

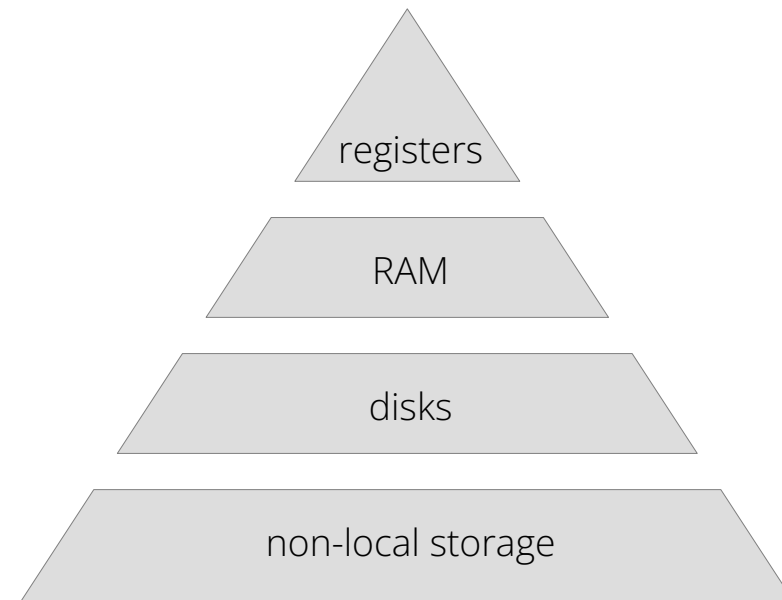


# First Things First: A Discussion of Modelling Approaches for Disruptive Memory Technologies

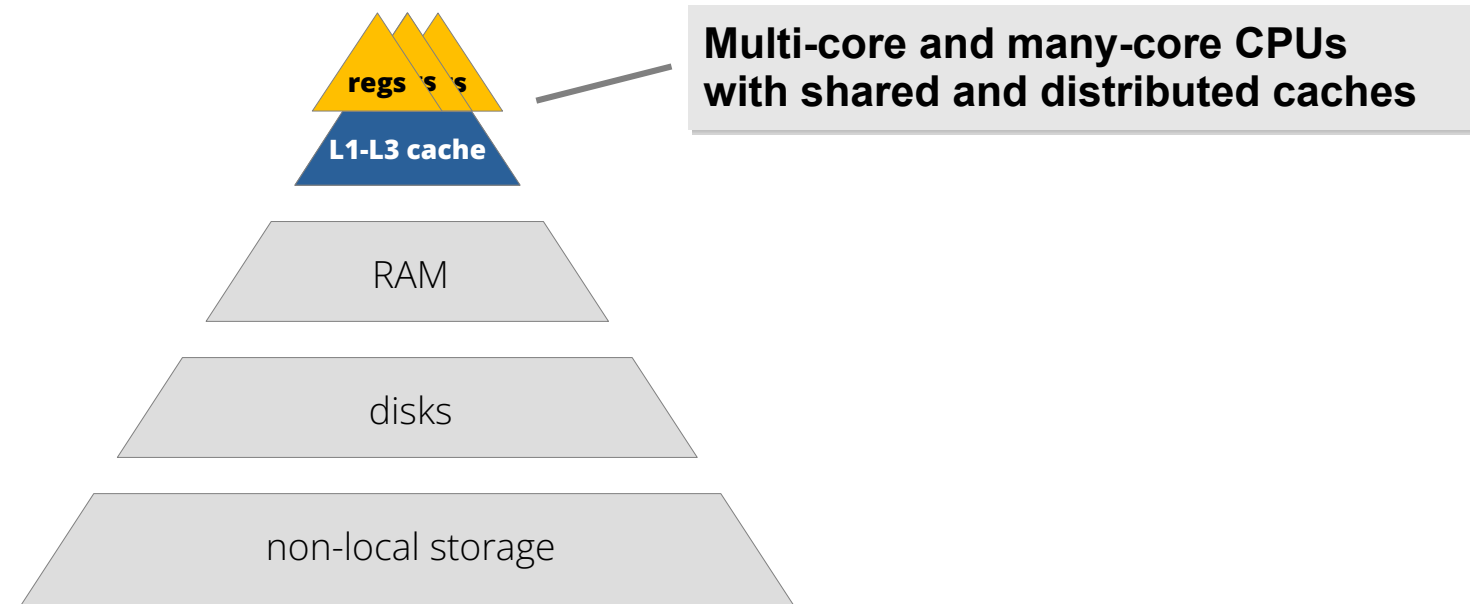
Herbsttreffen der Fachgruppe Betriebssysteme (FGBS'21), Trondheim (Online), 20.09.21

**Michael Müller**, Daniel Kessener, Olaf Spinczyk

# The complexity of modern memory hierarchies



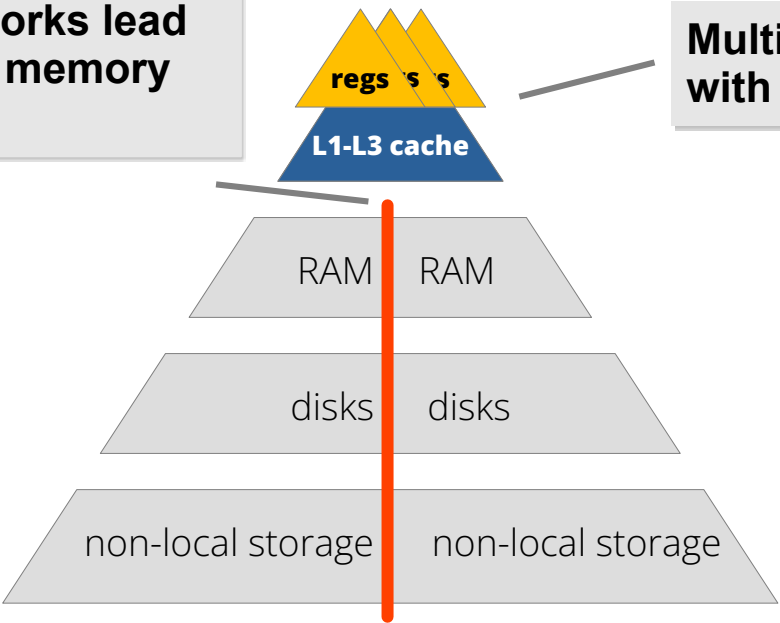
# The complexity of modern memory hierarchies



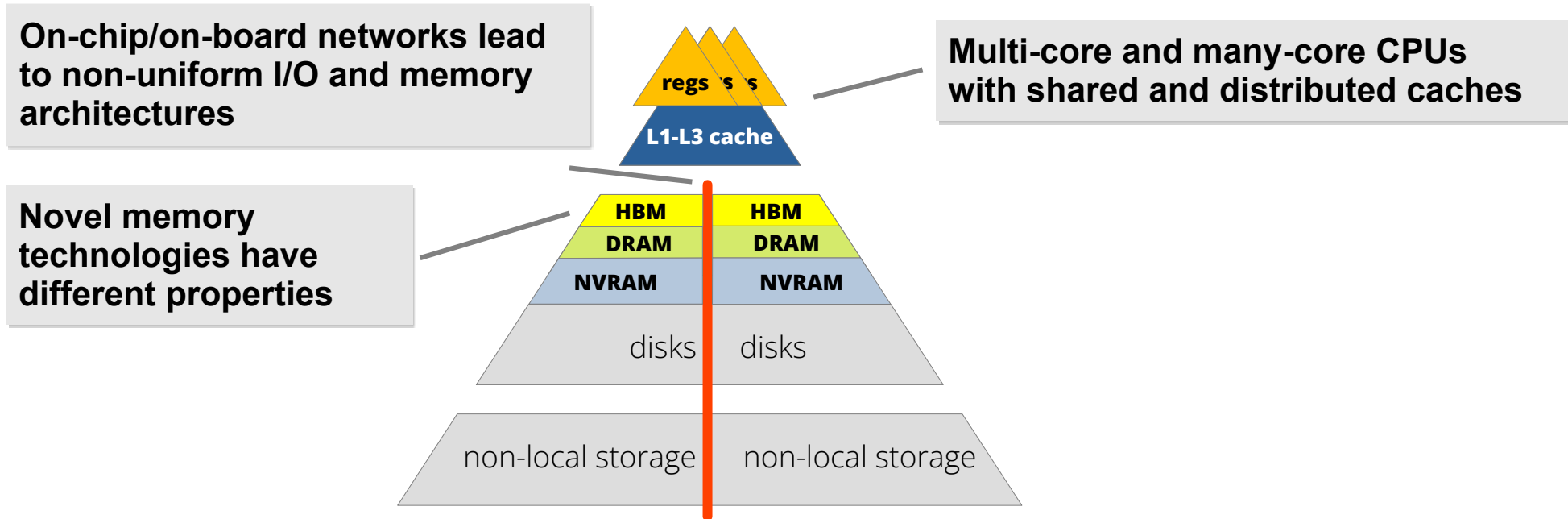
# The complexity of modern memory hierarchies

