MICROKERNELS IN THE ERA OF DATA-CENTRIC COMPUTING

Martin Děcký
martin.decky@huawei.com

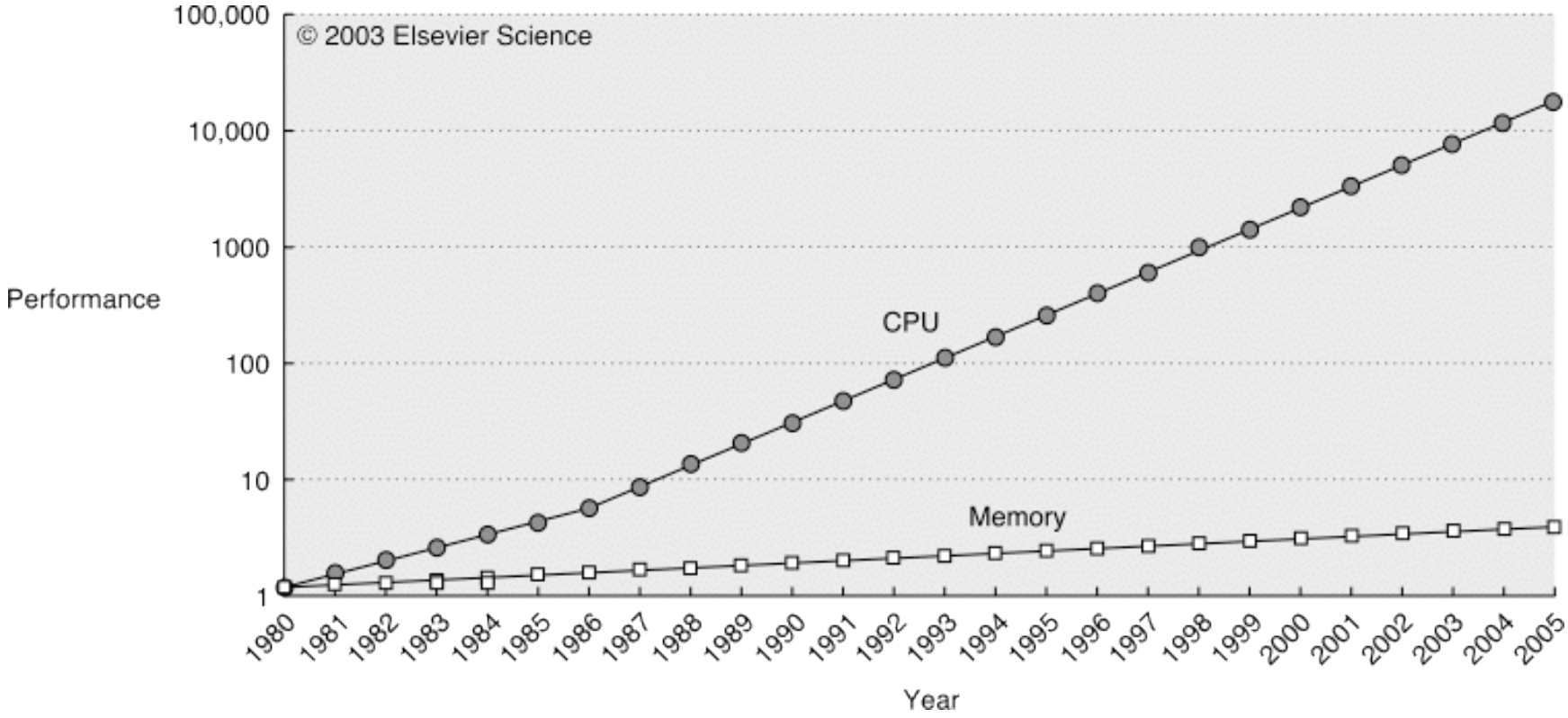
February 2018

Who Am I

- **Passionate programmer and operating systems enthusiast**
  - With a specific inclination towards multiserver microkernels
- **HelenOS developer since 2004**
- **Research Scientist since 2006**
  - Charles University (Prague), Distributed Systems Research Group
- **Senior Research Engineer since 2017**
  - Huawei Technologies (Munich), Central Software Institute

