Imperial College London

# Scalable Systems for the Cloud

Fall 2018

Rack-scale Computing

**Instructor:** Dr. Jana Giceva

# What is a rack?

- How does it fit within a data-centre?
- What does it consists of? Which resources?
- How do we build it?
- Example rack-computers

# Rack-scale

- What is a rack?

# Rack-scale

- What is a rack?
  - The rack is the new unit of deployment in data centres
  - Sweet spot between a single-server and cluster deployments
  - It has 42 units (rack-units – RU) that host the compute resources

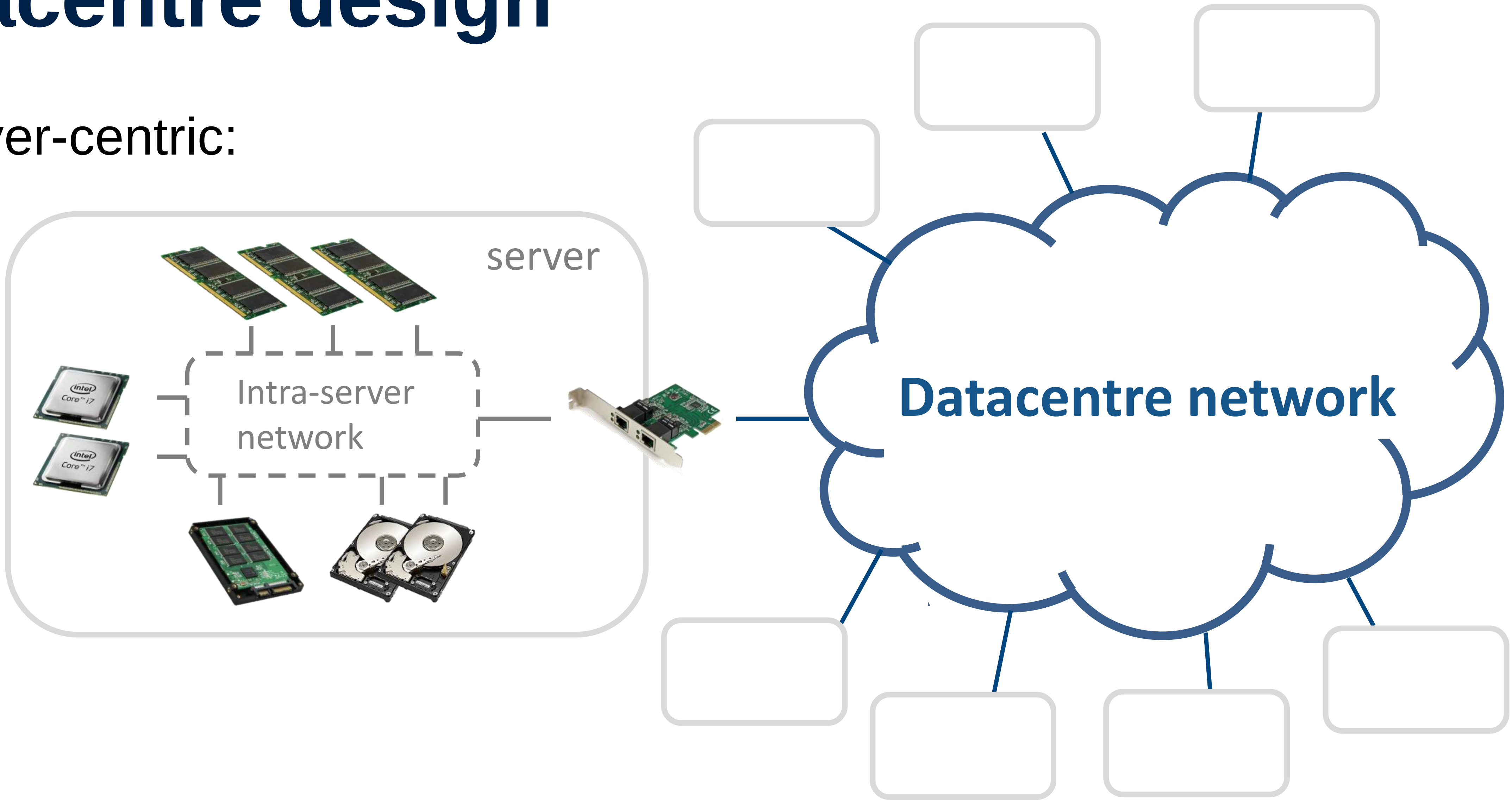
# What’s in a Rack-scale computer?