On-chip/on-board networks lead to non-uniform I/O and memory architectures

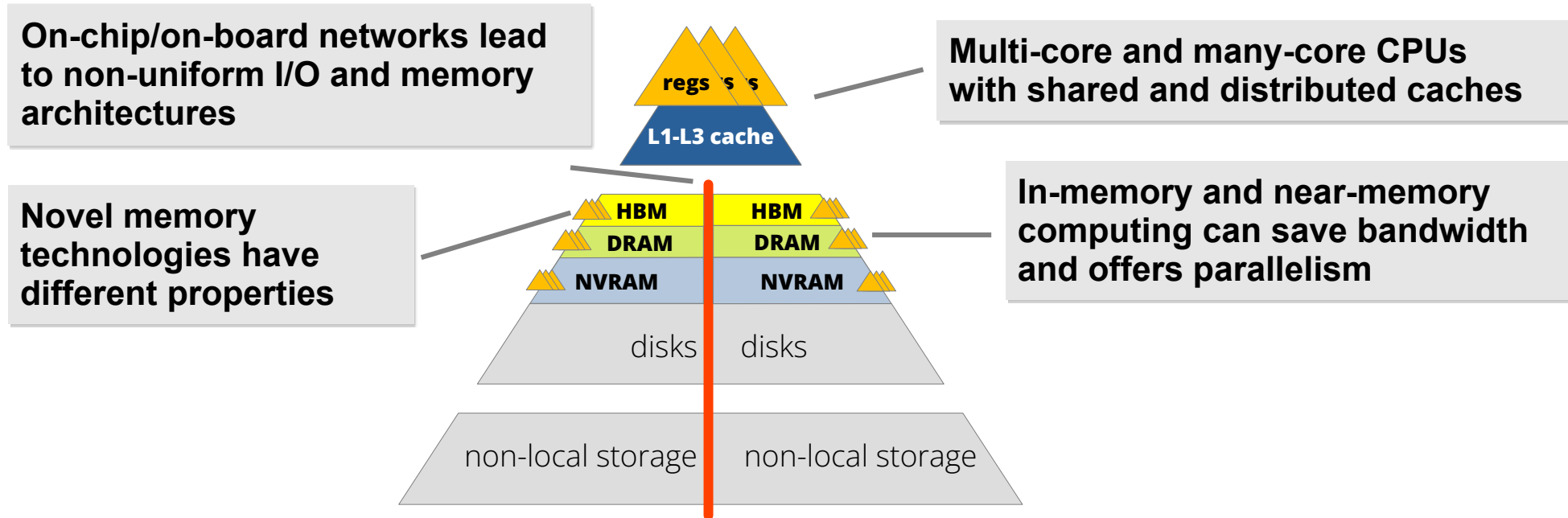
Multi-core and many-core CPUs with shared and distributed caches



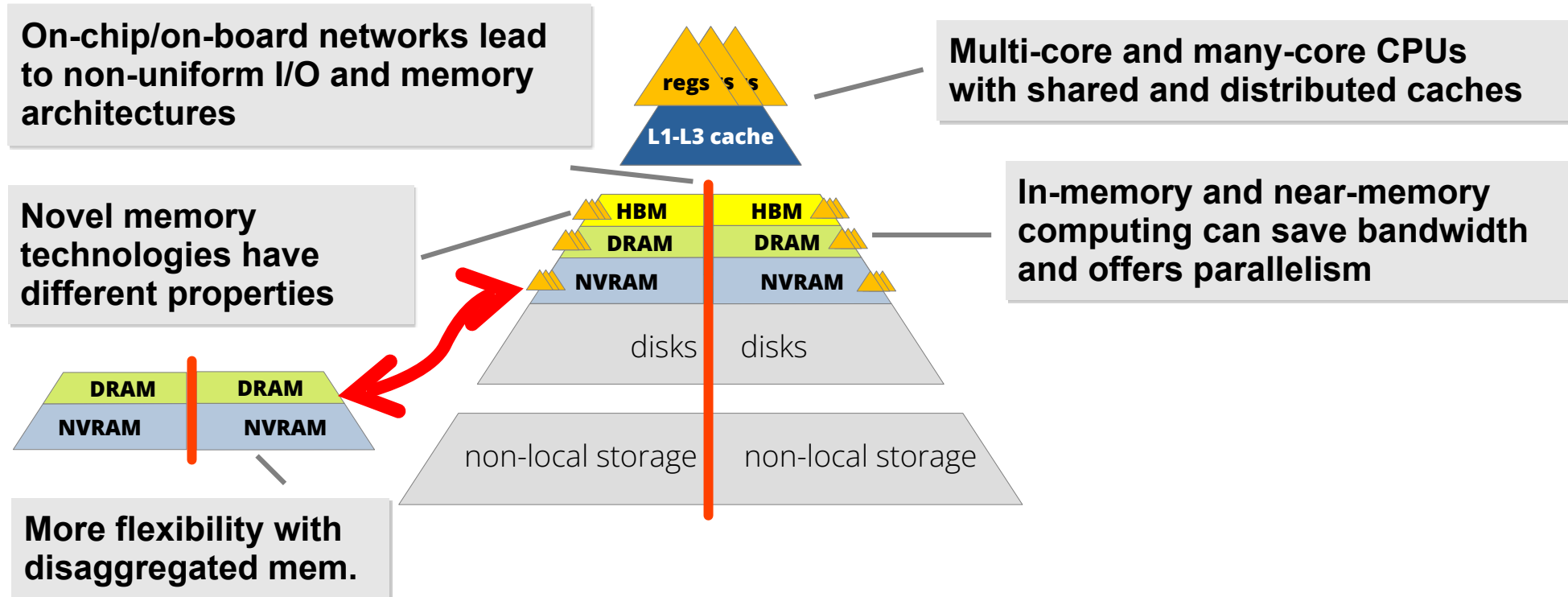
# The complexity of modern memory hierarchies



# The complexity of modern memory hierarchies

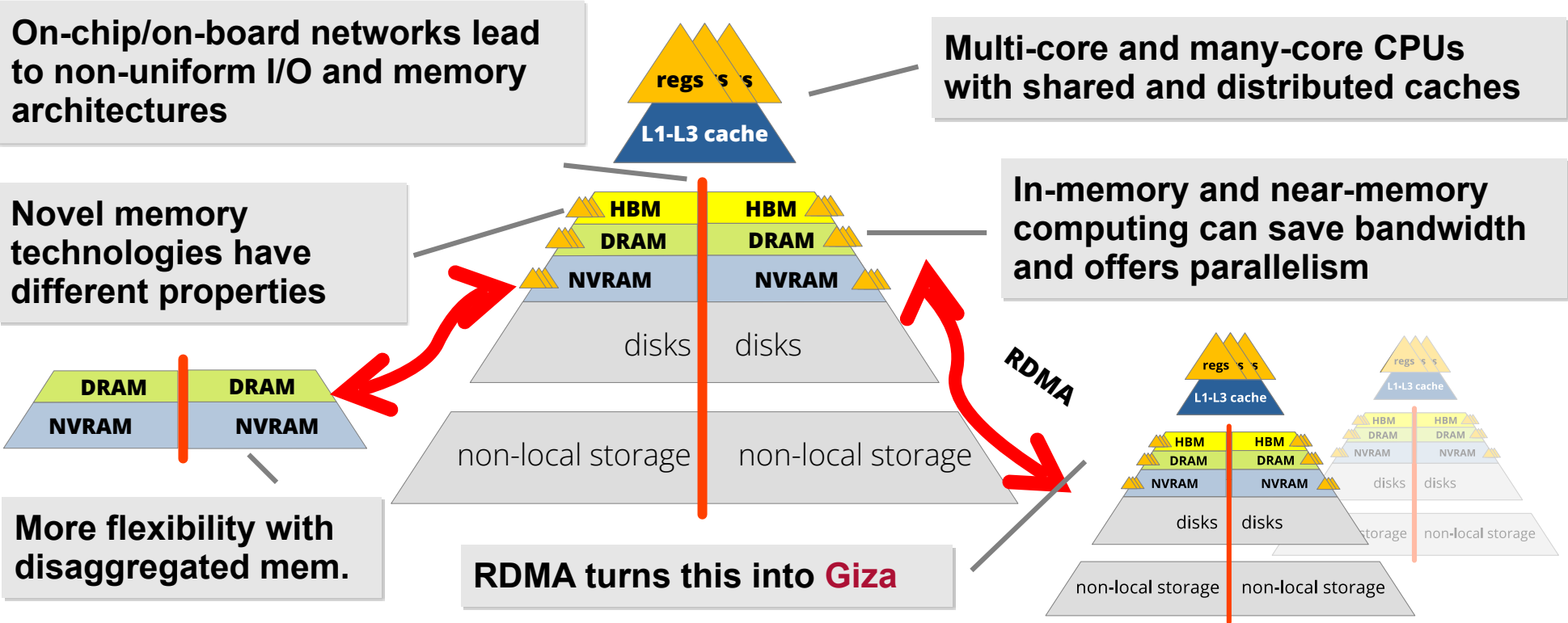


# The complexity of modern memory hierarchies





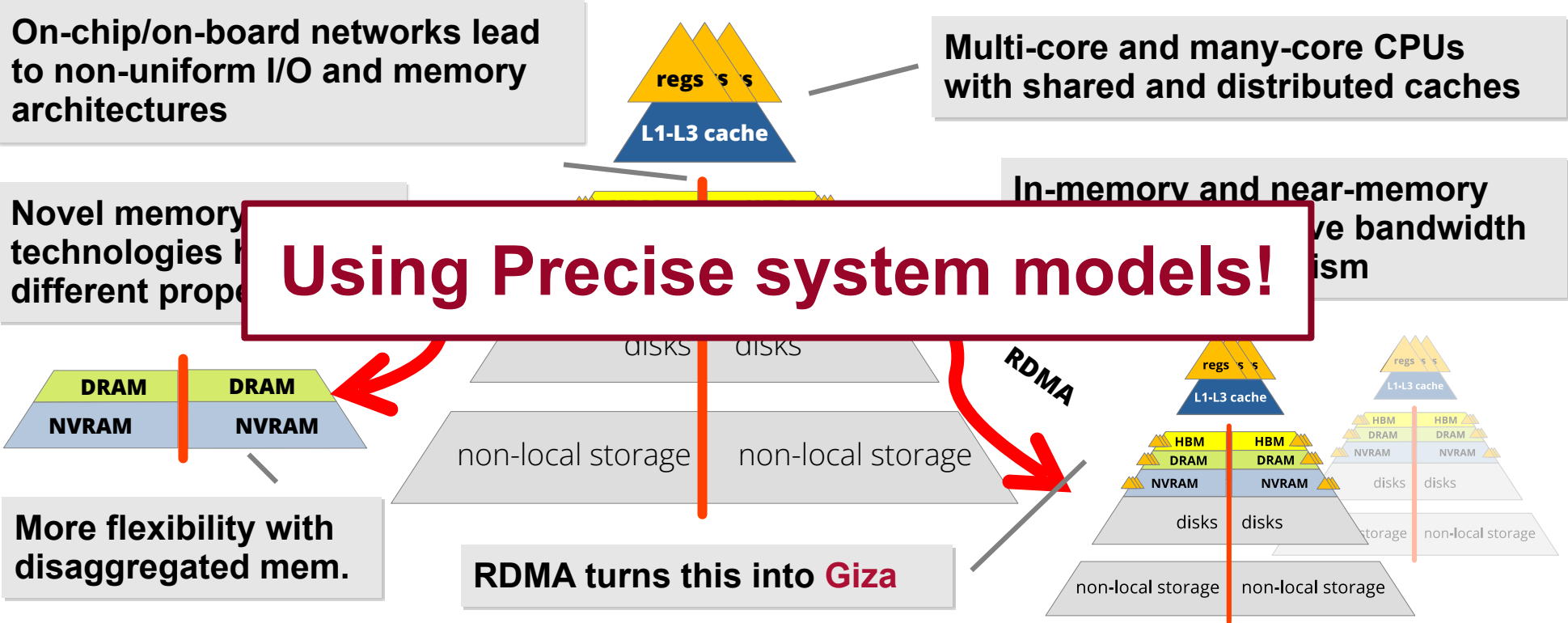
# The complexity of modern memory hierarchies



How could *any* system software efficiently manage these resources?



