4sf*)(b[5]+4)) , *((__v4sf*)(b[6]+4)) , *((__v4sf*)(b[7]+4)));
row_swap((__v4sf*)(b[0]+4),(__v4sf*)(b[4]));
row_swap((__v4sf*)(b[1]+4),(__v4sf*)(b[5]));
row_swap((__v4sf*)(b[2]+4),(__v4sf*)(b[6]));
row_swap((__v4sf*)(b[3]+4),(__v4sf*)(b[7]));
```

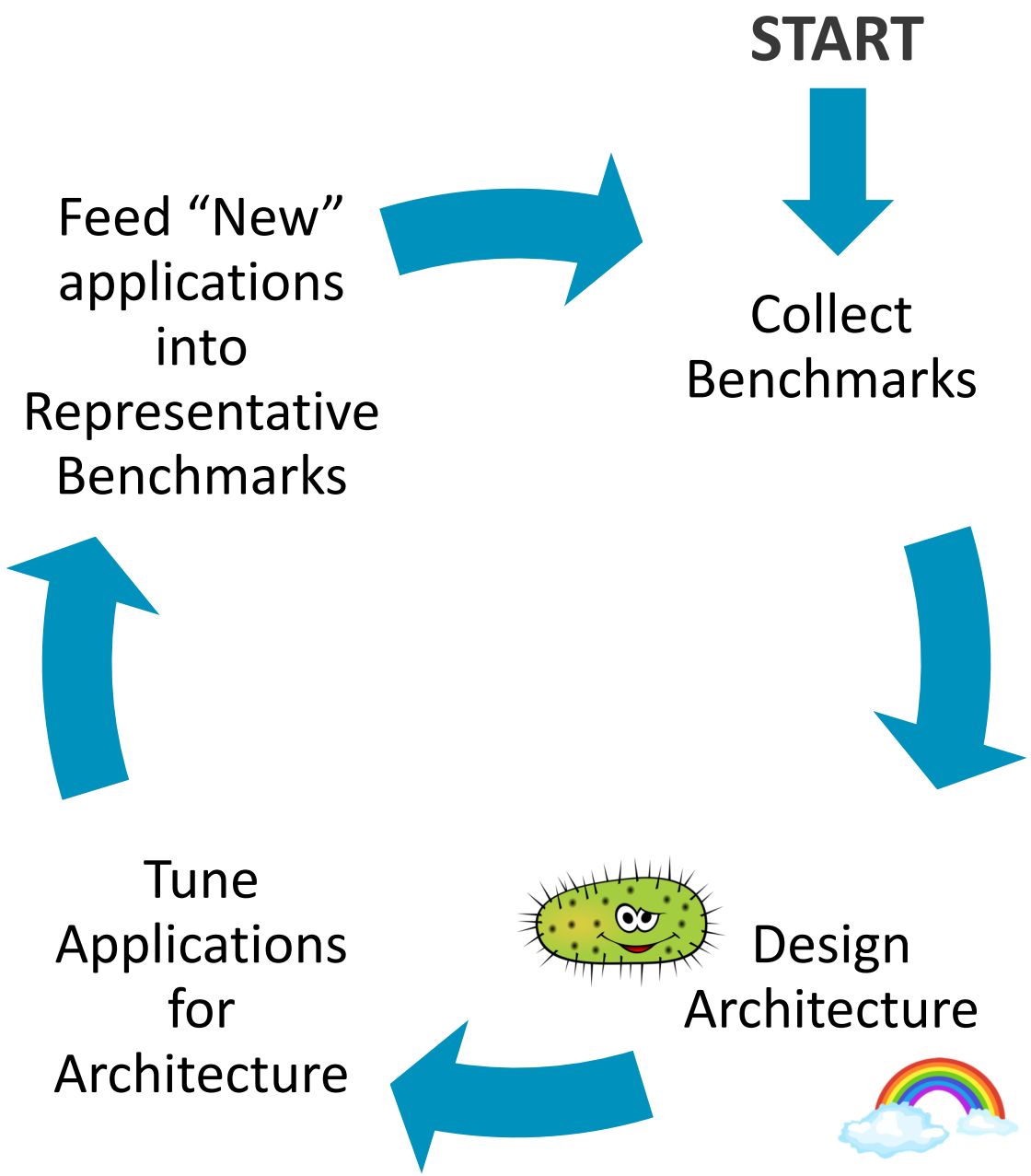
Gather workload requirements

Simulate Architecture

Environmental pressure yields almost identical architecture

src: https://goo.gl/1dQoCZ

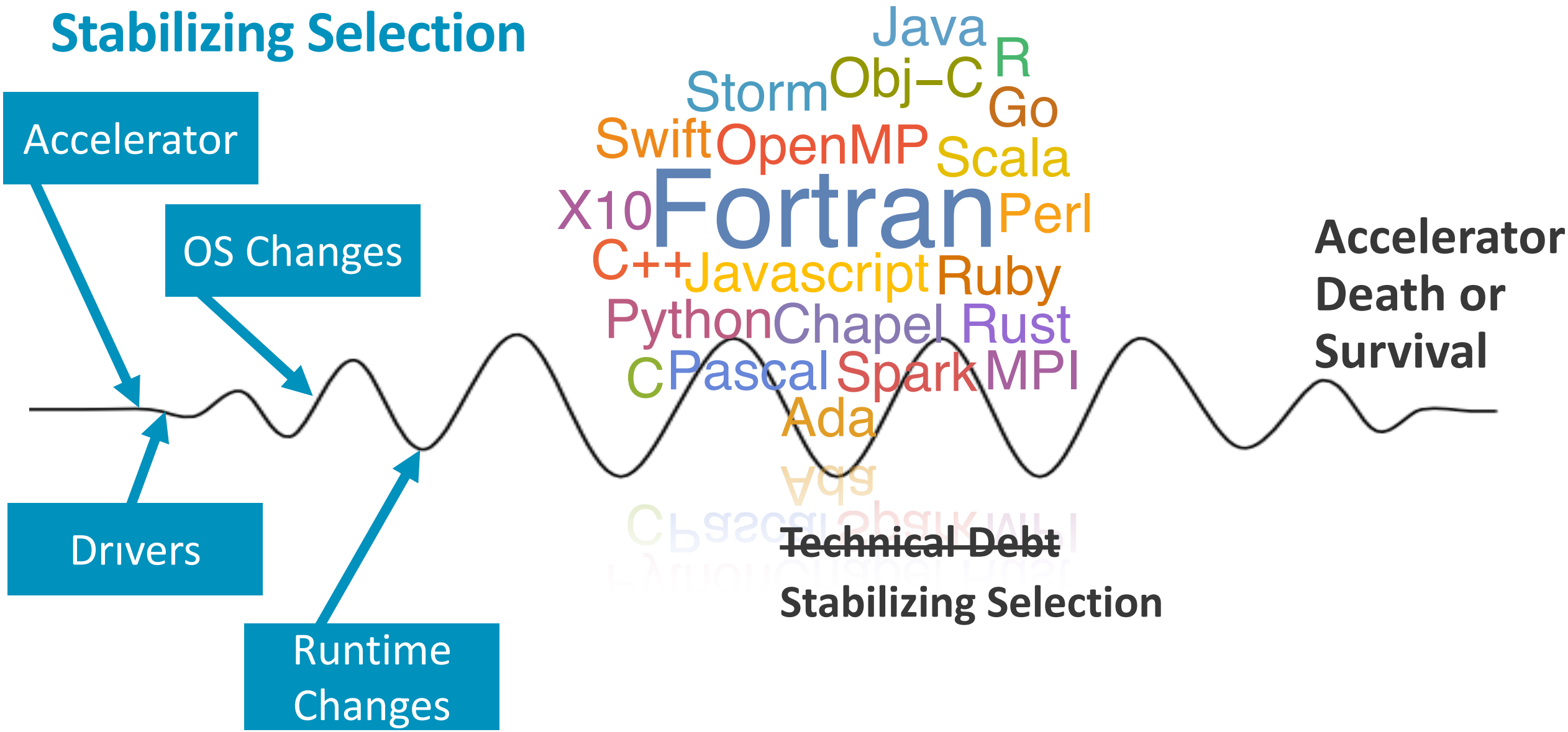
Coevolution



- At some point we have to break out of the cycle.