- Rack-scale computer (pre-packaged)
- Compute:
  - standard compute
  - accelerators
- Storage:
  - hot / warm / cold disks
- Networking:
  - interconnect
  - software defined networking


Image source: Supermicro RSD

# From server-centric to resource-centric datacentre design

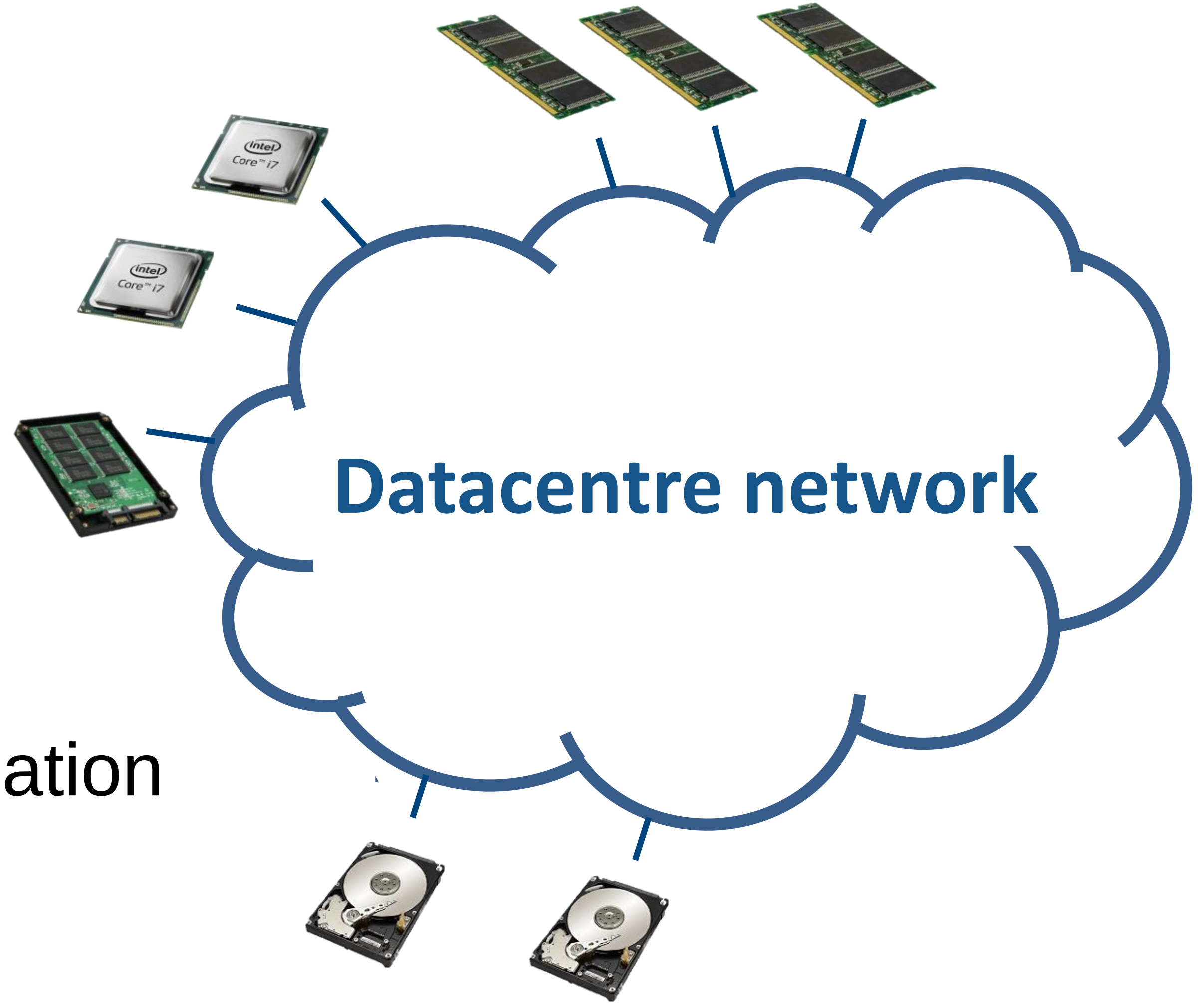
- Server-centric:



# From server-centric to resource-centric datacentre design

- Towards resource-centric

- **Past:** physical aggregation
  - shared power, cooling, rack-management
- **Now:** fabric integration
  - fast ***rack***-wide ***interconnect***
- **Future goal:** resource disaggregation
  - pooled compute, storage, memory resources



# Today's scale within a rack computer

- We already have *scale* within a rack itself.

| Machine | Core count |
|---|---|
| AMD SeaMicro *SM15000-64* | 2'048 |
| HP Moonshot *Redstone* | 11'520 |
| Boston Viridis | 7'680 |

| Machine | Memory |
|---|---|
| AMD SeaMicro *SM15000-XE* | 8 TB |
| HP Moonshot *Redstone* | 11.25 TB |

| Machine | Network |
|---|---|
| EDR Mellanox | 100 Gbps |
| Intel silicon photonics | 100-400 Gbps |