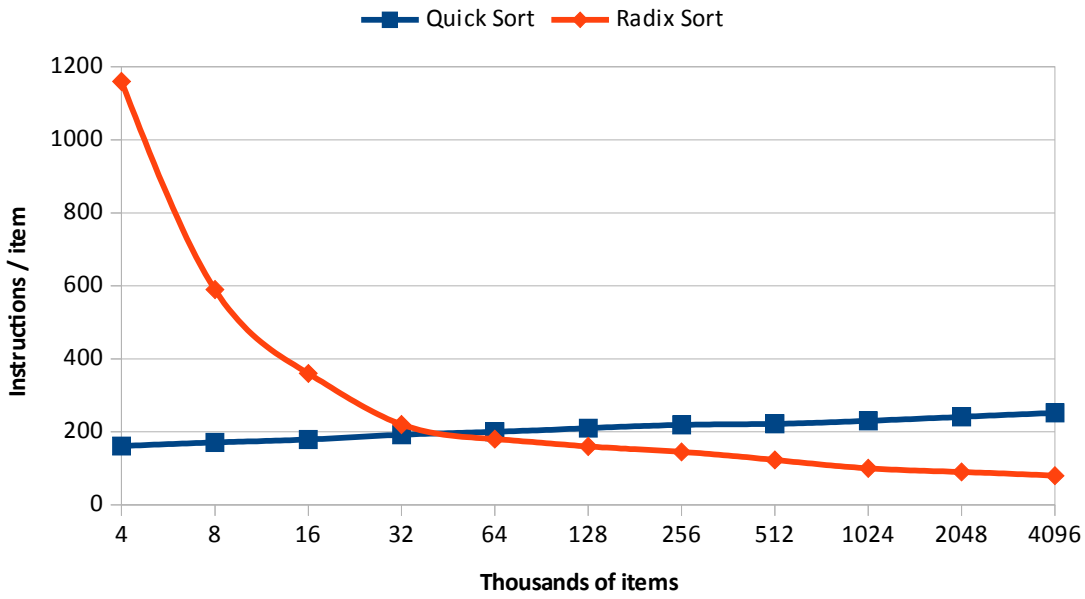
MOTIVATION

Memory Barrier



LaMarca, Ladner (1996)

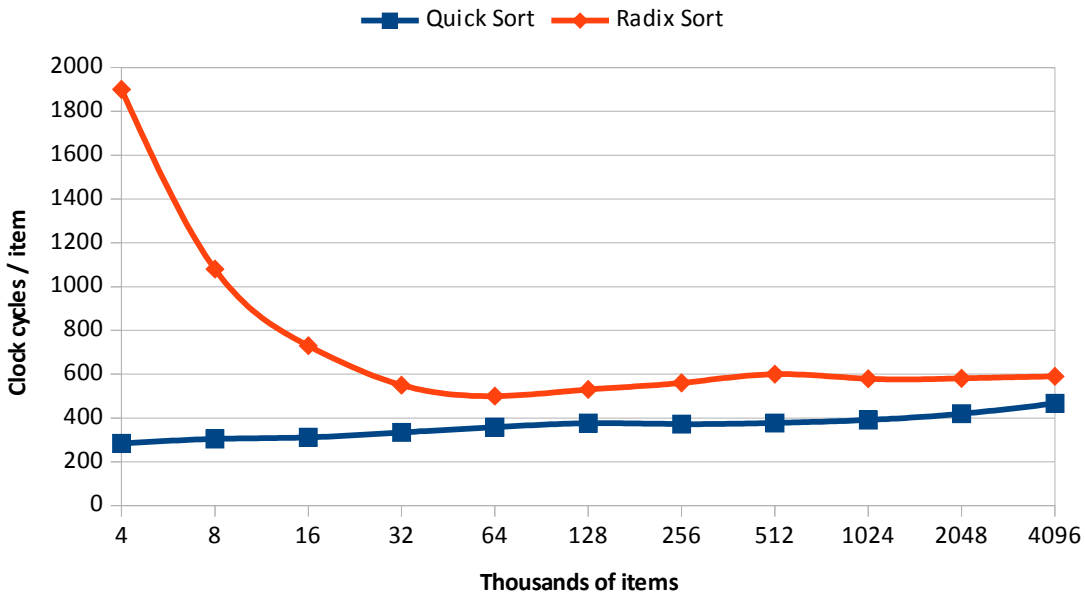
- **Quick Sort**
  - **O(*n*×log *n*)** operations
- **Radix Sort**
  - **O(*n*)** operations



HUAWEI

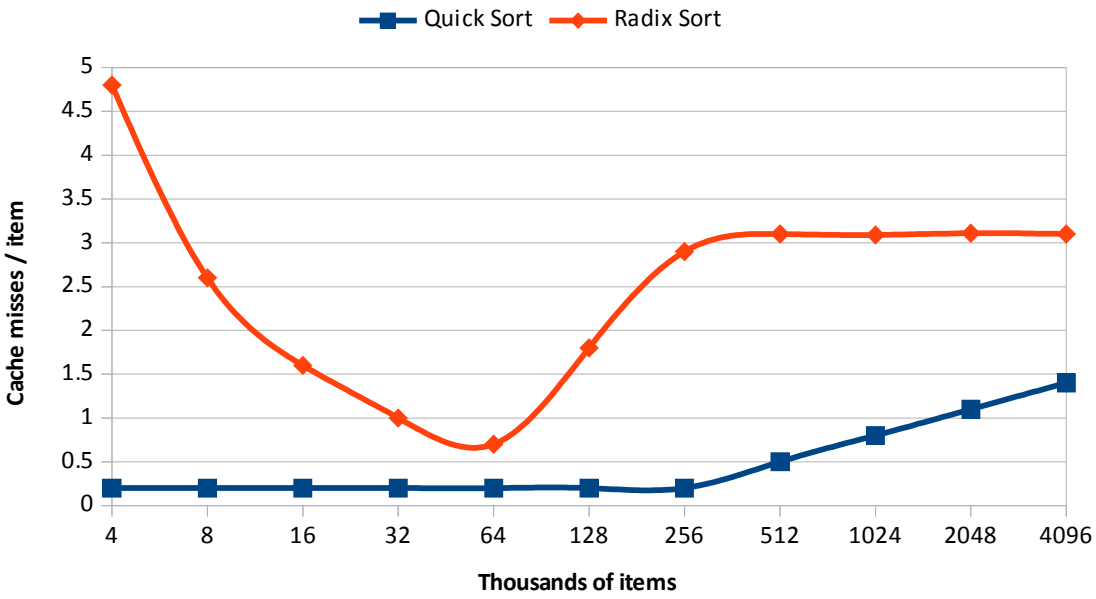
LaMarca, Ladner (1996)