# The complexity of modern memory hierarchies



How could *any* system software efficiently manage these resources?

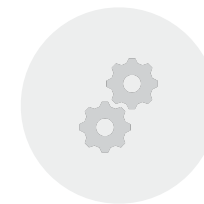
# An adequate system-level model



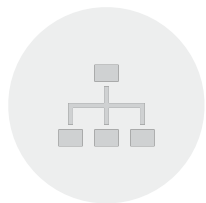
Represents the **overall hardware topology** including **DMTs**



Provides information about **communication costs**



Enables **updates** to the model **at runtime**



has a **whole system view** including **applications** and system services, i.e., an **application model**



Provides **performance predictions** and **optimized resource mappings**

# Performance metrics and guidelines for RDMA

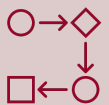
# Performance metrics and guidelines for RDMA



**Comprehensive work on “best practises” and guidelines for performance optimization**



Analyzes on using RDMA with NVM an in NUMA systems



Management model for scheduling RDMA transfers

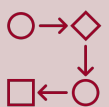
# Performance metrics and guidelines for RDMA



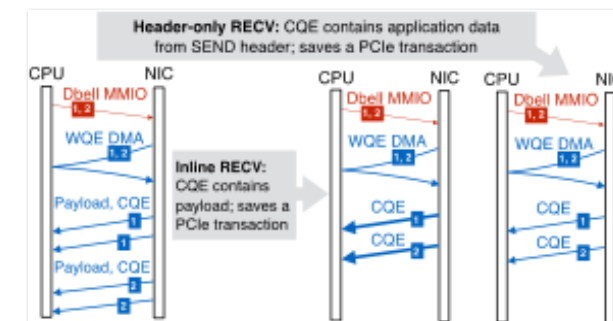
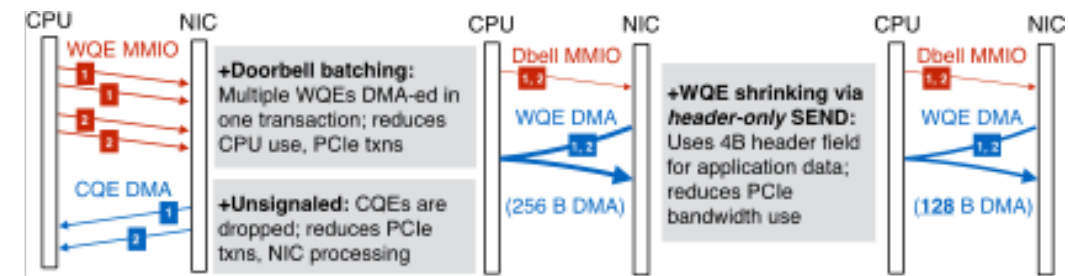
Comprehensive work on “best practises” and guidelines for performance optimization



Analyzes on using RDMA with NVM an in NUMA systems



Management model for scheduling RDMA transfers



[Kalia et al., 2016]

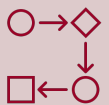
# Performance metrics and guidelines for RDMA



**Comprehensive work on “best practises” and guidelines for performance optimization**



Analyzes on using RDMA with NVM an in NUMA systems



Management model for scheduling RDMA transfers

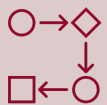
# Performance metrics and guidelines for RDMA



Comprehensive work on “best practises” and guidelines for performance optimization



**Analyzes on using RDMA with NVM an in NUMA systems**



Management model for scheduling RDMA transfers

[MacArthur and Russel ‘12, Nelson and Palmieri ‘19]



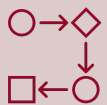
# Performance metrics and guidelines for RDMA



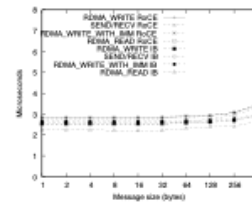
Comprehensive work on “best practises” and guidelines for performance optimization



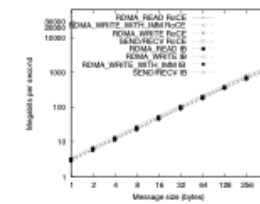
Analyzes on using RDMA with NVM an in NUMA systems



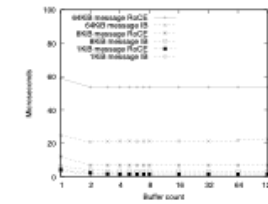
Management model for scheduling RDMA transfers



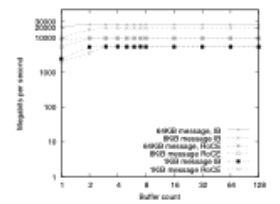
(a) Average one-way time with each opcode set for small messages using one buffer.



(b) Average throughput with each opcode set for small messages using one buffer.



(c) Average one-way time with RDMA WRITE for large messages using multiple buffers with multiple work requests per posting.



(d) Average throughput with RDMA WRITE for large messages using multiple buffers with multiple work requests per posting.

[MacArthur and Russel ‘12, Nelson and Palmieri ‘19]

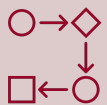
# Performance metrics and guidelines for RDMA



Comprehensive work on “best practises” and guidelines for performance optimization



**Analyzes on using RDMA with NVM an in NUMA systems**



Management model for scheduling RDMA transfers

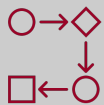
# Performance metrics and guidelines for RDMA



Comprehensive work on “best practises” and guidelines for performance optimization



Analyzes on using RDMA with NVM an in NUMA systems



**Management model for scheduling RDMA transfers**

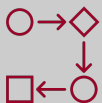
# Performance metrics and guidelines for RDMA



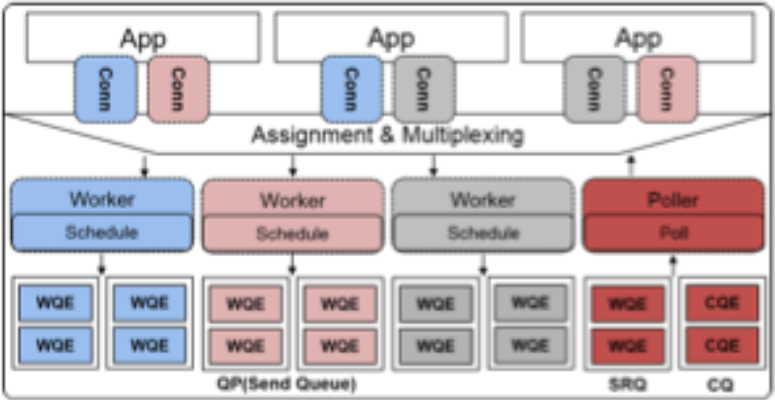
Comprehensive work on “best practises” and guidelines for performance optimization



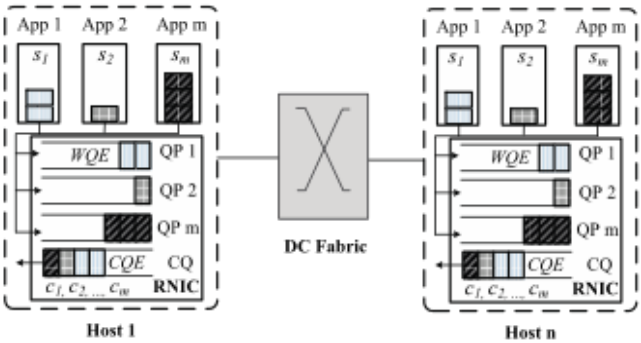
Analyzes on using RDMA with NVM an in NUMA systems



Management model for scheduling RDMA transfers



[Qui et al. '18]



[Shen 'et al. '20]

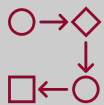
# Performance metrics and guidelines for RDMA



Comprehensive work on “best practises” and guidelines for performance optimization



Analyzes on using RDMA with NVM an in NUMA systems



**Management model for scheduling RDMA transfers**

# Performance metrics and Offloading simulations for NMC

# Performance metrics and Offloading simulations for NMC

[1]

Hsieh, K., Ebrahim, E., Kim, G., Chatterjee, N., O'Connor, M., Vijaykumar, N., Mutlu, O. and Keckler, S.W. 2016. Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems. *ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)* (Jun. 2016), 204–216.



**Lots of work on determining offload-ability of instructions**

[2]

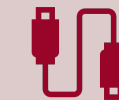
Khan, K., Pasricha, S. and Kim, R.G. 2020. A Survey of Resource Management for Processing-In-Memory and Near-Memory Processing Architectures. *Journal of Low Power Electronics and Applications*. 10, 4 (Dec. 2020), 30. DOI:<https://doi.org/10.3390/jlpea10040030>.



Frameworks for estimating performance and energy usage for specific applications

[3]

Mutlu, O., Ghose, S., Gómez-Luna, J. and Ausavarungnirun, R. 2020. A Modern Primer on Processing in Memory. *CoRR*. abs/2012.03112, (2020).