- And increasing *heterogeneity* of resources

**AMD Rack P47 – 1 PetaFLOP of compute at FP32 single precision**

| CPU | GPU | Memory | Network |
|---|---|---|---|
| 20x AMD EPYC 7601 | 80x Radeon Instinct | 10 TB DDR4 | 2x36 port EDR switch (100 Gbps) |

# Future: heterogeneous computing resources across the rack

- Accelerators
- Co-processors

- Intelligent storage
- Intelligent (active) memory
- Smart NICs
- In-network data processing

# Rack-scale *computing*

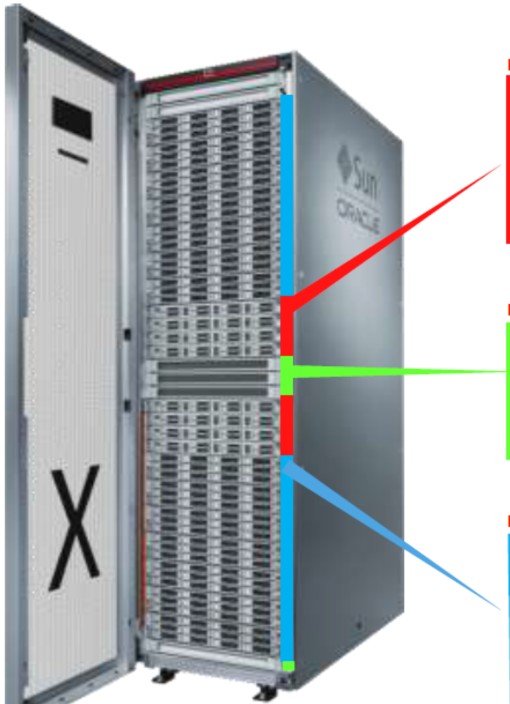
- How do we implement applications for a rack computer?
- How do we manage these resources?
- What is the failure model? How do we achieve fault tolerance?