- **Quick Sort**
  - **O(*n*×log *n*)** operations
- **Radix Sort**
  - **O(*n*)** operations



LaMarca, Ladner (1996)

- **Quick Sort**
  - **O(*n*×log *n*)** operations
- **Radix Sort**
  - **O(*n*)** operations



The Myth of RAM

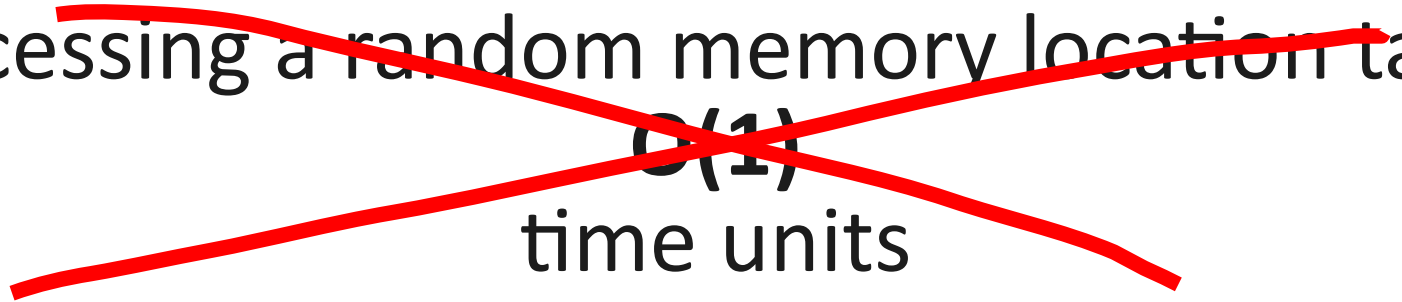
Accessing a random memory location requires
O(1)
operations

Accessing a random memory location takes
O(1)
time units

The Myth of RAM

Accessing a random memory location requires
O(1)
operations

Accessing a random memory location takes
O(1)
time units



The Myth of RAM

Accessing a random memory location requires

O(1)

operations

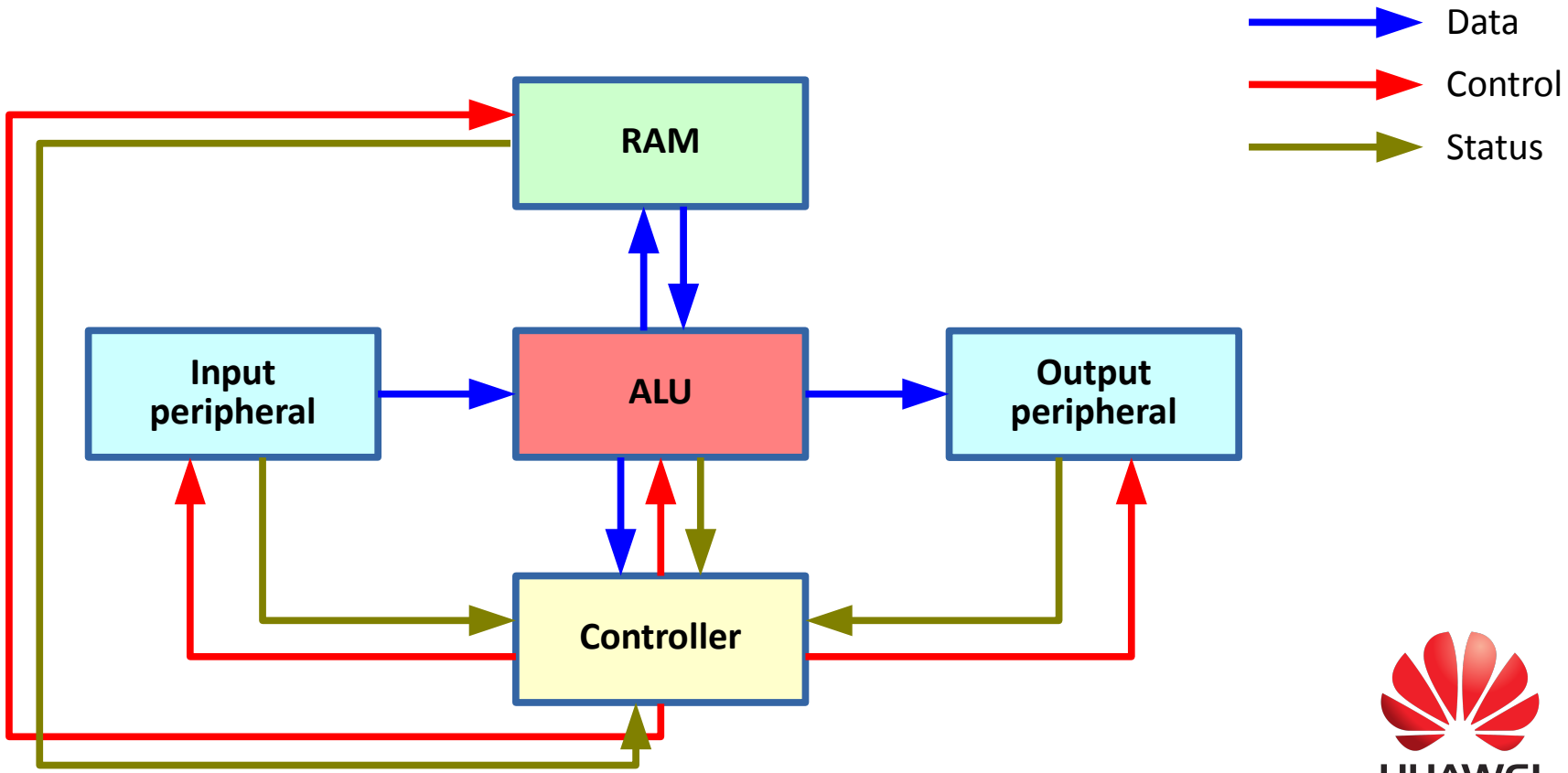
Accessing a random memory location takes