Performance metrics for NMC



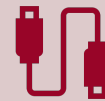
# Performance metrics and Offloading simulations for NMC



**Lots of work on determining offload-ability of instructions**

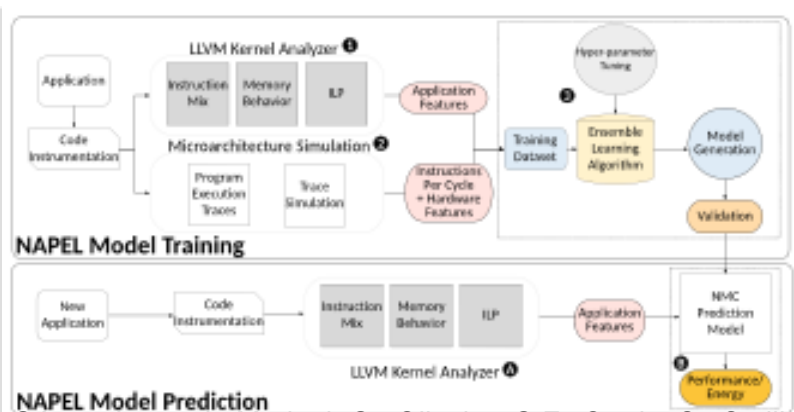


Frameworks for estimating performance and energy usage for specific applications

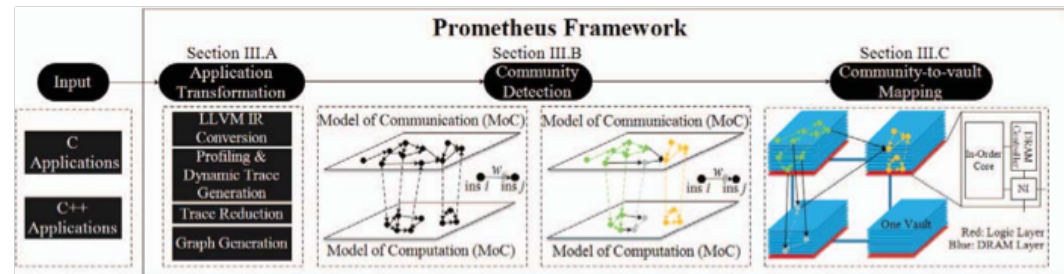


Performance metrics for NMC

# Performance metrics and Offloading simulations for NMC



Singh, G., Gómez-Luna, J., Mariani, G., Oliveira, G.F., Corda, S., Stuijk, S., Mutlu, O. and Corporaal, H. NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning. *DAC'2019*.



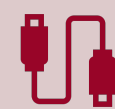
Xiao, Y., Nazarian, S. and Bogdan, P. 2018. Prometheus: Processing-in-memory heterogeneous architecture design from a multi-layer network theoretic strategy. *DATE'2018*.



Lots of work on determining offload-ability of instructions



Frameworks for estimating performance and energy usage for specific applications



Performance metrics for NMC

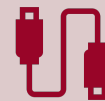
# Performance metrics and Offloading simulations for NMC



Lots of work on determining offload-ability of instructions



**Frameworks for estimating performance and energy usage for specific applications**



Performance metrics for NMC

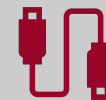
# Performance metrics and Offloading simulations for NMC



Lots of work on determining offload-ability of instructions

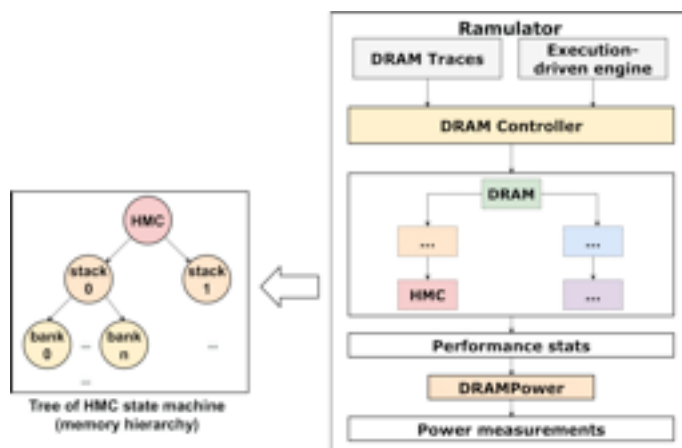


Frameworks for estimating performance and energy usage for specific applications



**Performance metrics for NMC**

# Performance metrics and Offloading simulations for NMC



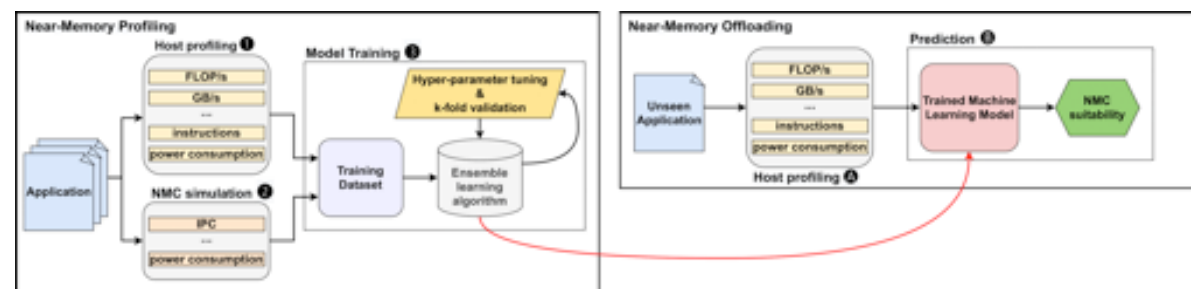
Lots of work on determining offload-ability of instructions



Frameworks for estimating performance and energy usage for specific applications



**Performance metrics for NMC**



Corda, S., Kumaraswamy, M., Awan, A.J., Jordans, R., Kumar, A. and Corporaal, H. NMPO: Near-Memory Computing Profiling and Offloading. *Euromicro DSD/SEAA 2021*.

# Models for NVRAM

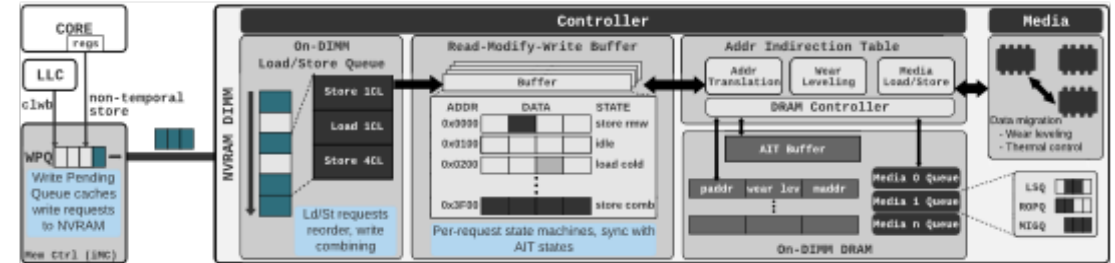
# Models for NVRAM

- Simulator for NVRAM architectures<sup>1)</sup>



# Models for NVRAM

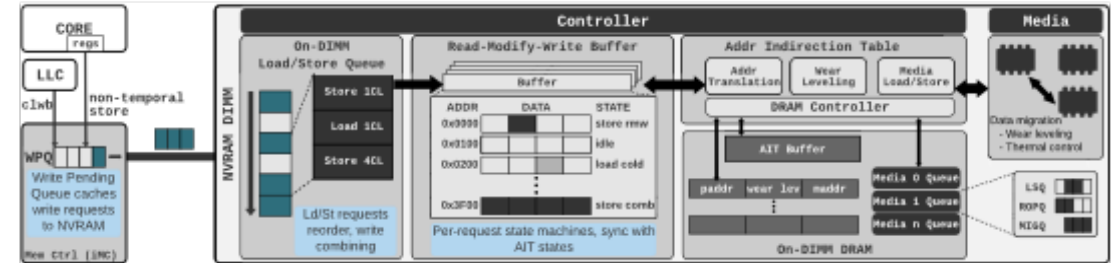
- Simulator for NVRAM architectures<sup>1)</sup>



1) Wang, Z. et al., Characterizing and Modelling Non-Volatile Memory Systems. *MICRO*'2020.

# Models for NVRAM

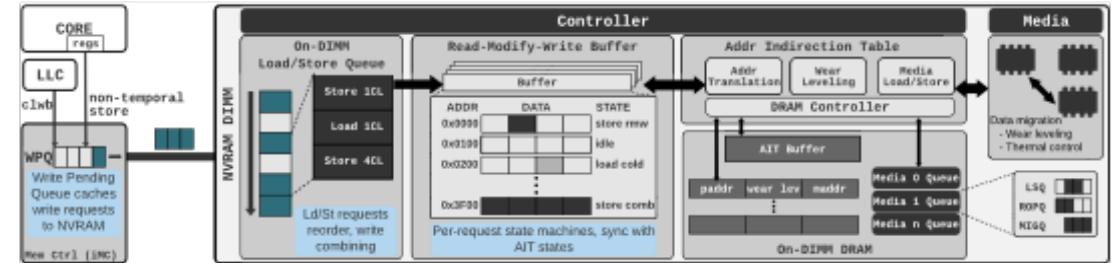
- Simulator for NVRAM architectures<sup>1)</sup>
- Application-specific cost model for data stream processing<sup>2)</sup>



1) Wang, Z. et al., Characterizing and Modelling Non-Volatile Memory Systems. *MICRO*'2020.

# Models for NVRAM

- Simulator for NVRAM architectures<sup>1)</sup>
- Application-specific cost model for data stream processing<sup>2)</sup>



1) Wang, Z. et al., Characterizing and Modelling Non-Volatile Memory Systems. *MICRO*' 2020.