# Data appliances – among the first rack-scale apps

Oracle's Exadata
rack-scale data analytics
engine since 2008

## Same Exadata Architecture

Complete | Optimized | Standardized | Hardened Database Platform



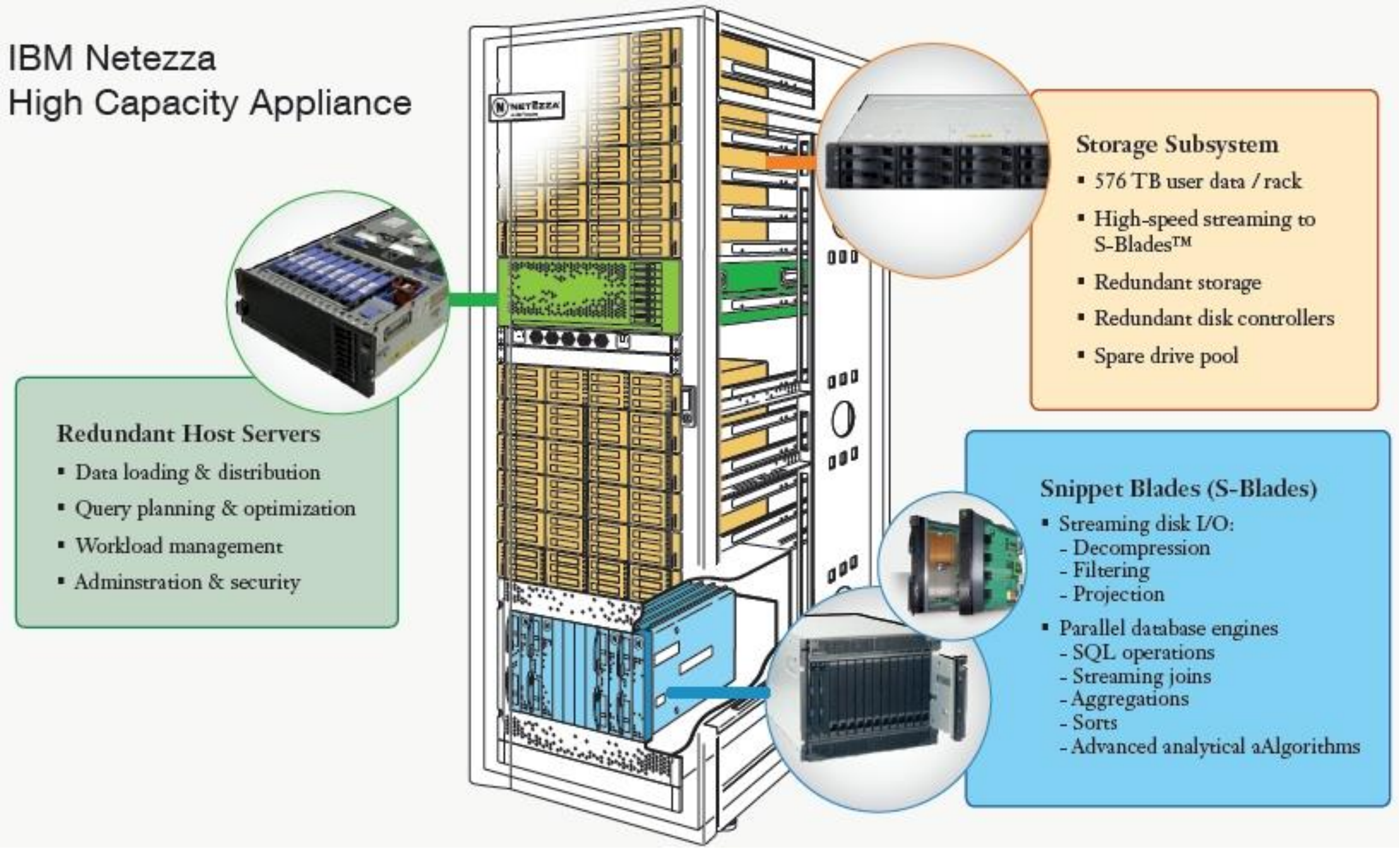
Fully Redundant

- **Standard Database Servers**
  - 8x 2-socket servers → 192 cores, 2TB DRAM
    or
  - 2x 8-socket servers → 160 cores, 4TB DRAM


- **Unified Ultra-Fast Network**
  - 40 Gb InfiniBand internal connectivity → all ports active
  - 10 Gb or 1 Gb Ethernet data center connectivity
- **Scale-out Intelligent Storage Servers**
  - 14x 2-socket servers → 168 *faster* cores in storage
  - 168 SAS disk drives → 672 TB HC or 200 TB HP
  - 56 Flash PCI cards → 44 TB Flash + compression

ORACLE

# Data appliances – among the first rack-scale apps



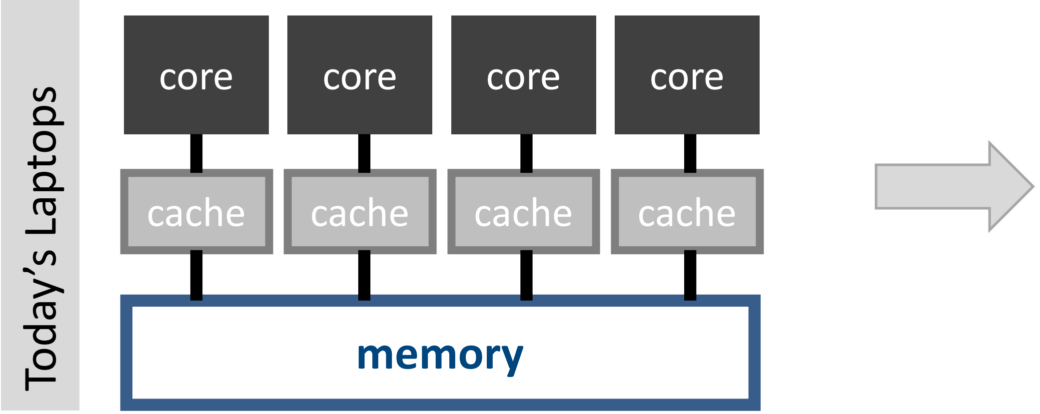

IBM Netezza
Heterogeneous appliance incorporating FPGA blades

Figure from 2011

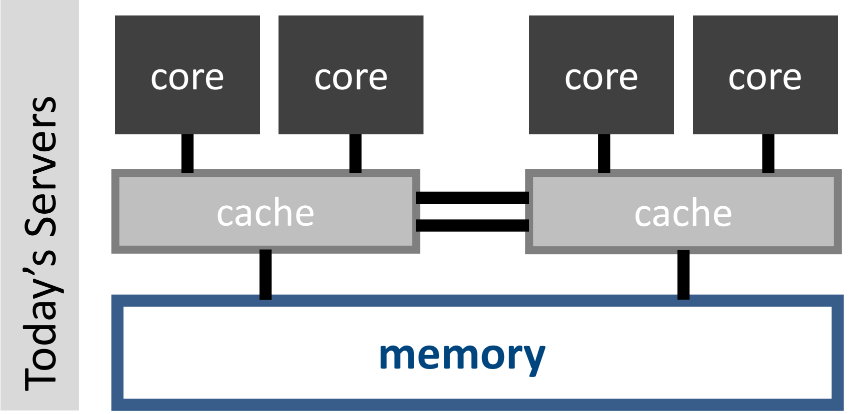
# How do we program with remote memory?