O($\sqrt{n}$)

time units

BREAKING THE BARRIER

Von Neumann Forever?



Von Neumann Forever?

- **New emerging memory technologies**
  - Bridging the gap between volatile and non-volatile memory
    - No longer necessary to keep the distinction between RAM and storage (peripherals)
      - Single-level memory (universal memory)
      - See also the talk by Liam Proven (*The circuit less traveled*), Janson, Sat 13:00
    - Many technologies in development
      - Magnetoresistive random-access memory (MRAM)
      - Racetrack memory
      - Ferroelectric random-access memory (FRAM)
      - Phase-change memory (PCM)
      - Nano-RAM (Nanotube RAM)

HUAWEI

Less Radical Solution

- **Near-Data Processing**
  - Moving the computation closer to the place where the data is
    - Not a completely new idea at all
      - Spatial locality in general
      - GPUs processing graphics data locally
    - Breaking the monopoly of CPU on data processing even more
      - CPUs are fast, but also power-hungry
      - CPUs can only process the data already fetched from the memory/storage
    - The more data we avoid moving from the memory/storage to the CPU, the more efficiently the CPU runs

HUAWEI

Near-Data Processing

- **Benefits**
  - Decreased latency, increased throughput
    - Not necessarily on an unloaded system, but improvement under load

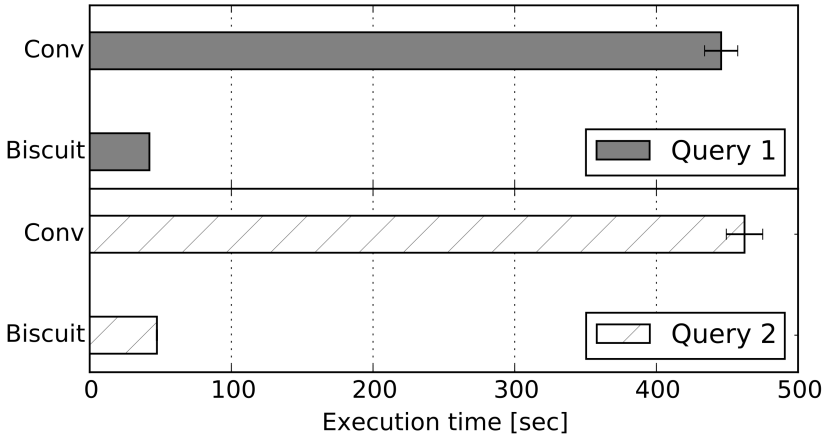


Figure 8. Performance of SQL queries on lineitem table.

[1] Gu B., Yoon A. S., Bae D.-H., Jo I., Lee J., Yoon J., Kang J.-U., Kwon M., Yoon C., Cho S., Jeong J., Chang D.: *Biscuit: A Framework for Near-Data Processing of Big Data Workloads*, in Proceedings of 43rd Annual International Symposium on Computer Architecture, ACM/IEEE, 2016

HUAWEI

Near-Data Processing

- **Benefits** (2)
  - Decreased energy consumption

Table 5: Comparison of normalized energy consumption.