- Cost of not doing so is getting more of the exact same architecture.
- Perhaps worse, architectures will evolve without much application input.

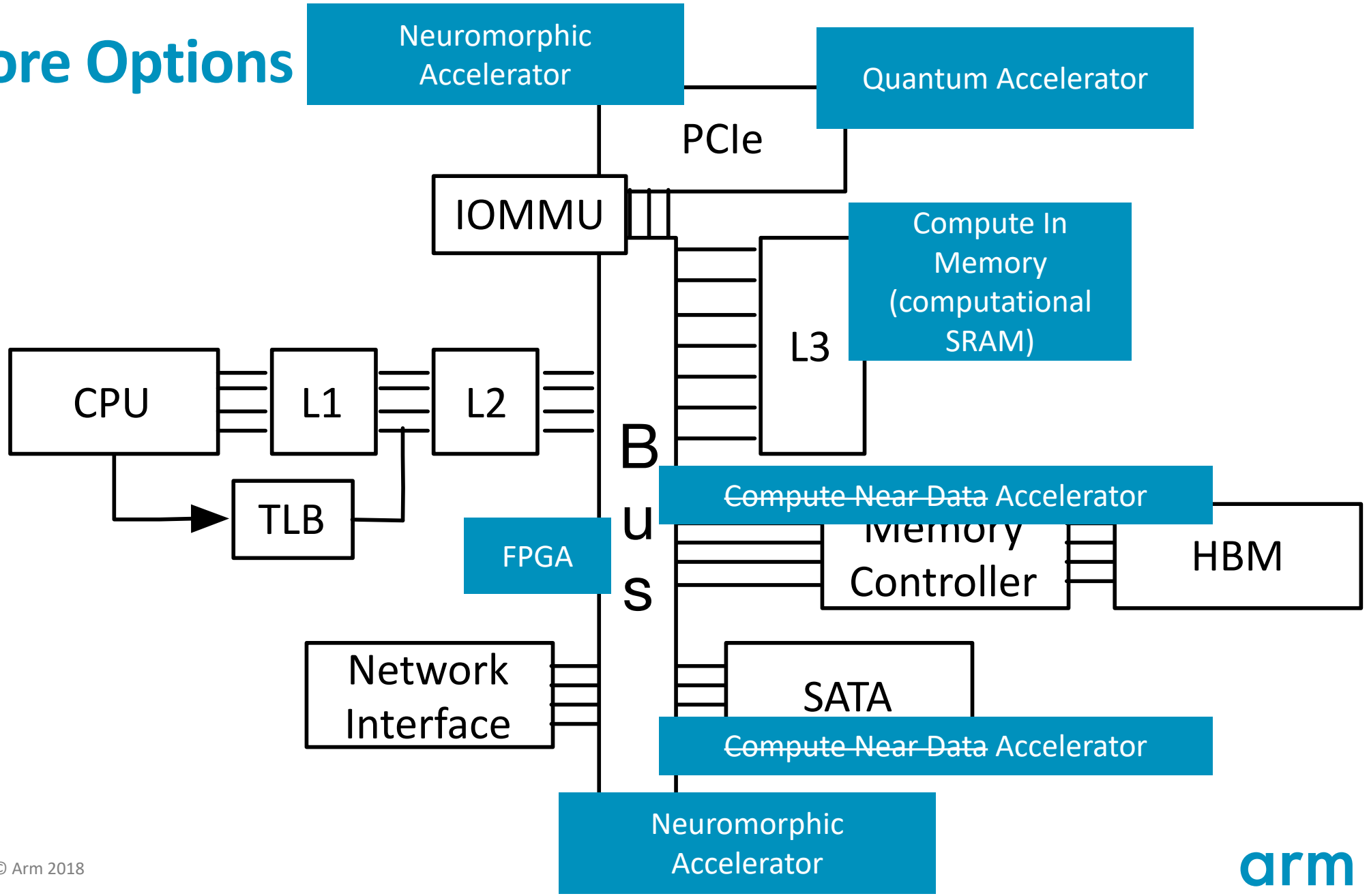
arm

Stabilizing Selection



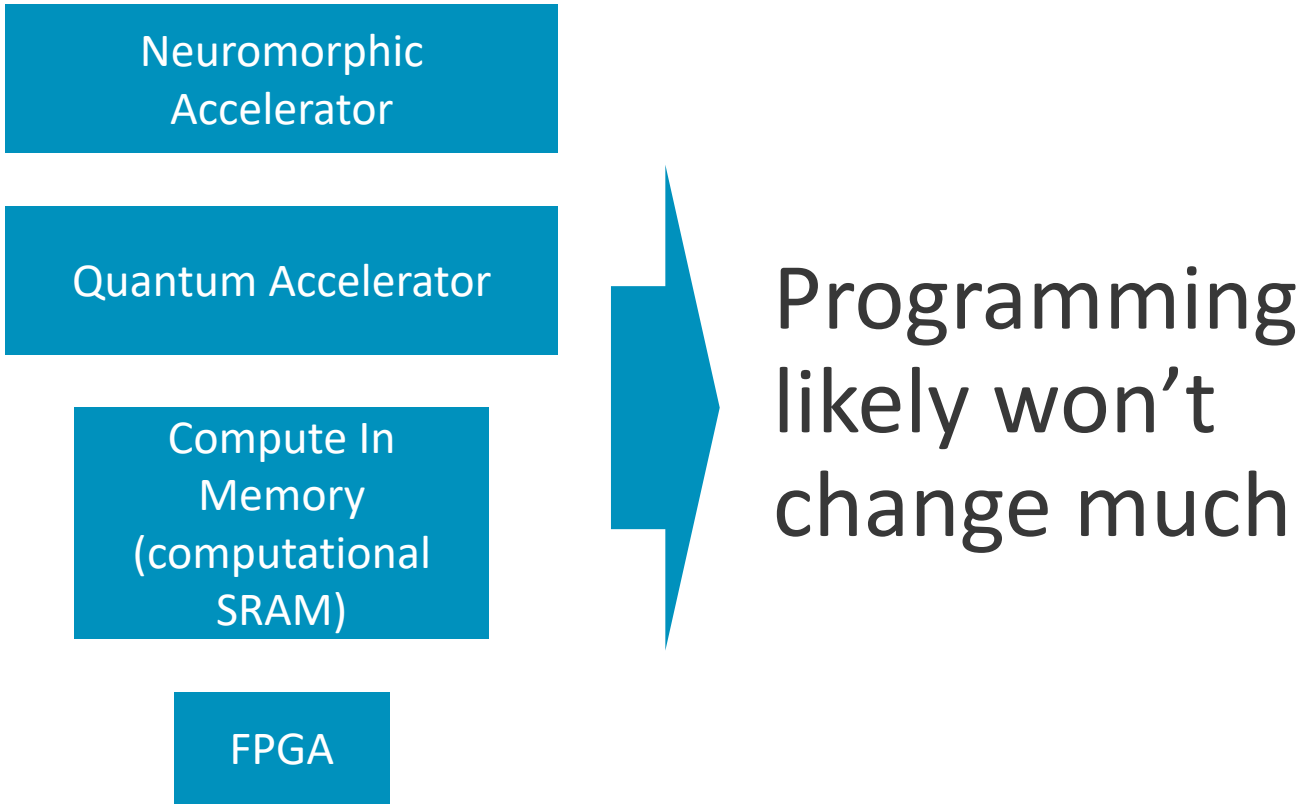
arm

Post-Moore Options



Some disruption will minimize itself

- Stabilizing selection (technical debt)
  - Keeps rapid oscillations in hardware stack under control
  - Convergent evolution driven by software to hardware will drive similar solutions from multiple vendors (also potentially a bad thing…)