# Parallel architectures

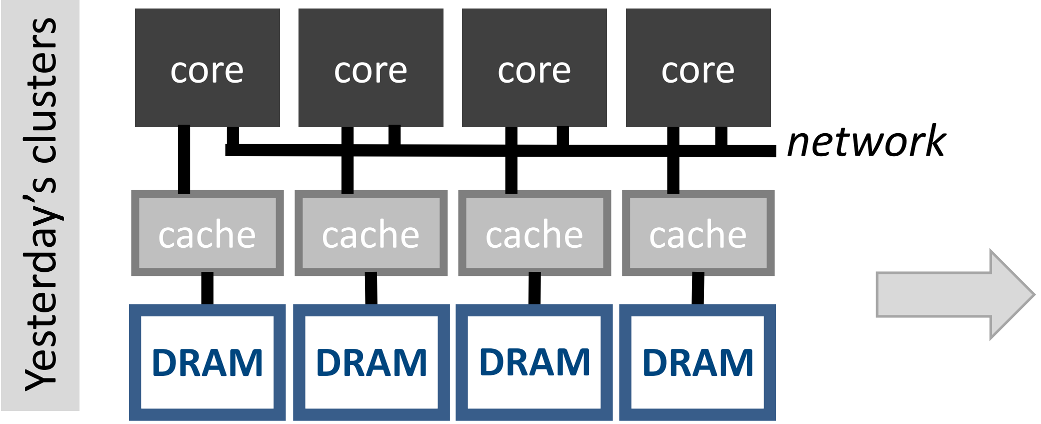
*Uniform memory access (UMA)*



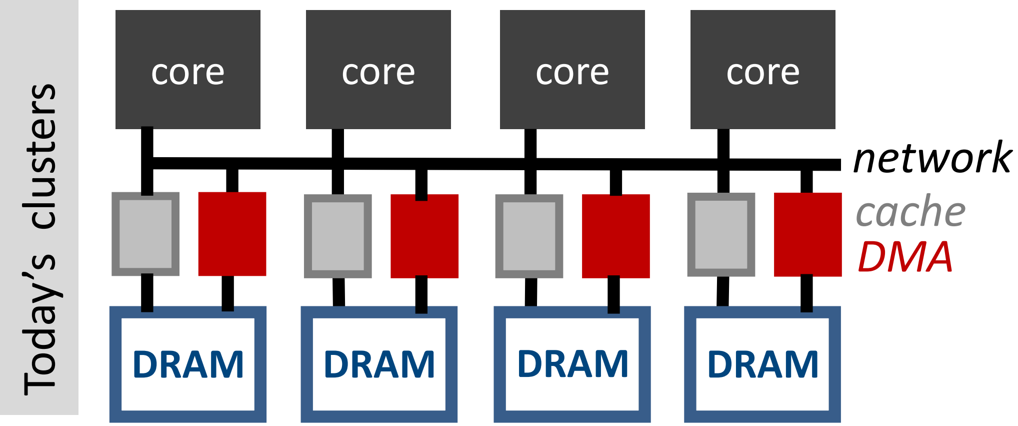

*Non-uniform memory access (NUMA)*




*Time-division multiplexing*




*Remote direct-memory access*

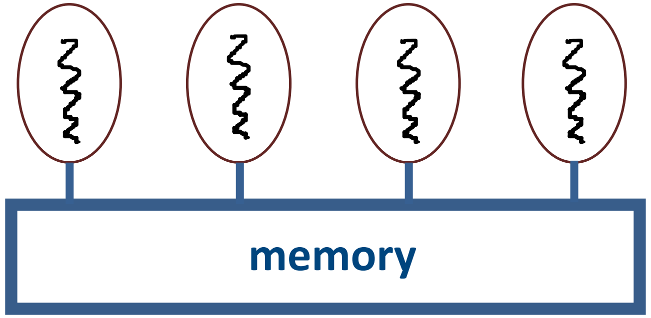


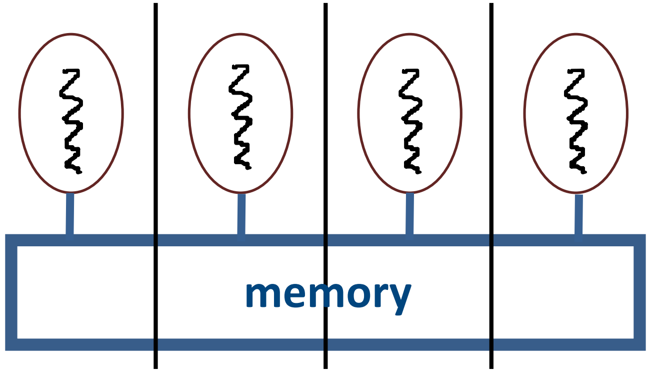
# Programming models

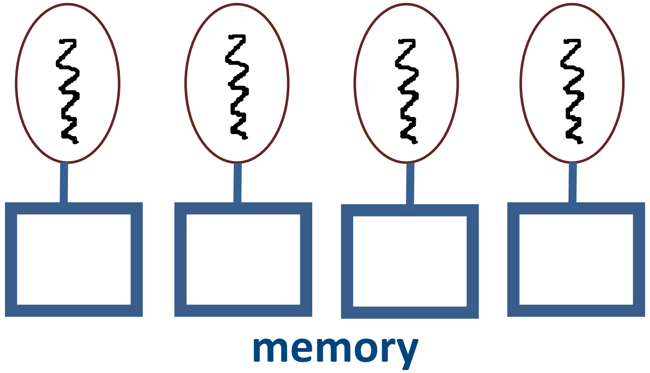
- Shared memory programming