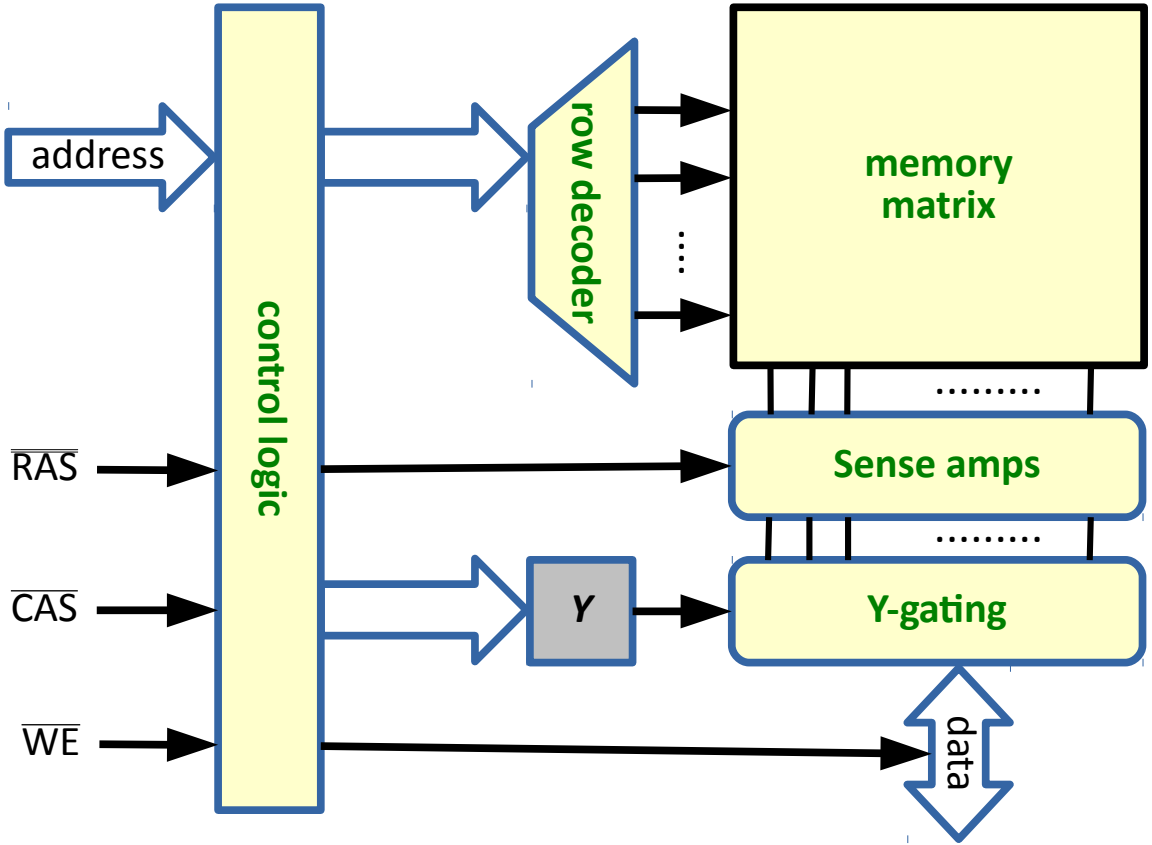
Processing method	Energy consumption
ISP (modified firmware)	0.142
IHP (conventional)	1.000

[2] Kim S., Oh H., Park C., Cho S., Lee S.-W.: *Fast, Energy Efficient Scan inside Flash Memory SSDs*, in Proceedings of 37th International Conference on Very Large Data Bases (VLDB), VLDB Endowment, 2011

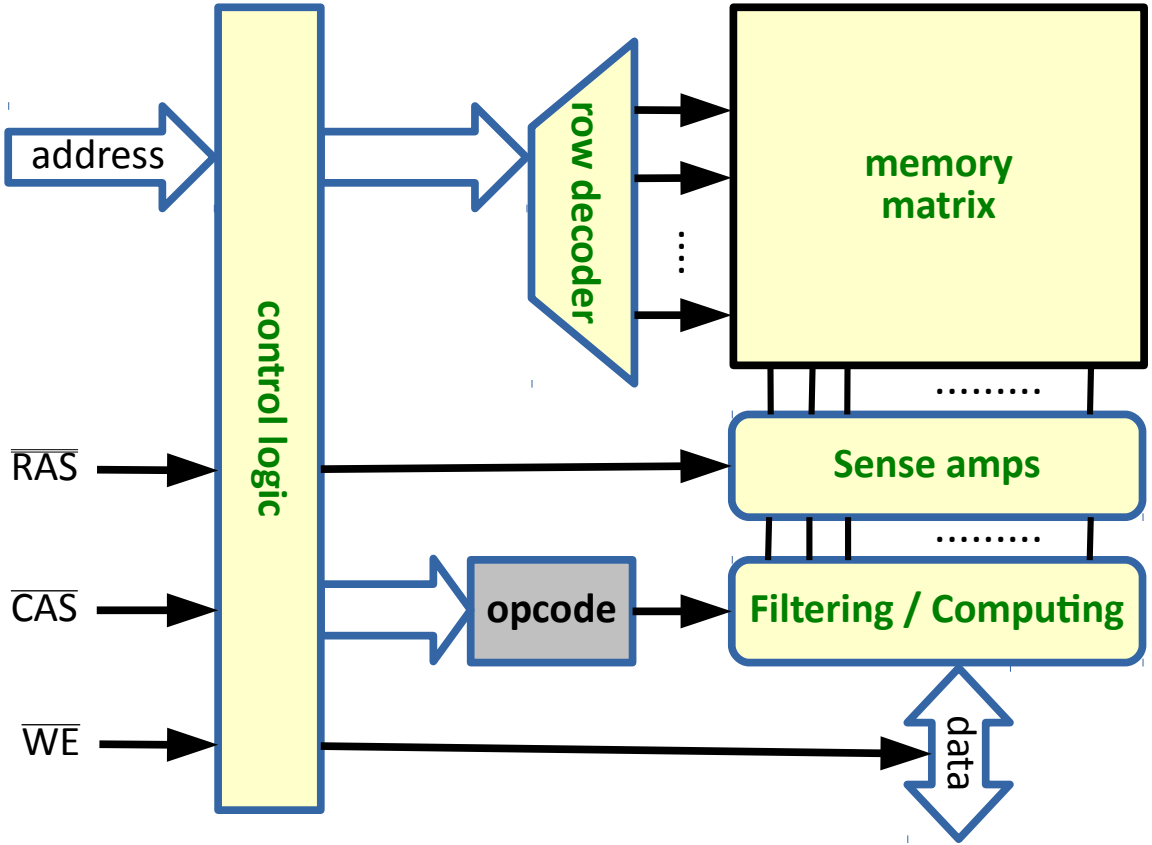
In-Memory Near-Data Processing

- **Adding computational capability to DRAM cells**
  - Simple logical comparators/operators that could be computed in parallel on individual words
    - Filtering based on bitwise pattern
    - Bitwise operations
  - Making use of the inherent parallelism
    - Avoiding moving unnecessary data out of DRAM
    - Avoiding linear processing of independent words of the data in CPU