- How do we minimize cost of evolutionary changes while maximizing ability to introduce variation?



arm

Natural Selection

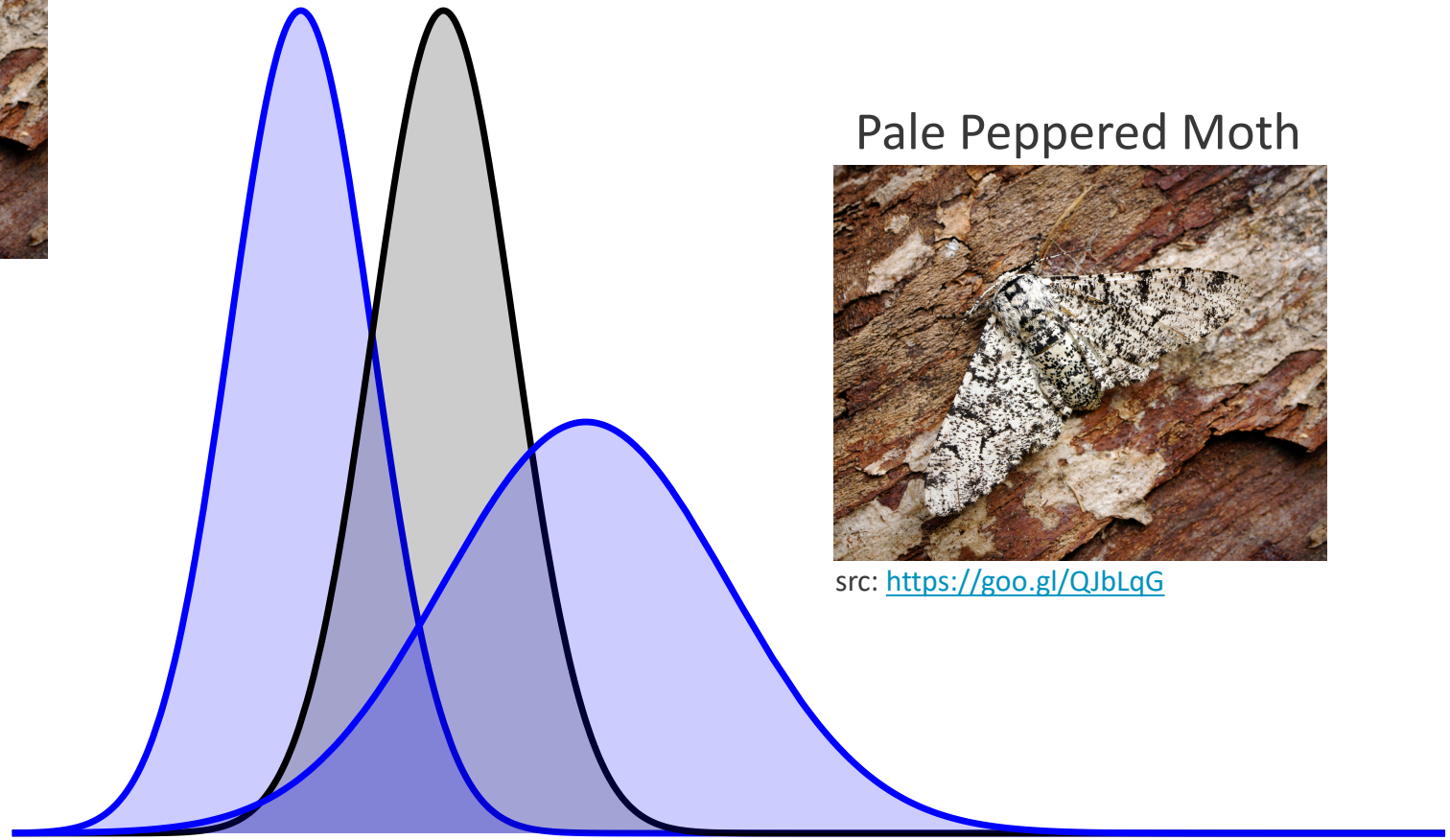
Pale Peppered Moth

src: https://goo.gl/QJbLqG

~~Pale~~ Peppered Moth

src: https://goo.gl/1BBZVG

src: https://goo.gl/yna3tH



Pale Peppered Moth

src: https://goo.gl/QJbLqG

arm

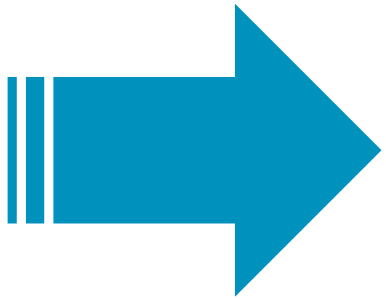
How to minimize disruption

~~Pale~~ Peppered Moth

src: https://goo.gl/1BBZVG

Pale Peppered Moth

src: https://goo.gl/QJbLqG



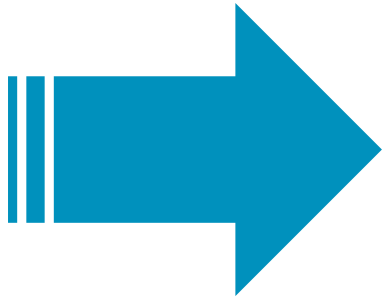
Moth…..