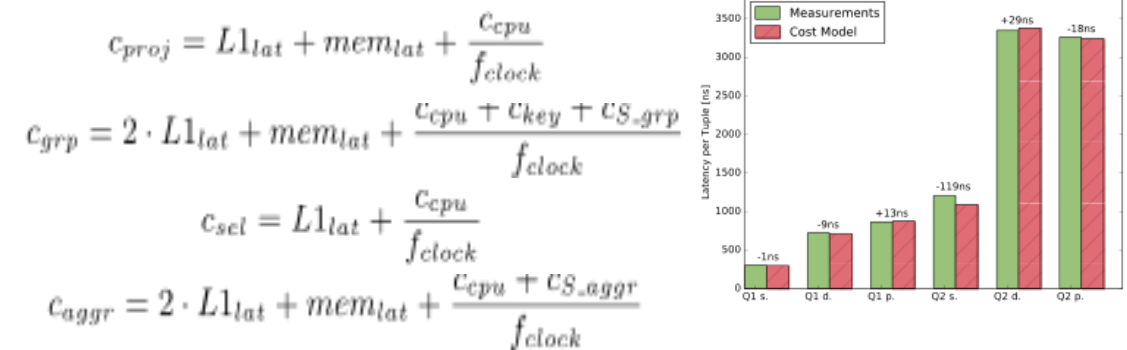
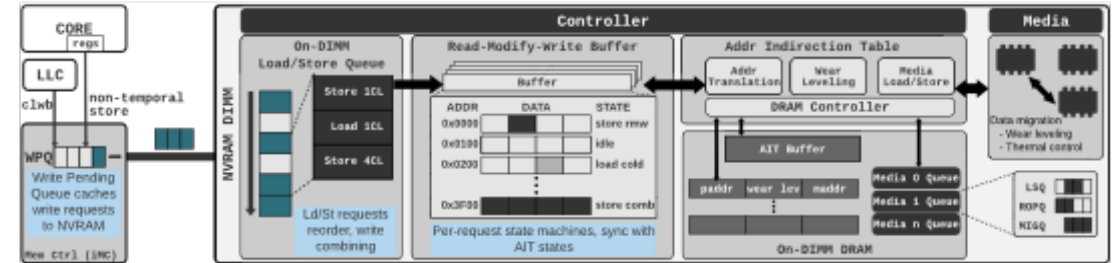


Figure 4: Query Results

2) Pohl, C. and Sattler, K.-U. A Cost Model for Data Stream Processing on Modern Hardware. *ADMS*' 2017.

# Models for NVRAM

- Simulator for NVRAM architectures<sup>1)</sup>
- Application-specific cost model for data stream processing<sup>2)</sup>
- Performance studies [Izraelevitz et al. '19]



1) Wang, Z. et al., Characterizing and Modelling Non-Volatile Memory Systems. *MICRO*'2020.

$$c_{proj} = L1_{lat} + mem_{lat} + \frac{c_{cpu}}{f_{clock}}$$

$$c_{grp} = 2 \cdot L1_{lat} + mem_{lat} + \frac{c_{cpu} + c_{key} + c_{s.grp}}{f_{clock}}$$

$$c_{sel} = L1_{lat} + \frac{c_{cpu}}{f_{clock}}$$

$$c_{aggr} = 2 \cdot L1_{lat} + mem_{lat} + \frac{c_{cpu} + c_{s.aggr}}{f_{clock}}$$

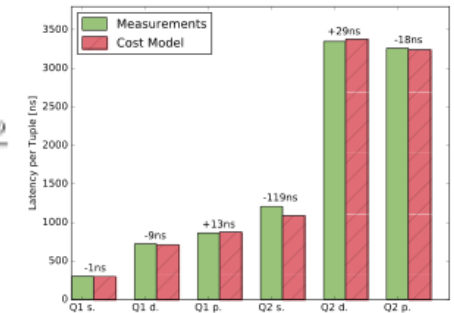
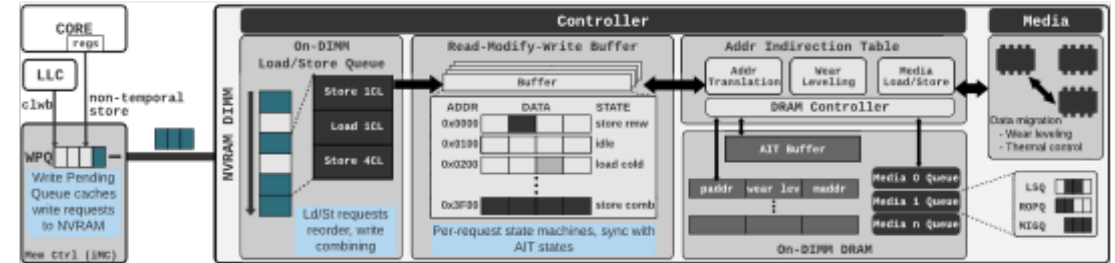


Figure 4: Query Results

2) Pohl, C. and Sattler, K.-U. A Cost Model for Data Stream Processing on Modern Hardware. *ADMS*'2017.

# Models for NVRAM

- Simulator for NVRAM architectures<sup>1)</sup>
- Application-specific cost model for data stream processing<sup>2)</sup>
- Performance studies [Izraelevitz et al. '19]
- Programming models [Scargall '20, George '20, Köppen '19]



1) Wang, Z. et al., Characterizing and Modelling Non-Volatile Memory Systems. *MICRO* '2020.

$$c_{proj} = L1_{lat} + mem_{lat} + \frac{c_{cpu}}{f_{clock}}$$

$$c_{grp} = 2 \cdot L1_{lat} + mem_{lat} + \frac{c_{cpu} + c_{key} + c_{s.grp}}{f_{clock}}$$

$$c_{sel} = L1_{lat} + \frac{c_{cpu}}{f_{clock}}$$

$$c_{aggr} = 2 \cdot L1_{lat} + mem_{lat} + \frac{c_{cpu} + c_{s.aggr}}{f_{clock}}$$

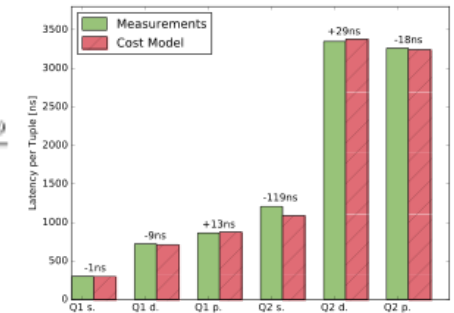


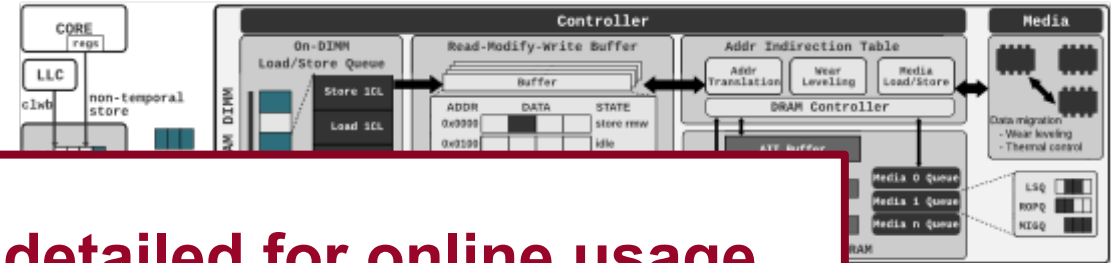
Figure 4: Query Results

2) Pohl, C. and Sattler, K.-U. A Cost Model for Data Stream Processing on Modern Hardware. *ADMS* '2017.

# Models for NVRAM

- Simulator for NVRAM architecture
- Application for data
- Performance et al. '19
- Programming models [Scargall '20, George '20, Köppen '19]

Existing models too detailed for online usage  
Models without holistic view  
Isolated performance analyzes (best case, single application)



Memory

$$c_{sel} = L1_{lat} + \frac{c_{cpu}}{f_{clock}}$$

$$c_{aggr} = 2 \cdot L1_{lat} + mem_{lat} + \frac{c_{cpu} + c_{S\_aggr}}{f_{clock}}$$

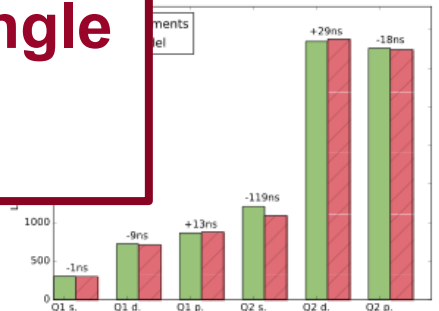


Figure 4: Query Results  
2) Pohl, C. and Sattler, K.-U. A Cost Model for Data Stream Processing on Modern Hardware. ADMS' 2017.

# Managing strategies and measurements for HBM



# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das  
Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal  
Washington University in St. Louis  
kunal@wustl.edu

Michael A. Bender  
Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry  
Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley  
Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips  
Sandia National Laboratories  
caphill@sandia.gov

# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das

Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal

Washington University in St. Louis  
kunal@wustl.edu

Michael A. Bender

Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry

Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley

Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips

Sandia National Laboratories  
caphill@sandia.gov

- HBM as additional cache layer

# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das

Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal

Washington University in St. Louis  
kunal@wustl.edu

Michael A. Bender

Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry

Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley

Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips

Sandia National Laboratories  
caphill@sandia.gov

- HBM as additional cache layer
- Replacement strategy for pages in HBM

# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das

Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal

Washington University in St. Louis  
kunal@wustl.edu

Michael A. Bender

Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry

Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley

Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips

Sandia National Laboratories  
caphill@sandia.gov

- HBM as additional cache layer
- Replacement strategy for pages in HBM
- Priority-based strategy

# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das

Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal

Washington University in St. Louis  
kunal@wustl.edu

Michael A. Bender

Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry

Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley

Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips

Sandia National Laboratories  
caphill@sandia.gov

- HBM as additional cache layer
- Replacement strategy for pages in HBM
- Priority-based strategy
- Performance metrics for HBM used

# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das  
Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal  
Washington University in St. Louis  
kunal@wustl.edu

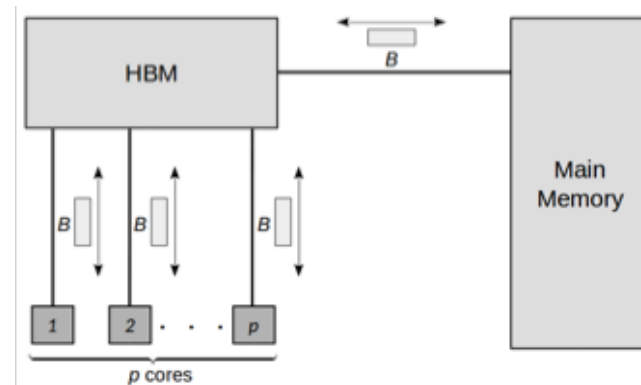
Michael A. Bender  
Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry  
Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley  
Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips  
Sandia National Laboratories  
caphill@sandia.gov