Dynamic RAM


address
control logic
row decoder
memory matrix
RAS
CAS
WE
Sense amps
Y
Y-gating
data

Dynamic RAM with NDP



In-Storage Near-Data Processing

- **Adding computational capability to SSD controllers**
  - Again, making use of inherent parallelism of flash memory
  - But SSD controllers also contain powerful embedded cores
    - Flash Translation Layer, garbage collection, wear leveling
    - Thus computation is not limited to simple bitwise filtering and operations
  - Complex trade-offs between computing on the primary CPU and offloading to the SSD controller

Our Prototype

- **Based on OpenSSD**
  - http://openssd.io/
    - Open source (GPL) NVMe SSD controller
      - Hanyang University (Seoul), Embedded and Network Computing Lab
    - FPGA design for Xilinx Xynq-7000
      - ONFI NAND flash controller with ECC engine
      - PCI-e NVMe host interface with scatter-gather DMA engine
    - Controller firmware
      - ARMv7
      - Flash Translation Layer, page caching, greedy garbage collection

Our Prototype (2)

- **NVMe NDP extensions**
  - NDP module deployment
    - Static native code so far, moving to safe byte-code (eBPF)
  - NDP datasets
    - Safety boundaries for the NDP modules (for multitenancy, etc.)
  - NDP Read / Write
    - Extensions of the standard NVMe Read / Write commands
    - NDP module executed on each block, transforms/filters data
  - NDP Transform
    - Arbitrary data transformations (in-place copying, etc.)
    - Flow-based computational model

HUAWEI

Our Prototype (3)

- **Fast prototyping**
  - QEMU model of the OpenSSD hardware (for running the OpenSSD firmware on ARMv7)
  - Connected to a second host QEMU/KVM (as a regular PCI-e NVMe storage device)
- **Planned evaluation**
  - Real-world application proof-of-concept
    - Custom storage engine for MySQL with operator push-down
    - Ceph operator push-down
    - Key-value store
    - Generic file system acceleration
  - Still a very much work-in-progress
    - Preliminary results quite promising

HUAWEI

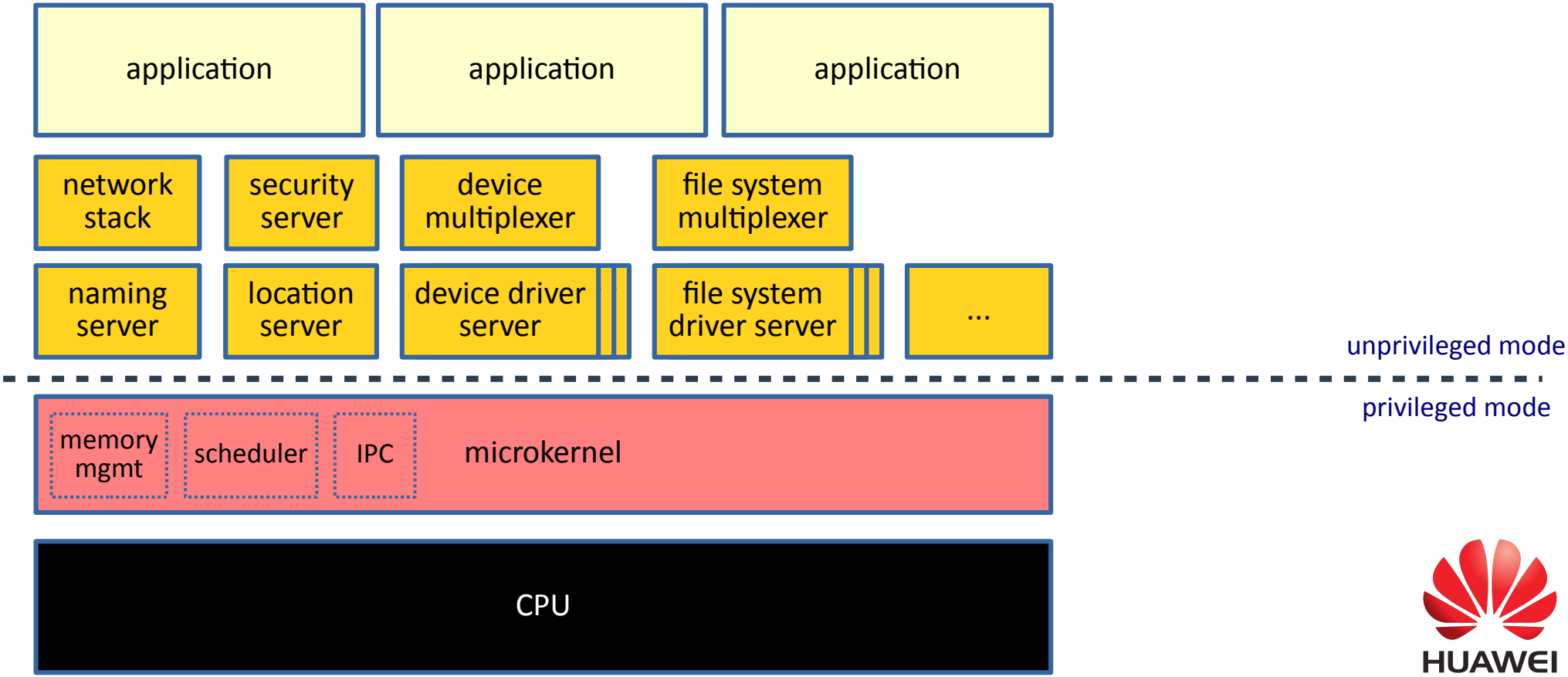
HOW MICROKERNELS FIT INTO THIS?