arm

How to minimize disruption

Stable Digital Core

src: https://goo.gl/1dQoCZ



Neuromorphic Accelerator

src: https://goo.gl/1dQoCZ

FPGA

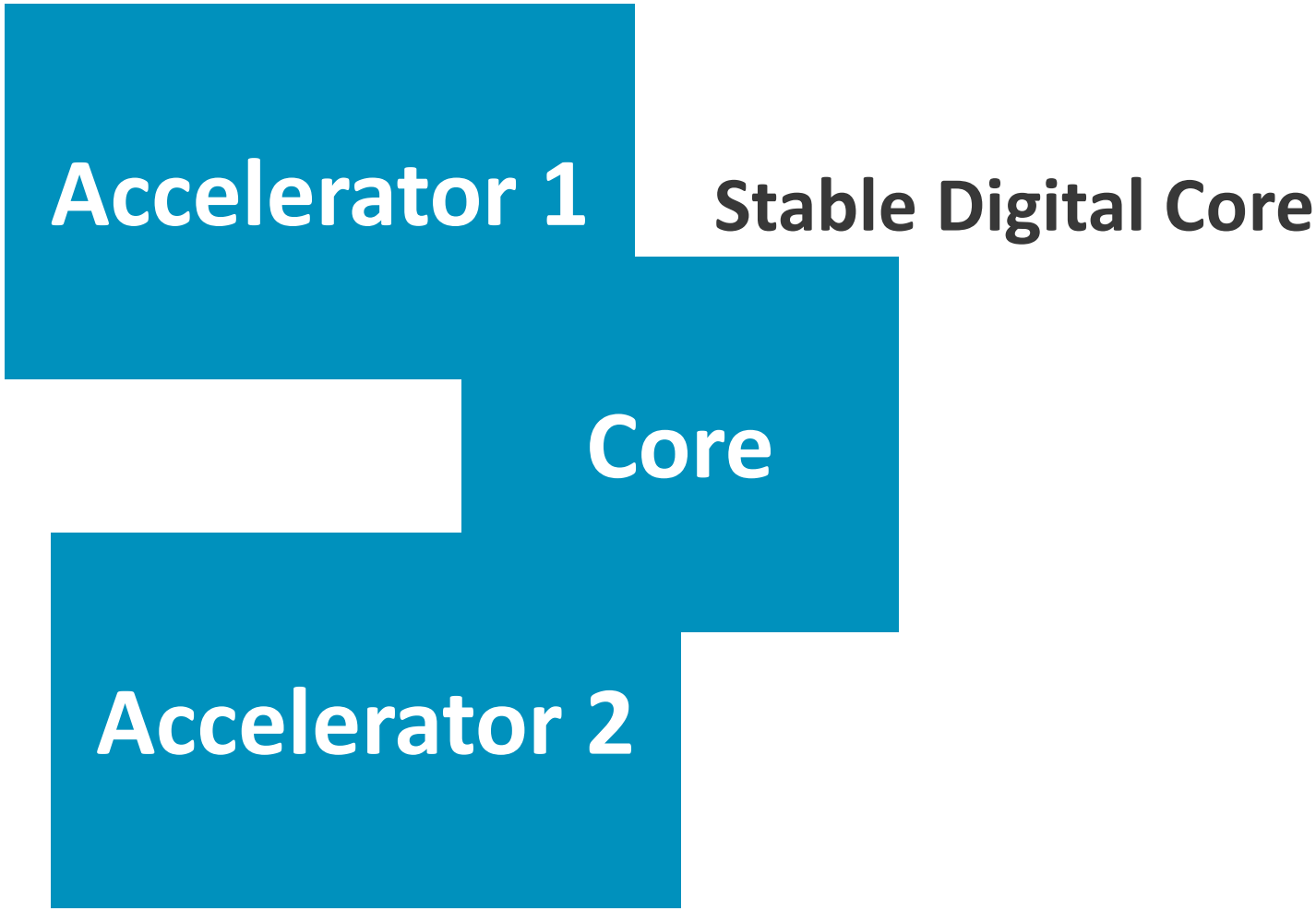
src: https://goo.gl/1dQoCZ

Quantum Accelerator

src: https://goo.gl/1dQoCZ

arm

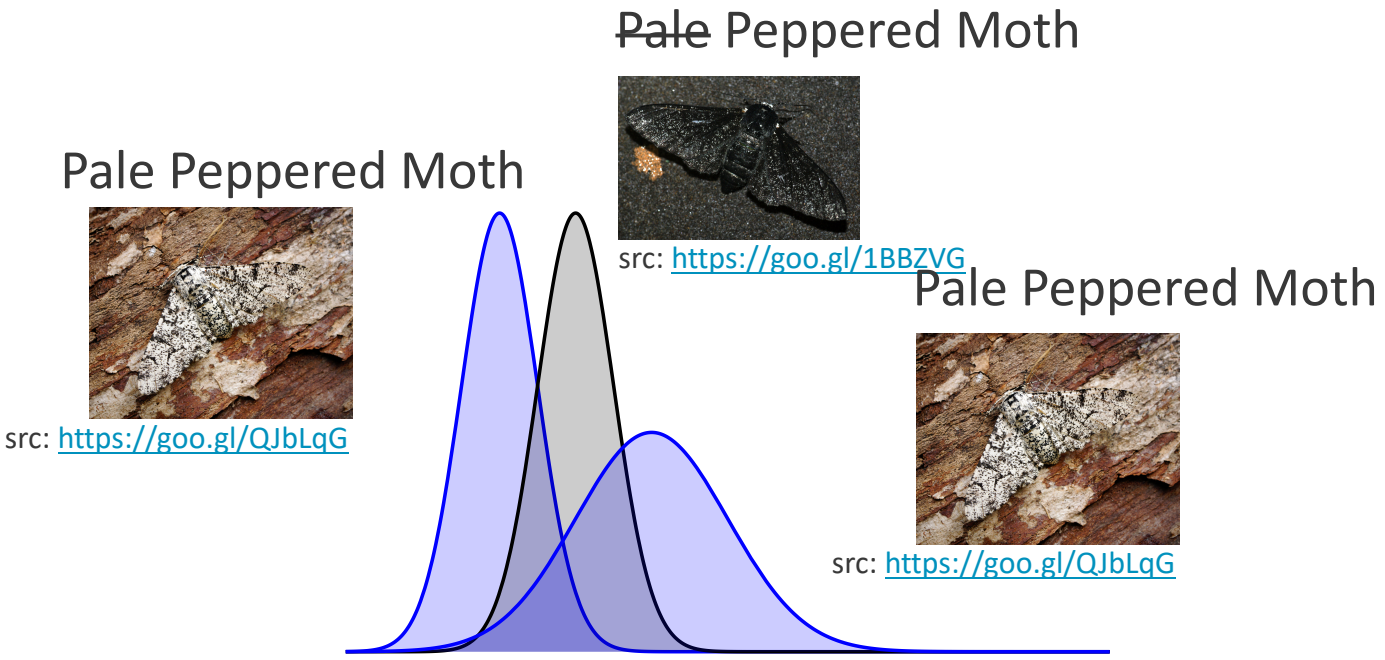
Common core: multi-part chip



arm

Minimize the cost of being eaten

- Enable faster integration of accelerators
- Enable easier integration of new memory technologies
- Enable code to run on multiple architectures more transparently
- Streamline virtual memory and security so accelerators are included natively
- Minimize data movement (tomorrow's talk)
- **Most importantly: re-enable portability**