  - shared address space
  - implicit communication
  - cache-coherent NUMA
  - e.g., pthreads or OpenMP

- (Partitioned) global address space
  - Remote Memory Access
  - Remote vs. local memory (e.g., ncc NUMA)

- Distributed memory programming
  - Explicit communication (e.g., with messages)
  - Message passing



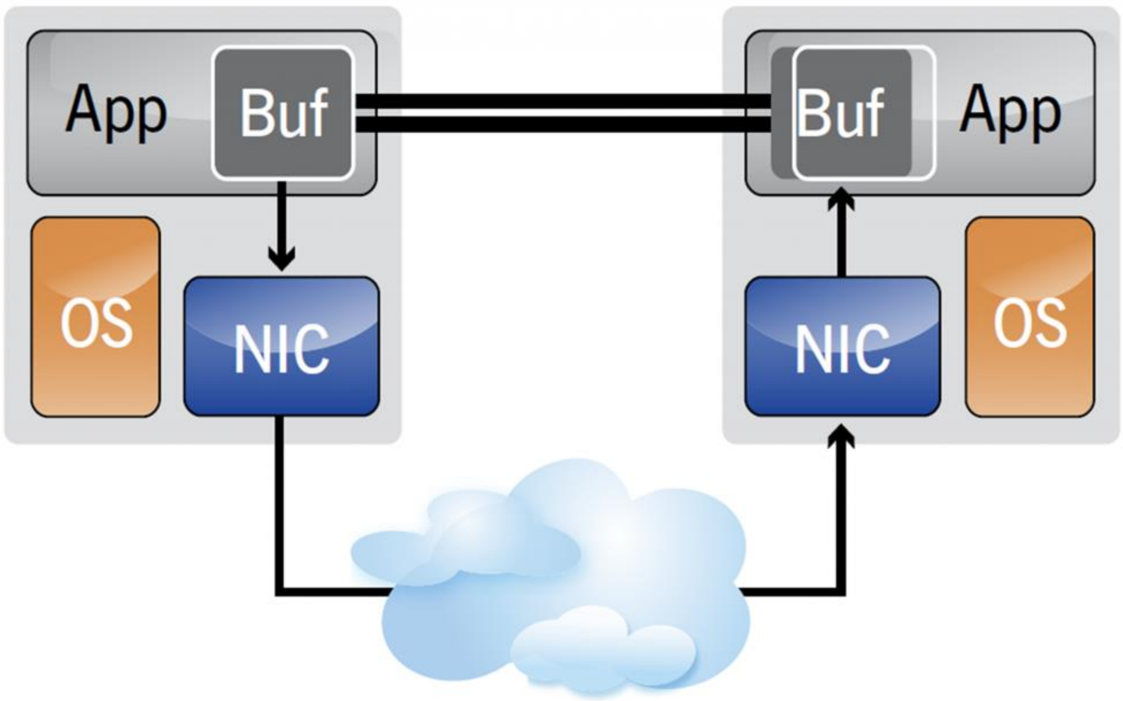



# A popular approach – RDMA

- What is RDMA?