Future Vision

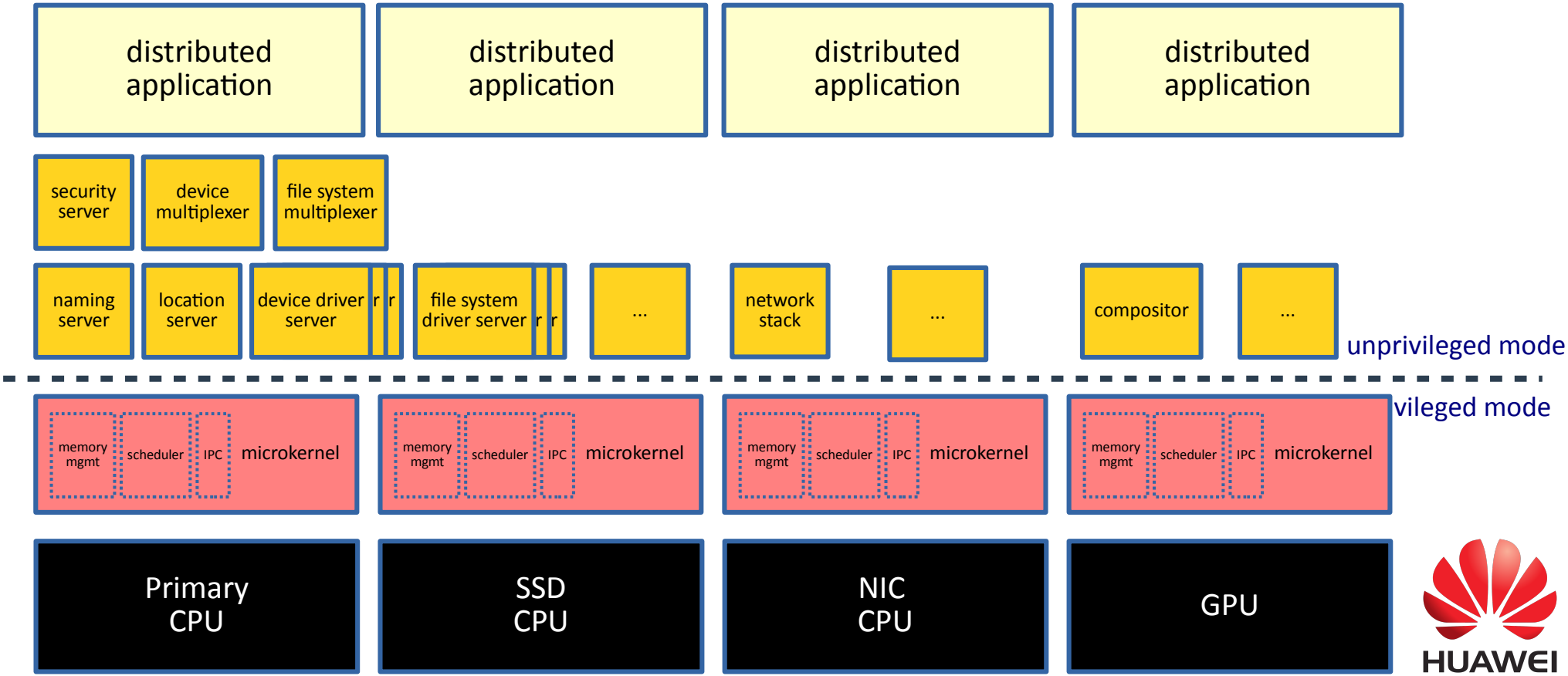
- **Not just near-data processing, but data-centric computing**
  - As opposed to CPU-centric computing
    - Running the computation dynamically where it is the most efficient
    - Not necessarily moving the data to the central processing unit
      - The CPU is the orchestrator
  - Massively distributed systems
    - Within the box of your machine (desktop, laptop, smartphone)
    - Outside your machine (edge cloud, fog, cloud, data center)
  - Massively heterogeneous systems
    - Different ISAs
    - Fully programmable, partially programmable, fixed-function

HUAWEI

Future Vision & Microkernels



Future Vision & Multi-Microkernels



Microkernels for Data-Centric Computing

- **Multiserver microkernels**
  - Naturally support distribution and heterogeneity
    - IPC is a well-defined mechanism to communicate across isolation boundaries
      - Communicating between multiple (heterogeneous) computing cores is just a different transport
      - Message passing as the least common denominator
        - But also memory coherency emulation is possible
    - Multiserver designers are used to think in terms of fine-grained components
      - Composing an application from multiple components
      - Missing piece: Framework for automatic workload migration