arm

Folding (a.k.a. Manual Virtual Memory)

- Virtual memory started b/c of lack of available memory vs. storage
- Programmers wrote code that manually folded/unfolded to storage
- Huge controversy existed b/c all code at the time was written this way – Automatic folding hardware was ~25% slower than manual folding

"In ceasing to expend energy (item 1) **in a process whose main result is to make programs less fit to run on other machine configurations** (item 2), or **to run in company with other programs** (item 3), or to **run with temporarily reduced resources** (item 4), we do more than reduce costs; **we remove self-created obstacles** which today are impeding the development of needed types of systems"

- D. Sayre, IBM Yorktown Research (1969)

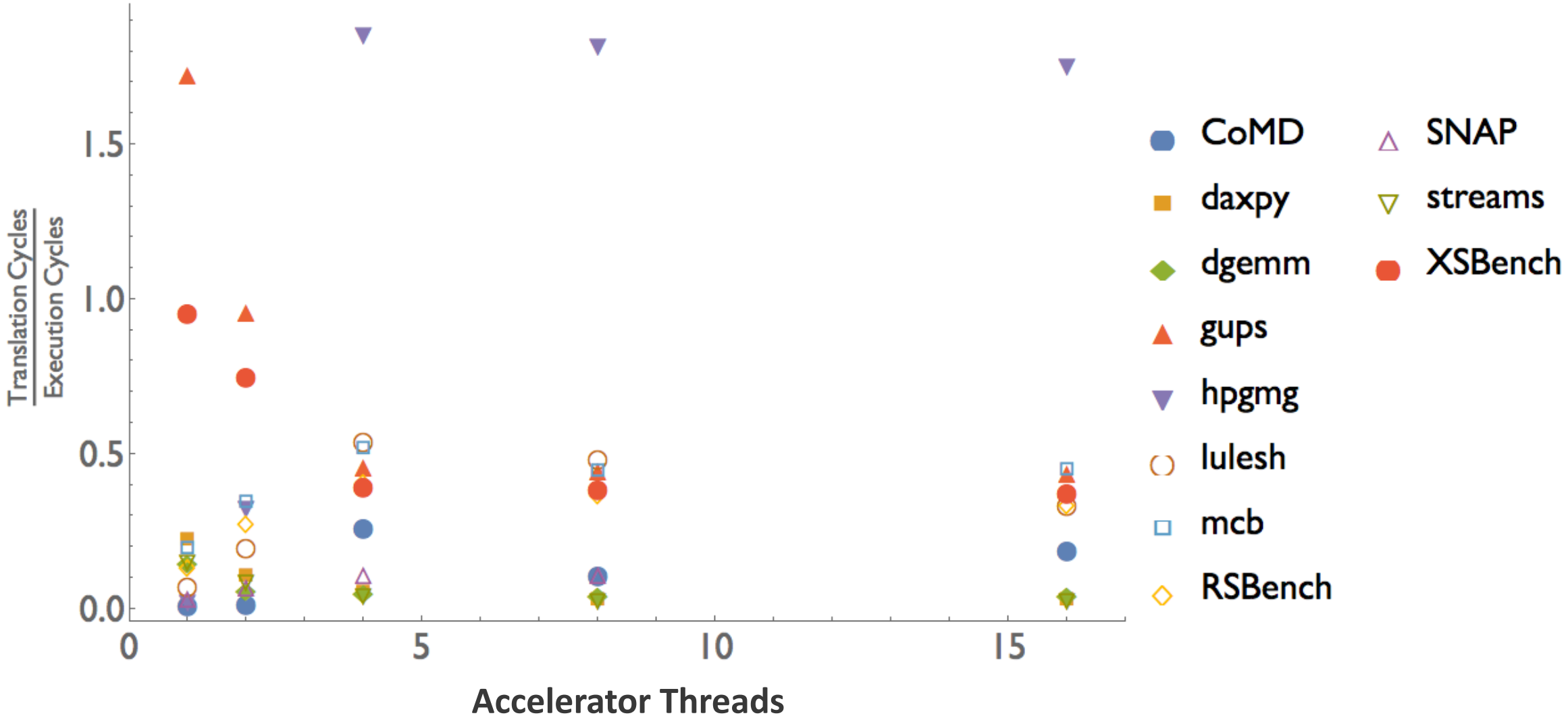
1970 - The Circle of Tech - “Self Created Obstacles”

Move In

Move Out

arm

Constraining force: cost of extending virtual memory

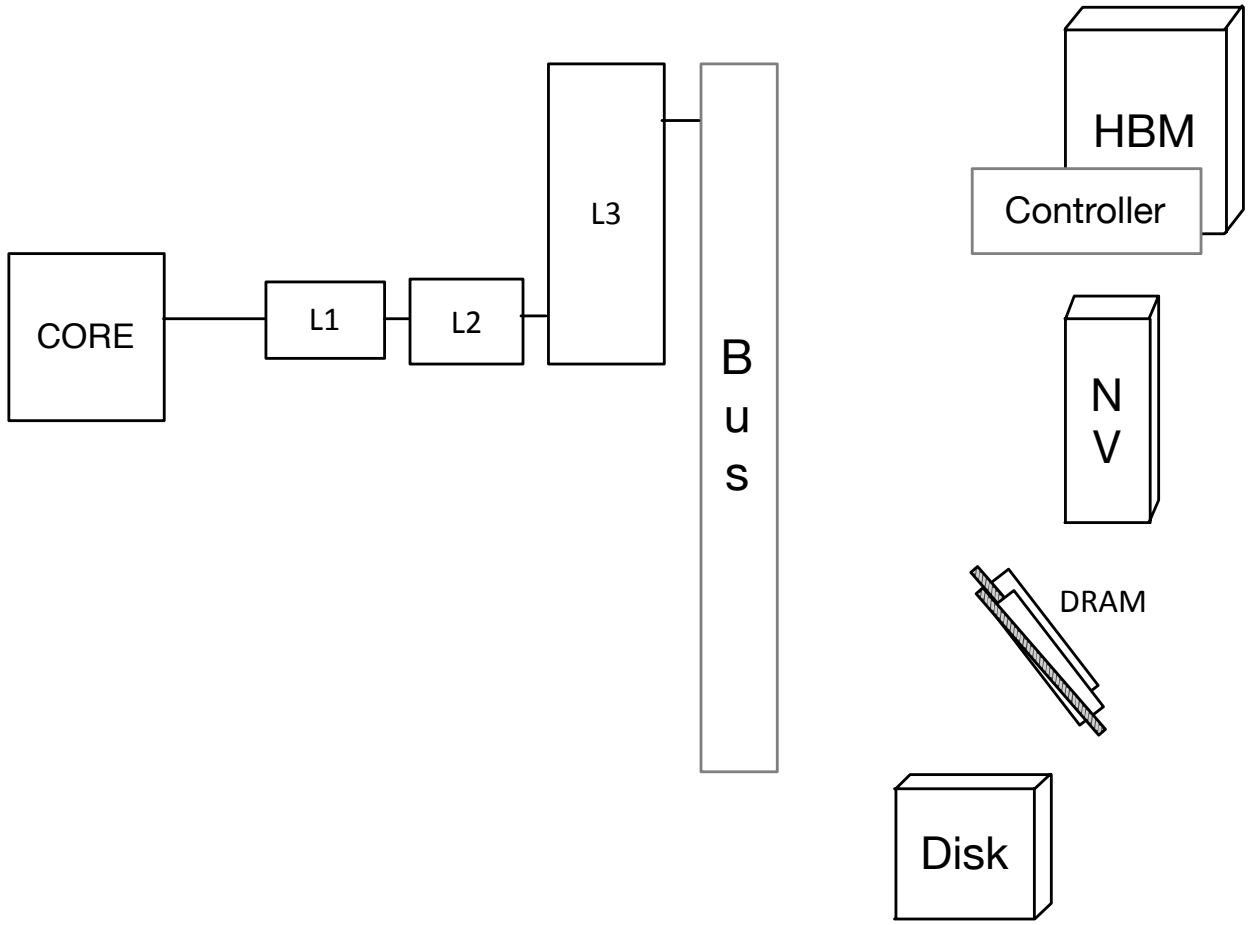


arm

Memory Interface

Software Interface —?— Hardware Interface

- New memory technologies are perpetually 5 years out.
- Except when they're not.
- On the horizon, NV (various flavors), higher bandwidth DRAMs….How to interface with?

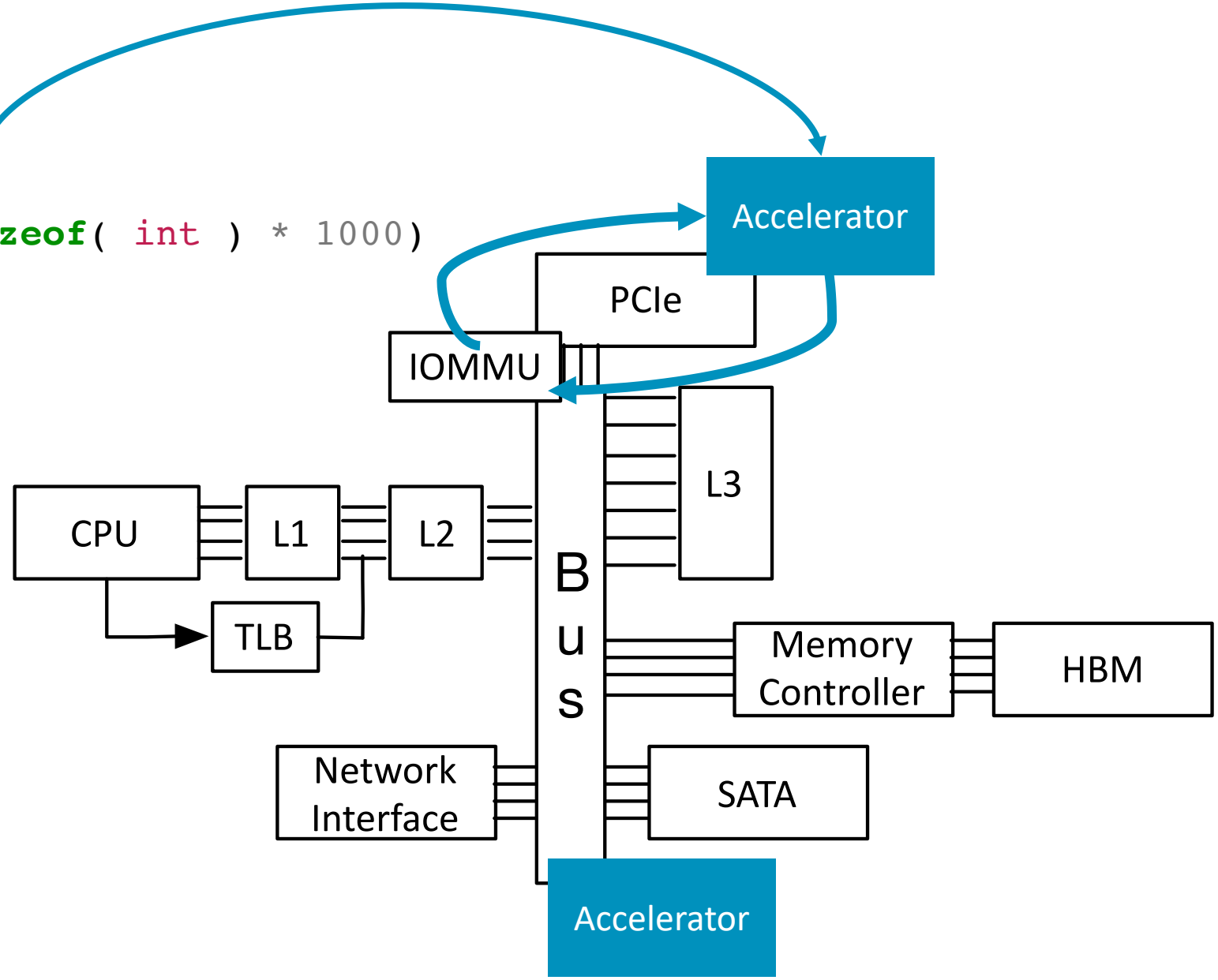


arm

Virtual Memory

```
auto *foo(
        (int*) malloc( sizeof( int ) * 1000)
    );
```

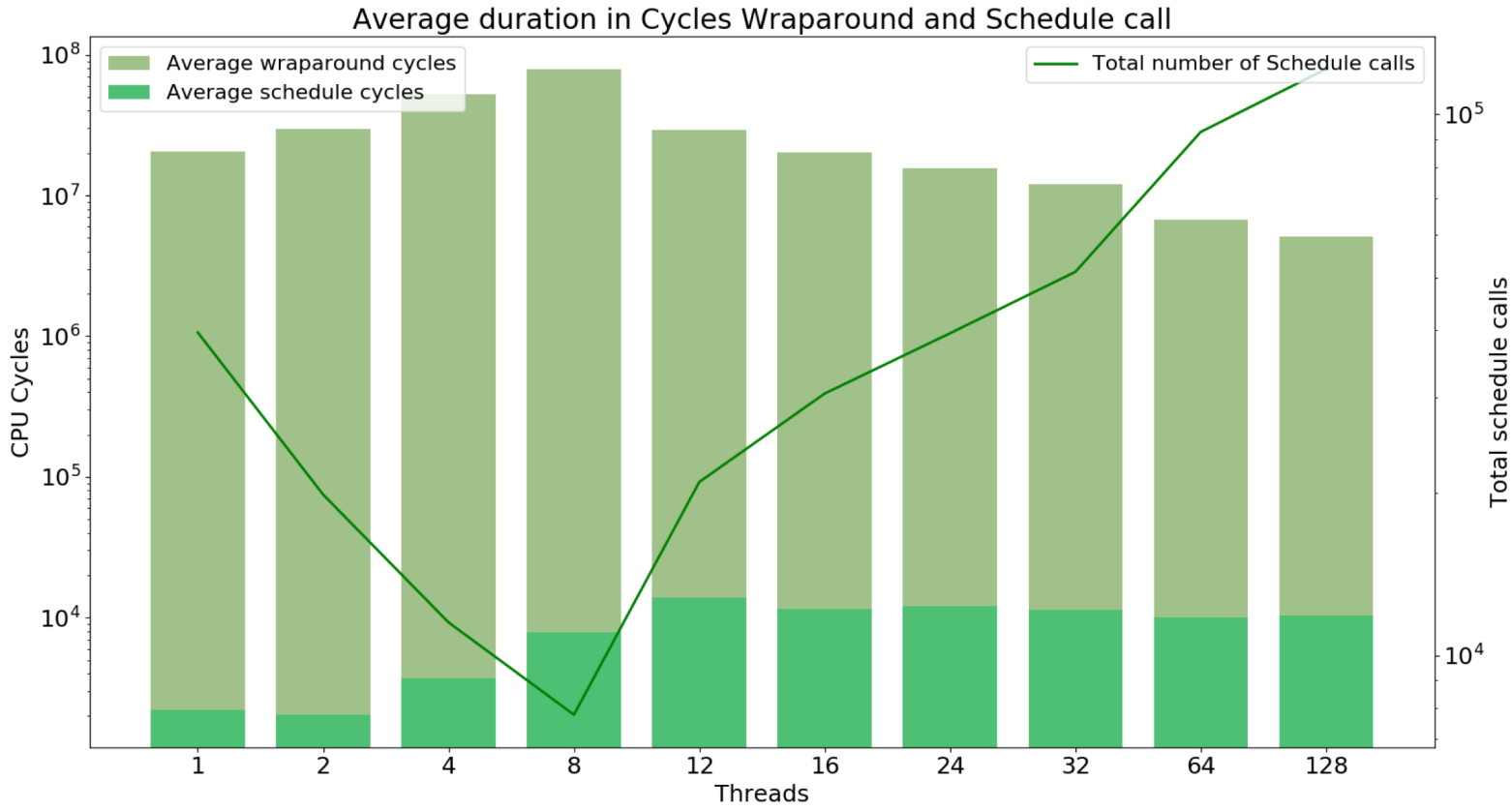