  Remote Direct Memory Access

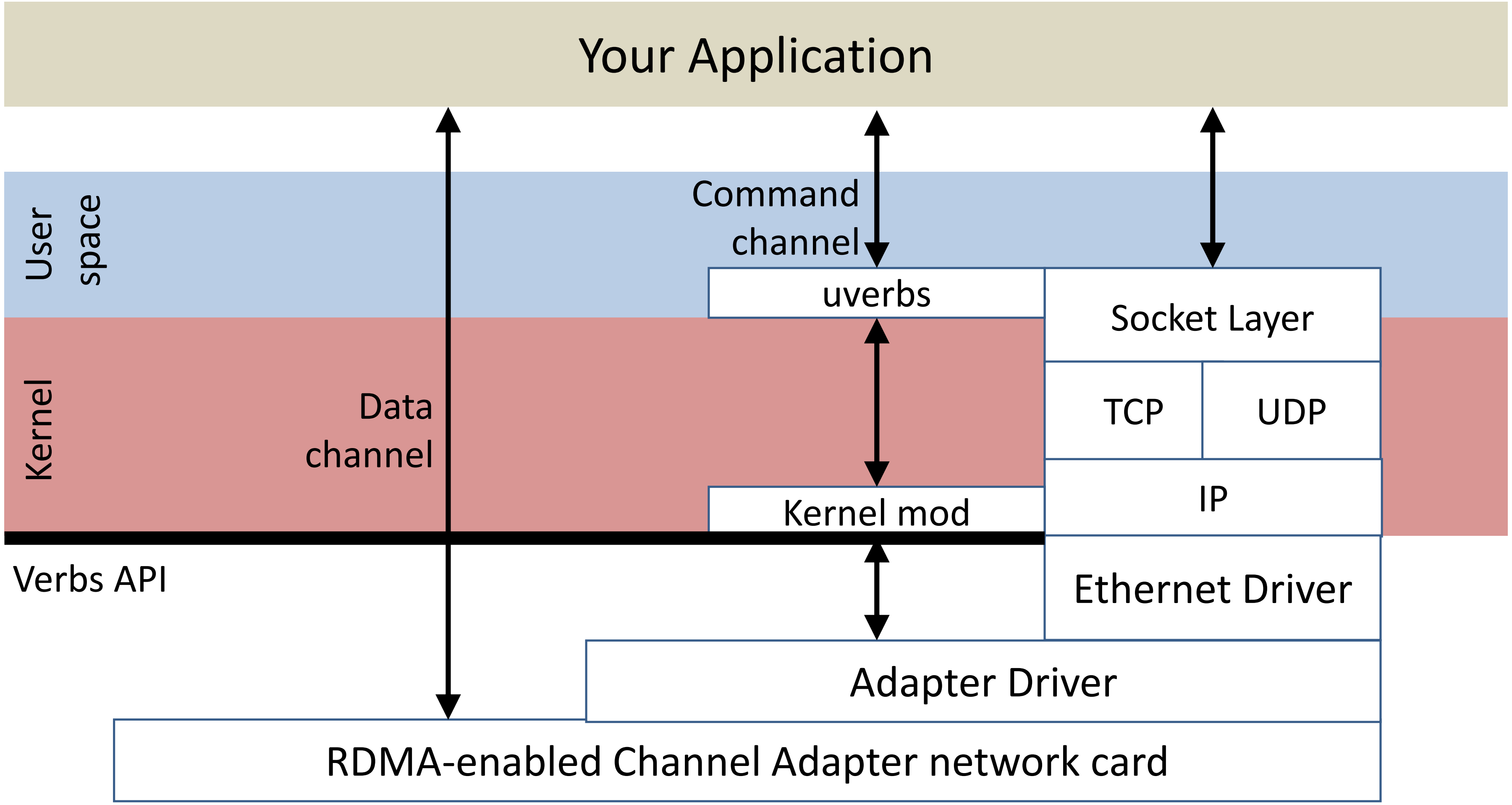
- RDMA is a hardware mechanism through which the network card can directly access all or parts of the main memory of a remote node without involving the processor.



# RDMA properties

- Bypass the CPU → low CPU utilization
- Bypass the OS kernel → no interrupts, no context switching
- Zero-copy data → low memory bus contention
- Message based transactions
- Asynchronous operations → overlap communication and computation