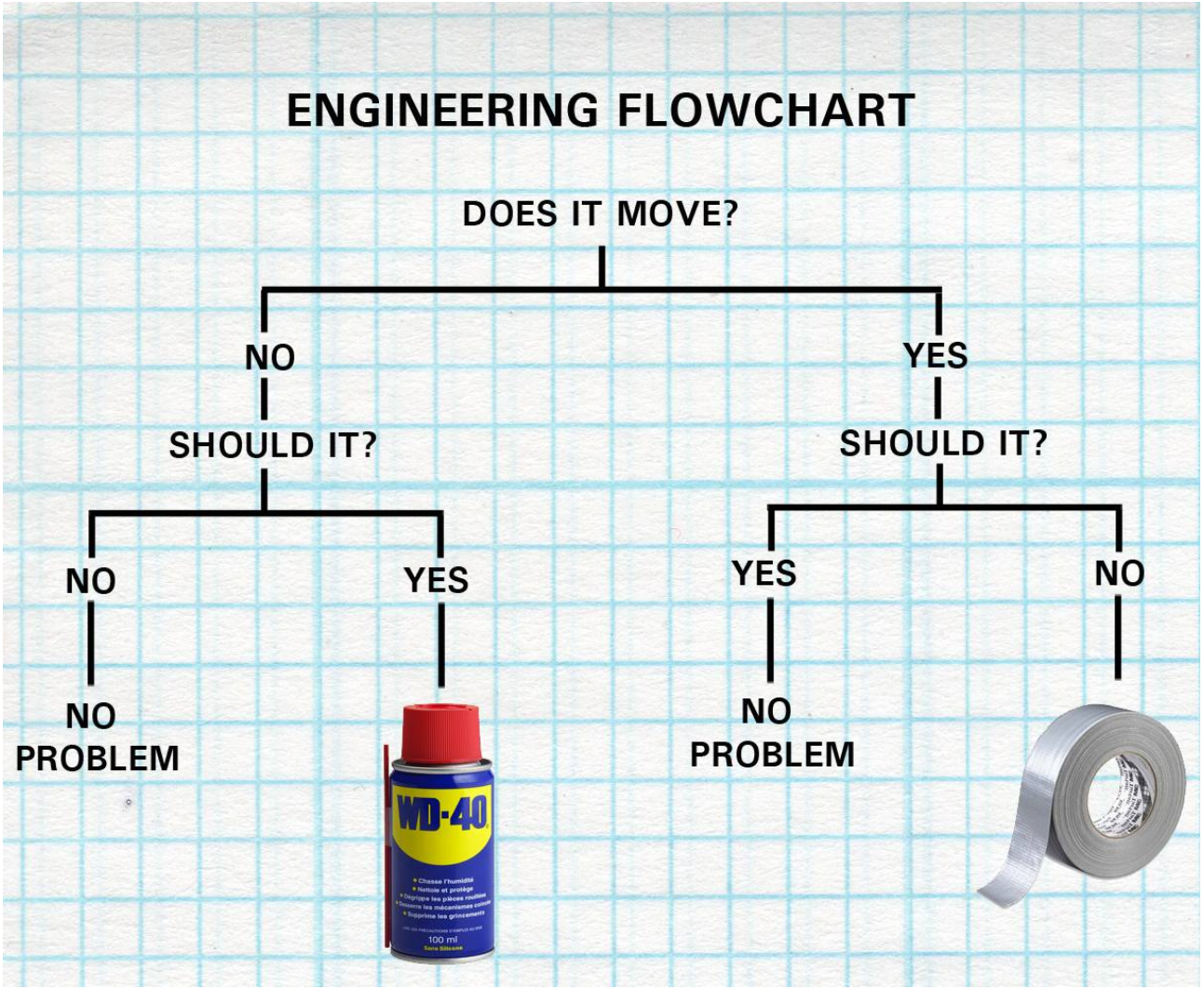
HUAWEI

Microkernels for Data-Centric Computing (2)

- **Why not multikernels?**
  - Yes, why not!
  - Many multikernels (i.e. Barrelfish) are essentially microkernels to the extreme
    - Each core running its independent microkernel
    - No memory coherency assumed
- **Why not unikernels?**
  - OK, if you insist
  - A microkernel running a single fixed payload (e.g. a virtual machine) could be considered a unikernel

Road Forward

- **As usual, incremental steps**
  - Start with just the basic workload off-loading to the SSD, NIC, etc.
  - Gradually transform the run-time for the workload off-loading into a proto-microkernel
  - Consider the dynamic workload migration framework
  - Gradually move to a full-fledged multi-microkernel


ENGINEERING FLOWCHART
DOES IT MOVE?
NO
YES
SHOULD IT?
SHOULD IT?
NO
YES
YES
NO
NO PROBLEM
NO PROBLEM
WD-40

Summary

- **There are potential ways of abandoning the von Neumann architecture**
  - Revolutionary: Single-level (universal) memory
  - Evolutionary: Near-Data Processing
- **Near-Data Processing is beneficial in terms of performance and energy consumption**
- **Data-centric computing leads to massively distributed and heterogeneous systems**
  - Multi-microkernels are a natural match for such architectures

HUAWEI

Acknowledgements

- **Huawei Technologies in-storage NDP team**
  - Antonio Barbalace
  - Martin Decky
  - Anthony Iliopoulos
  - Javier Picorel
  - Dmitry Voytik
  - Hangjun Zhao
- **References & Inspiration**
  - Mark Silberstein (Technion)
  - Biscuit framework (Samsung Electronics)

Q&A


HUAWEI
THANK YOU!