- HBM as additional cache layer
- Replacement strategy for pages in HBM
- Priority-based strategy
- Performance metrics for HBM used
- Model only for HBM



# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das  
Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal  
Washington University in St. Louis  
kunal@wustl.edu

Michael A. Bender  
Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry  
Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley  
Carnegie Mellon University  
moseleyb@andrew.cmu.edu

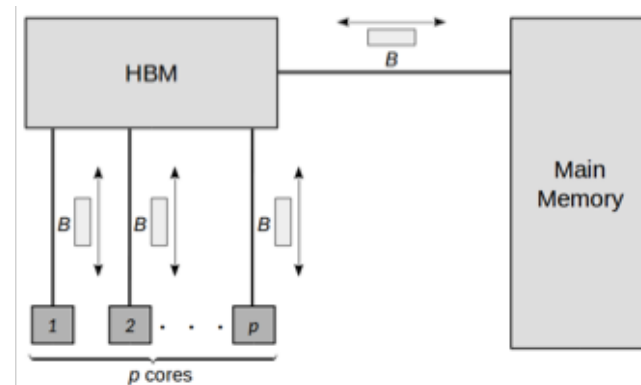
Cynthia A. Phillips  
Sandia National Laboratories  
caphill@sandia.gov

## Object Placement for High Bandwidth Memory Augmented with High Capacity Memory

Mohammad Laghari

Didem Unat

- HBM as additional cache layer
- Replacement strategy for pages in HBM
- Priority-based strategy
- Performance metrics for HBM used
- Model only for HBM



# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das  
Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal  
Washington University in St. Louis  
kunal@wustl.edu

Michael A. Bender  
Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry  
Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley  
Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips  
Sandia National Laboratories  
caphill@sandia.gov

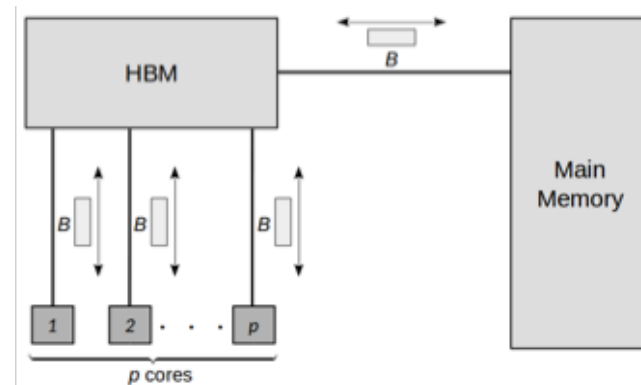
## Object Placement for High Bandwidth Memory Augmented with High Capacity Memory

Mohammad Laghari

Didem Unat

- Automatic object placement in hybrid DRAM/HBM systems

- HBM as additional cache layer
- Replacement strategy for pages in HBM
- Priority-based strategy
- Performance metrics for HBM used
- Model only for HBM





# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das  
Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal  
Washington University in St. Louis  
kunal@wustl.edu

Michael A. Bender  
Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry  
Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley  
Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips  
Sandia National Laboratories  
caphill@sandia.gov

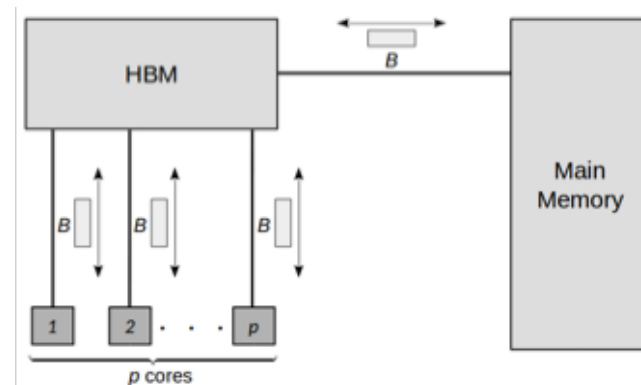
- HBM as additional cache layer
- Replacement strategy for pages in HBM
- Priority-based strategy
- Performance metrics for HBM used
- Model only for HBM

## Object Placement for High Bandwidth Memory Augmented with High Capacity Memory

Mohammad Laghari

Didem Unat

- Automatic object placement in hybrid DRAM/HBM systems
- Heuristic based on access frequency for objects



# Managing strategies and measurements for HBM

## How to Manage High-Bandwidth Memory Automatically

Rathish Das  
Stony Brook University  
radas@cs.stonybrook.edu

Kunal Agrawal  
Washington University in St. Louis  
kunal@wustl.edu

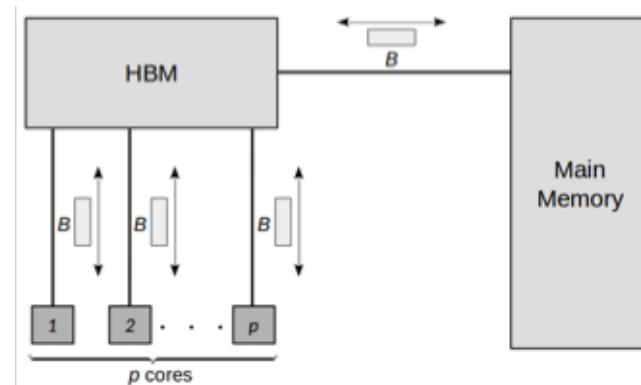
Michael A. Bender  
Stony Brook University  
bender@cs.stonybrook.edu

Jonathan Berry  
Sandia National Laboratories  
jberry@sandia.gov

Benjamin Moseley  
Carnegie Mellon University  
moseleyb@andrew.cmu.edu

Cynthia A. Phillips  
Sandia National Laboratories  
caphill@sandia.gov

- HBM as additional cache layer
- Replacement strategy for pages in HBM
- Priority-based strategy
- Performance metrics for HBM used
- Model only for HBM



## Object Placement for High Bandwidth Memory Augmented with High Capacity Memory

Mohammad Laghari

Didem Unat

- Automatic object placement in hybrid DRAM/HBM systems
- Heuristic based on access frequency for objects
- No performance model

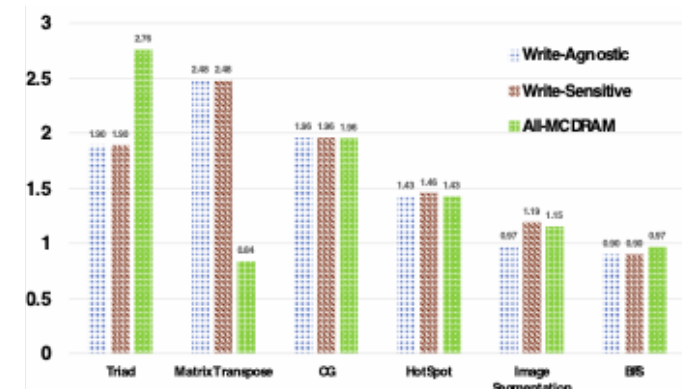
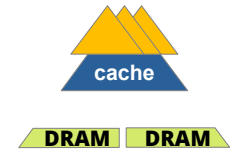


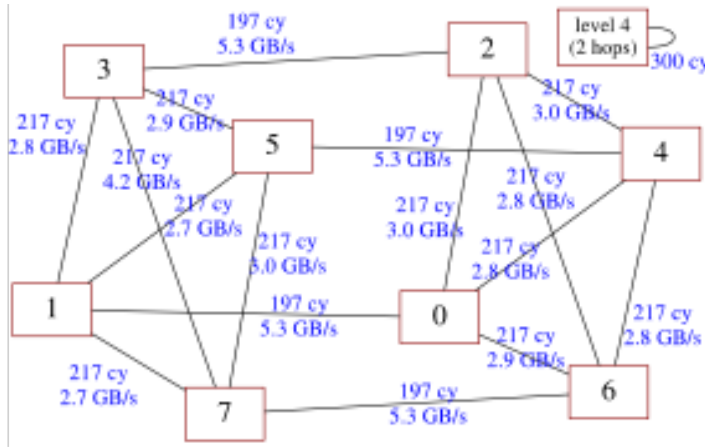
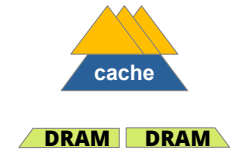
Fig. 5: Speedup of our placement configuration achieved over placing all objects in the slow memory

# System models: MCTOP



Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.

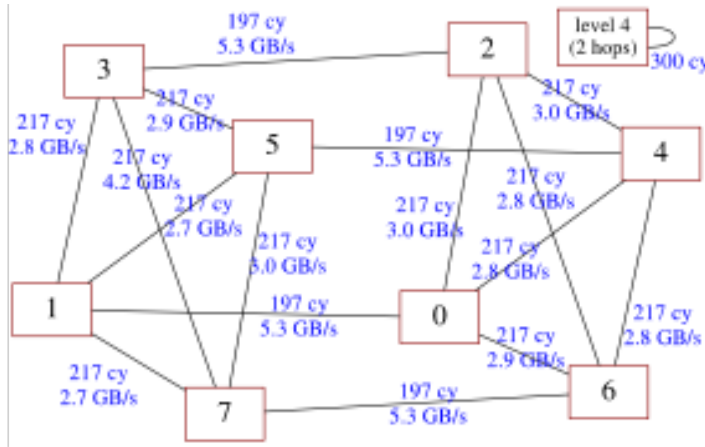
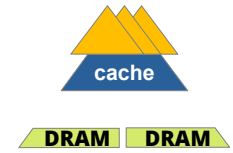
# System models: MCTOP



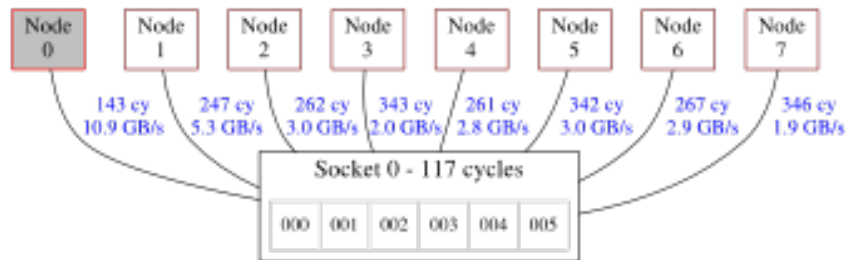
Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.