- Plug accelerators into VA space as 1st class citizens.
- Can hide underlying memory interface
- We must develop a better virtual memory that accepts accelerators more easily.
- CCIX, GenZ, etc. are a step forward
- Co-design needed for further dev



arm

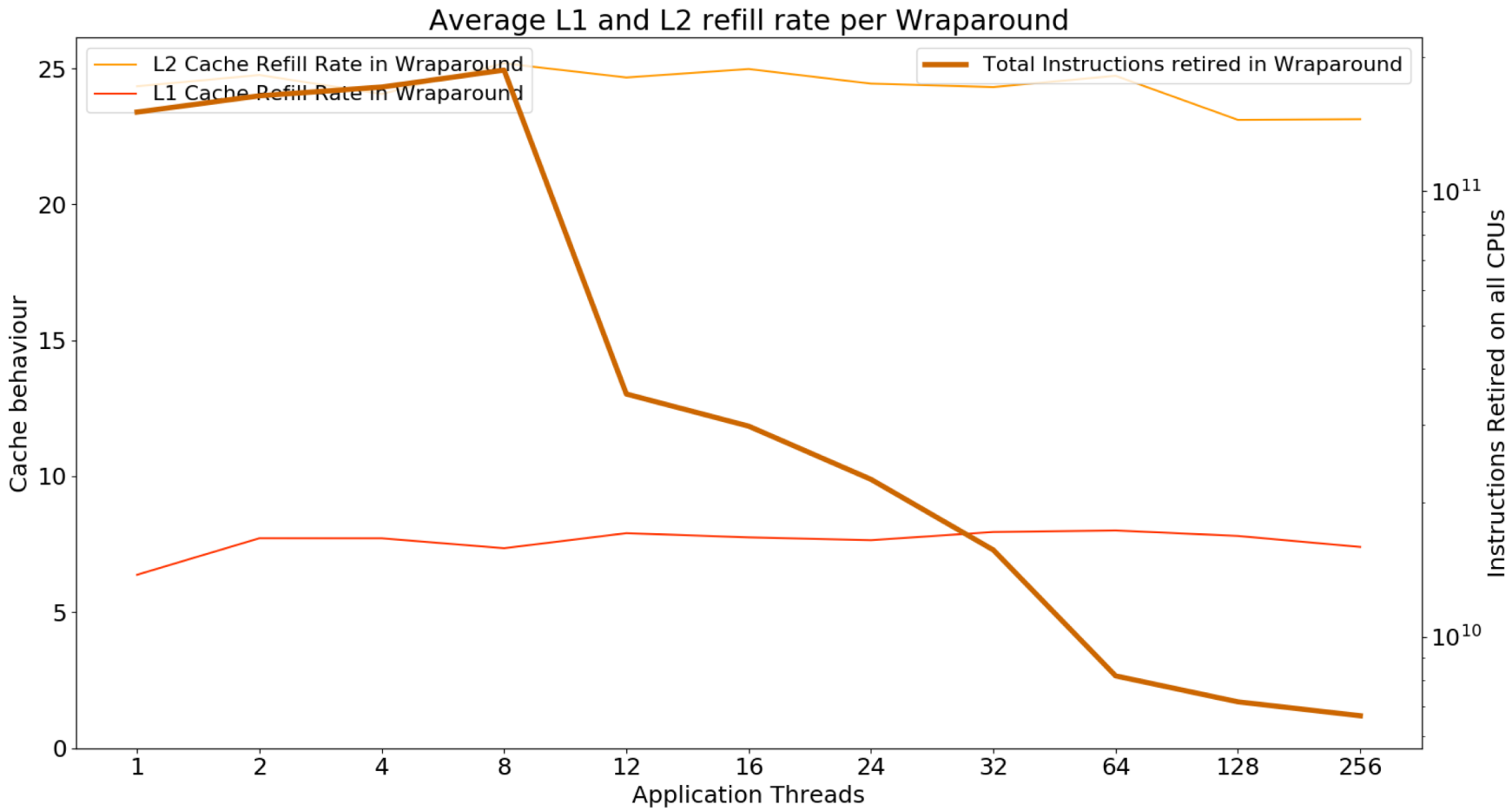
Example: current context swap behavior

- Newer NV tech enables access in 100-2000 cycle range
- Reduce context swap overhead (currently Linux overhead for swap is 2000—5000 cycles) to enable hiding of intermediate latencies.



Secondary context swap behavior

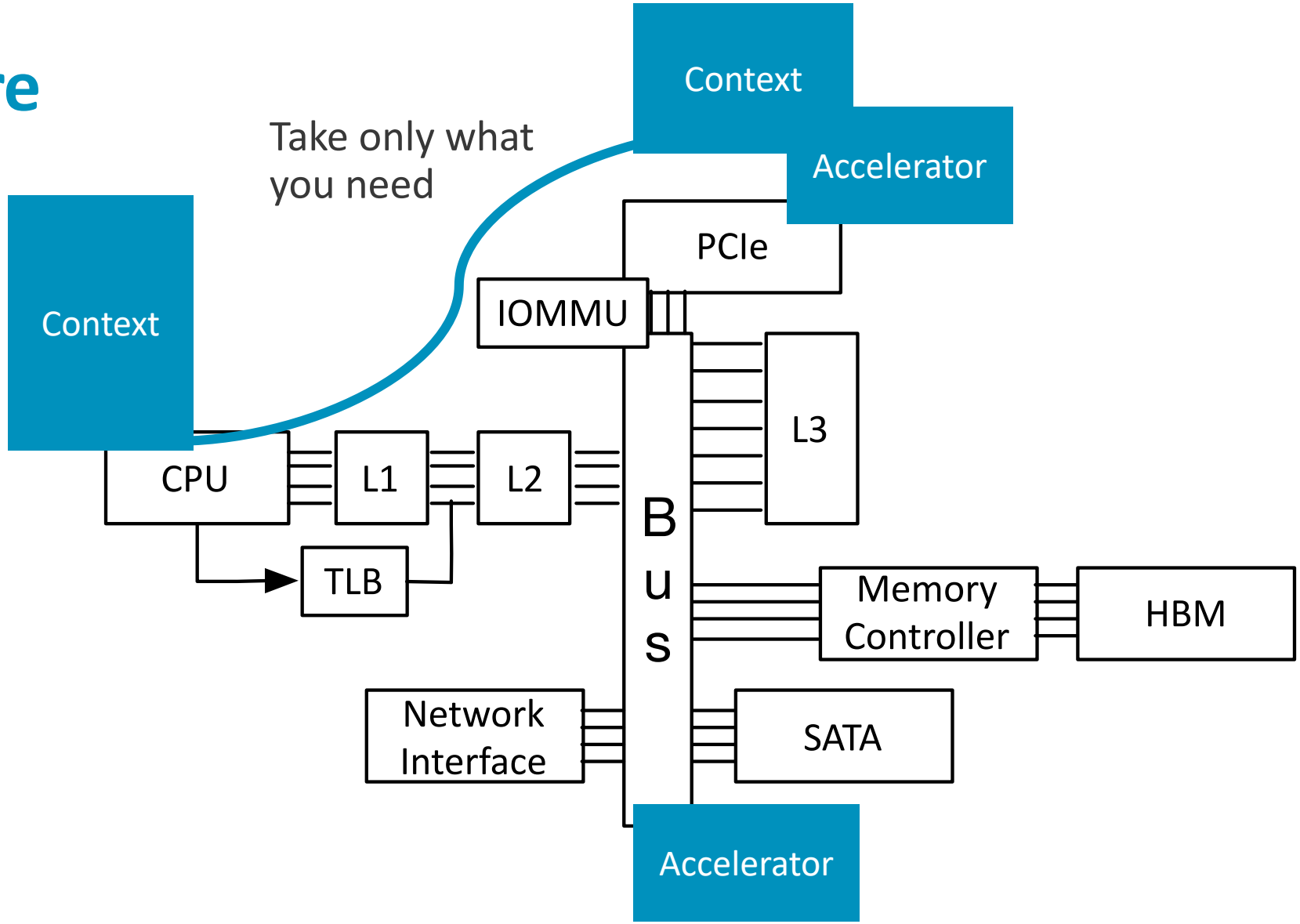
- Future systems must reduce data movement cost
- Current overhead is relatively constant for swap, directly impacts application performance



arm

Context Architecture

- Reduce/remove drivers from accelerator offload mechanism
  - Enable standardized exception formats
  - Enable simple state transference.



arm

Conclusion

- We will see some serious attempts at disruption over next 5—20 years
- Many disruption will die quickly b/c of our technical debt
- Providing a better hardware/software abstraction can enable faster evolution while reducing destabilizing impact to software of technology disruption.
- Better interfaces are needed for:
  - Virtual memory
  - Context migration
  - Exception handling
- Slides will be available at: http://www.jonathanbeard.io/media
- Feel free to Tweet me @jonathan_beard
- Thanks for listening!