# Traditional TCP/IP sockets vs RDMA




src: InfiniBand Trade Association: Introduction to IB for end users

# “Expanding” the Memory hierarchy

registers

10’000 Gbps, 1ns

CPU cache

500 Gbps, 50 ns

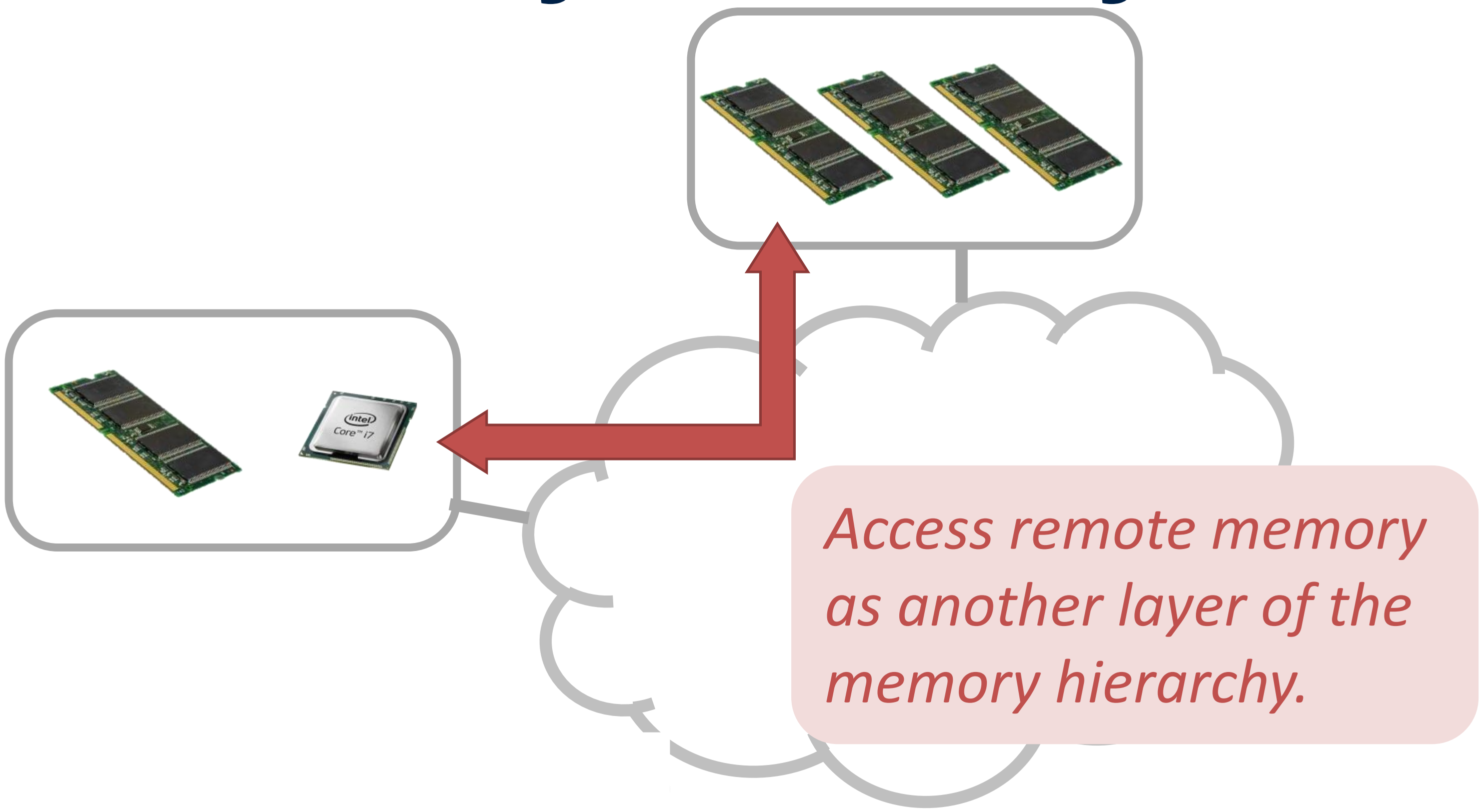
Local memory

100 Gbps, 5’000 ns

Remote memory

1-10 Gbps, 50’000 ns

SSD / HDD




Microsoft Research showed that using Remote Memory (and RDMA) improves the latency of TPC-H and TPC-DS queries by 2-100x

Li et al. [SIGMOD 2016]

# RDMA in research

***High Performance Computing*** is the home research domain for RDMA

## *Databases*

- ***Distributed transactions***
  FaSST [OSDI'16], FaRM [NSDI'14, SOSP'15], DrTM [SOSP'15], Tell [SIGMOD'15], NAM-DB [VLDB'17]
- ***RDMA KV-stores***
  RAMCloud [FAST'11, SOSP'11, SOSP'15], HERD [SIGCOMM'14], Pilaf [ATC'13]
- ***Distributed join processing***
  Barthels et al. [SIGMOD'15], Frey et al. [ICDCS'10], Rödiger et al. [ICDE'16]
- ***Accelerating RDBMS with RDMA***
  Li et al. [SIGMOD'16], BatchDB [SIGMOD'17]

## *Operating Systems*

- Data-centres / Rack-scale computing: LITE [OSDI'17]

# Efficient way to access remote storage?

# What about remote storage?

- Traditionally:
  - Accessing remote storage requires traversing the whole system stack.
  - But, hardware and software latencies are additive.
- Future:
  - Intelligent storage
  - BlueDBM [ISCA'15]
  - Ibex [VLDB'14]