# System models: MCTOP

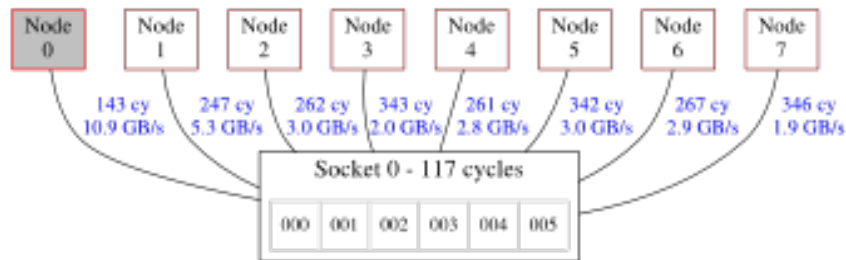
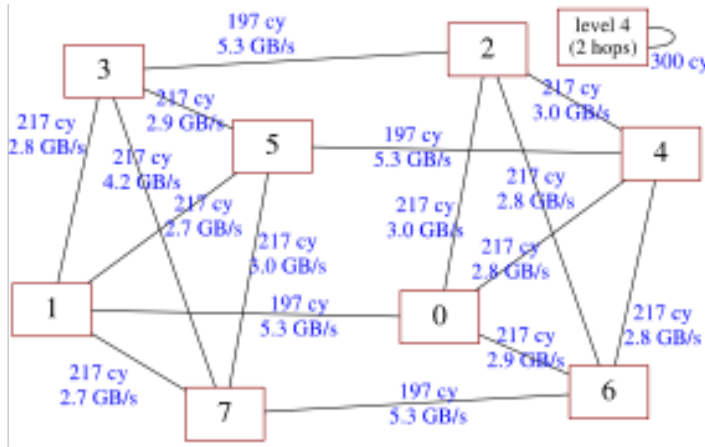
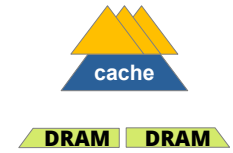


Includes communication costs



Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.

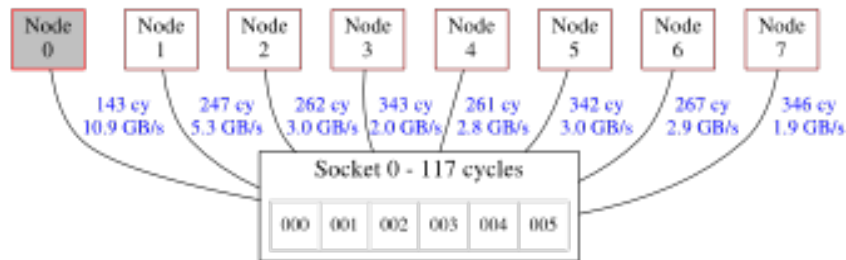
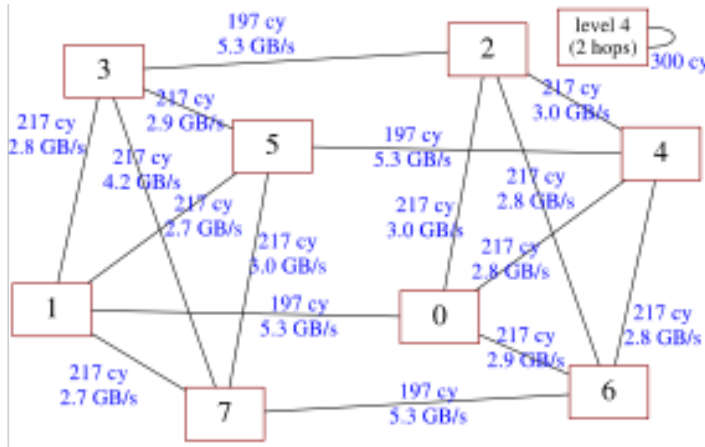
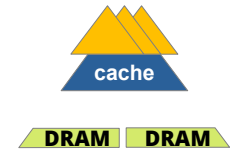
# System models: MCTOP



- Includes communication costs
- Performance predictions

Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.

# System models: MCTOP

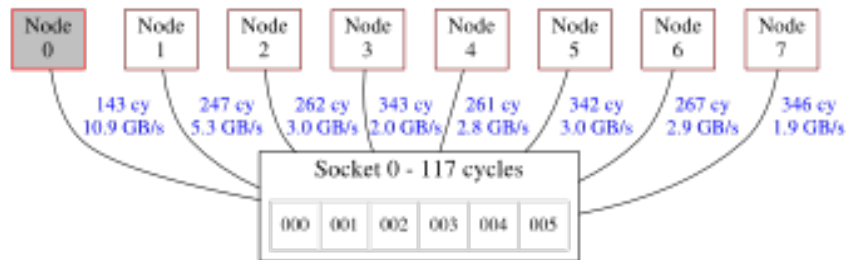
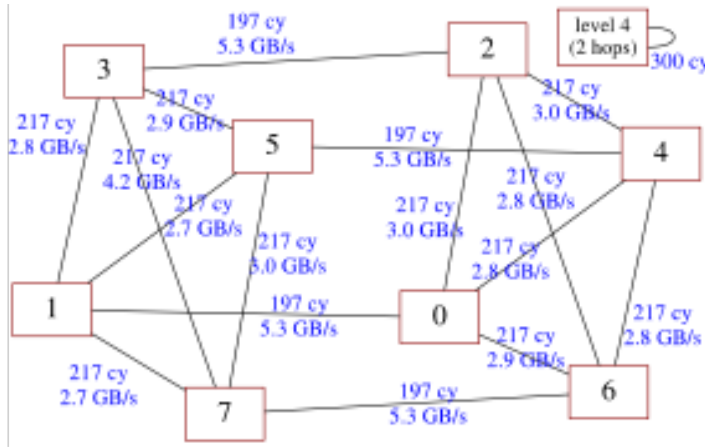
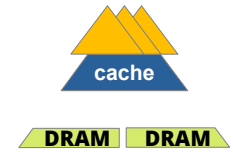






- Includes communication costs
- Performance predictions
- Optimized mapping decisions

Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.



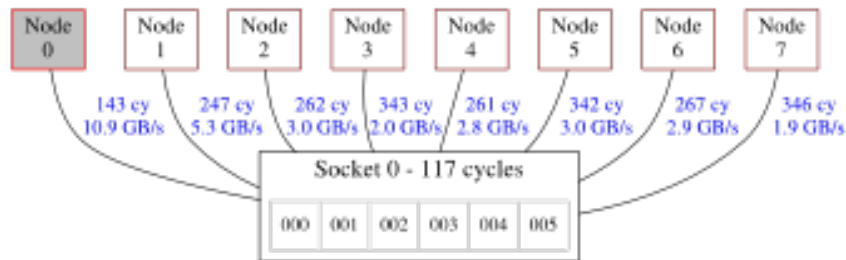
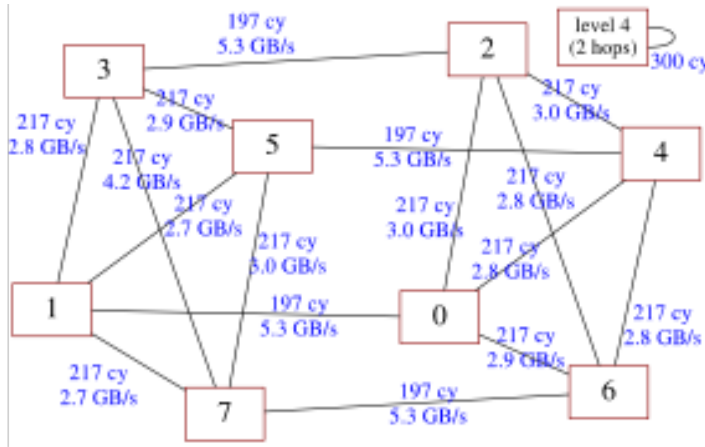
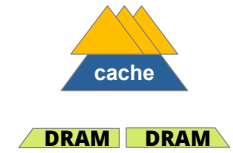
# System models: MCTOP








-  Includes communication costs
-  Performance predictions
-  Optimized mapping decisions
-  User-space library

Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.

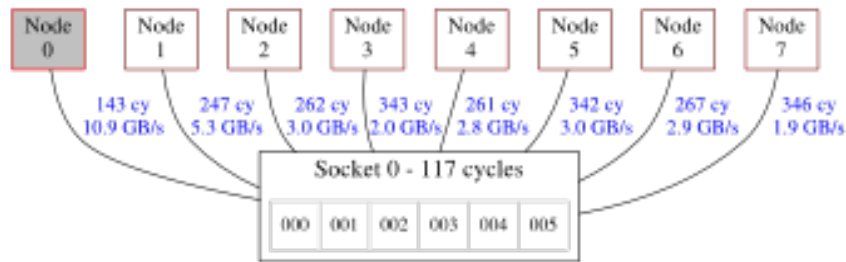
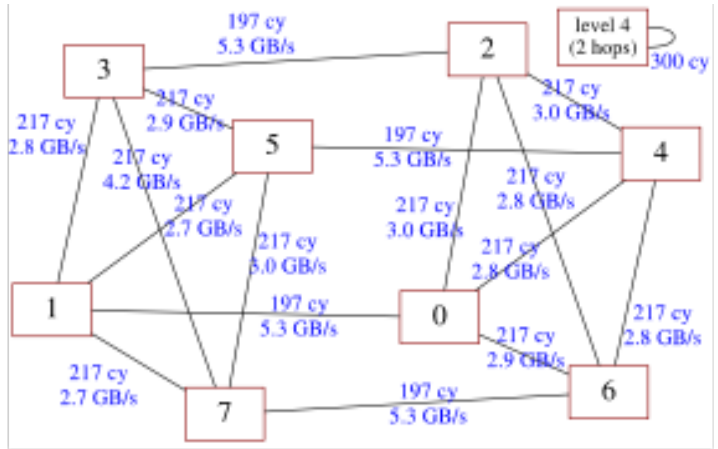
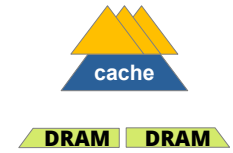
# System models: MCTOP



-  Includes communication costs
-  Performance predictions
-  Optimized mapping decisions
-  User-space library
-  No updates at runtime

Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.

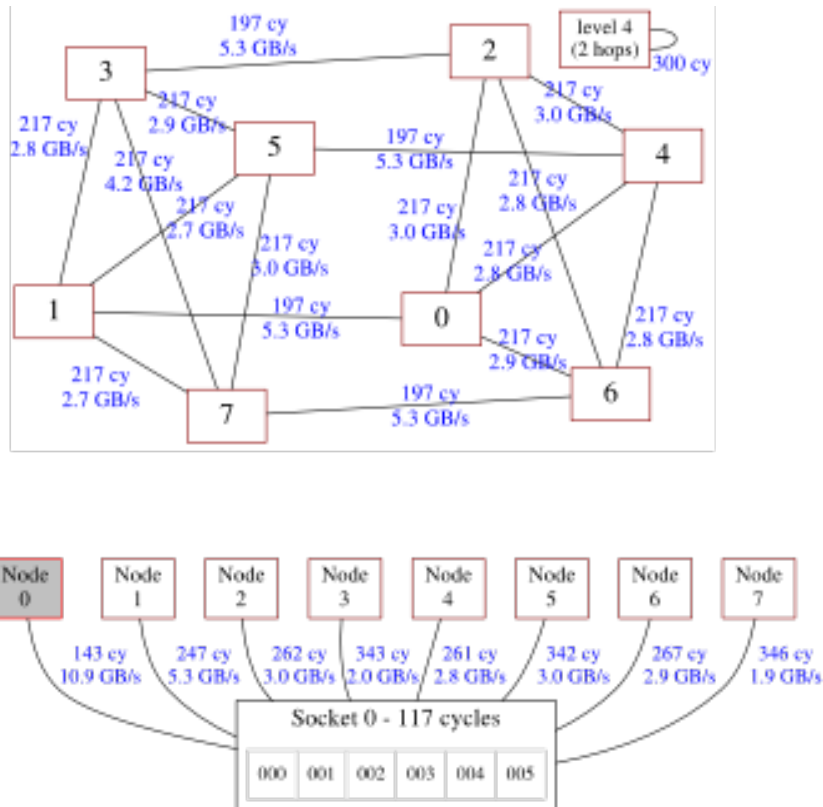
# System models: MCTOP



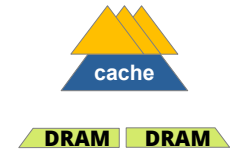
- Includes communication costs
- Performance predictions
- Optimized mapping decisions
- User-space library
- No updates at runtime
- No support for DMT

Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.

## System models: MCTOP

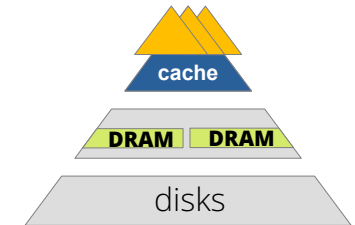


Chatzopoulos, G., Guerraoui, R., Harris, T. and Trigonakis, V. 2017. Abstracting Multi-Core Topologies with MCTOP. *Proceedings of the Twelfth European Conference on Computer Systems* (Belgrade Serbia, Apr. 2017), 544–559.



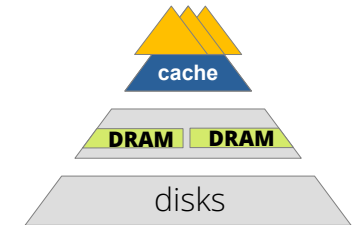
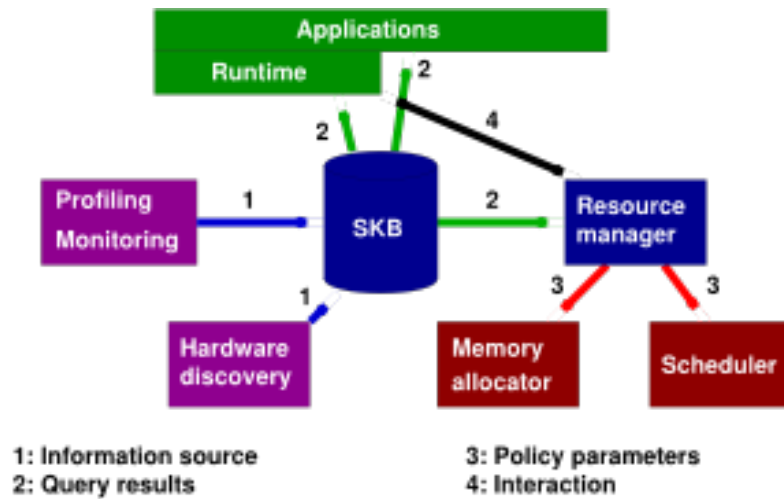
- Includes communication costs
- Performance predictions
- Optimized mapping decisions
- User-space library
- No updates at runtime
- No support for DMT
- Only application view
- No application model

# System models: System Knowledge Base



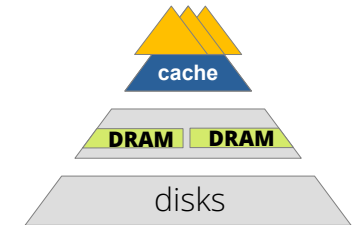
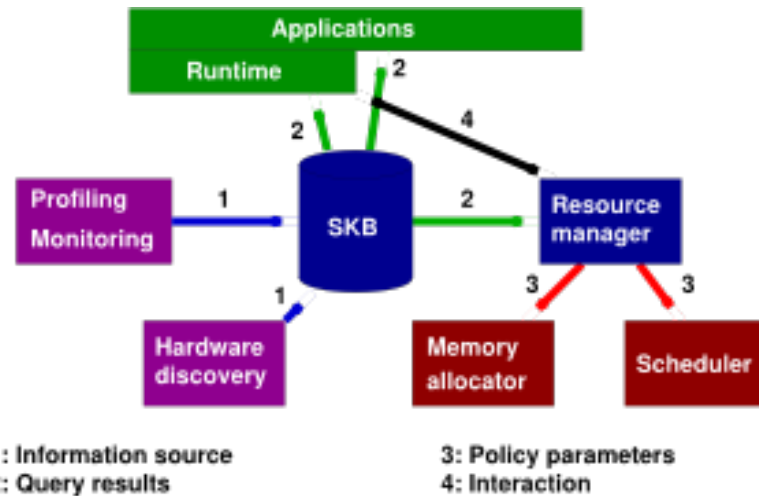
Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.

# System models: System Knowledge Base



Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.

# System models: System Knowledge Base

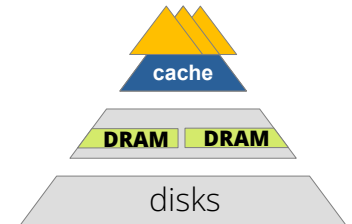
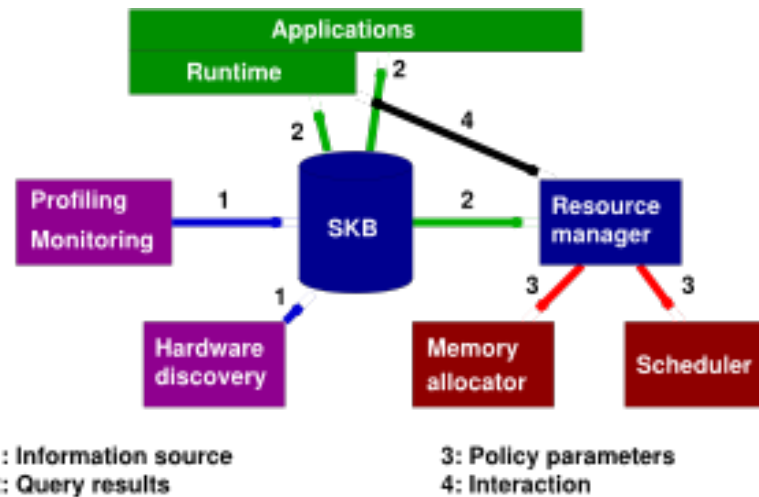


```

apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARNr, Base, Size, mem|io,
      (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
    
```

Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.

# System models: System Knowledge Base



 Includes communication costs

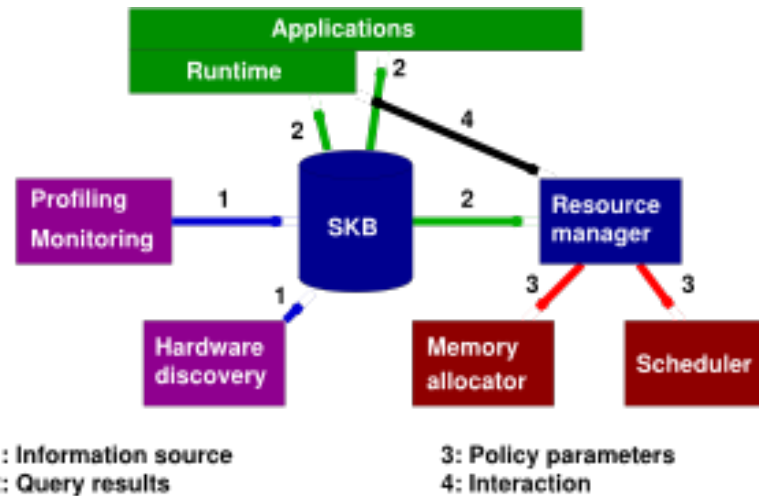
```

apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARNr, Base, Size, mem|io,
    (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
    
```

Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.

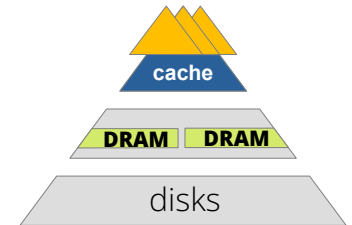




# System models: System Knowledge Base



```

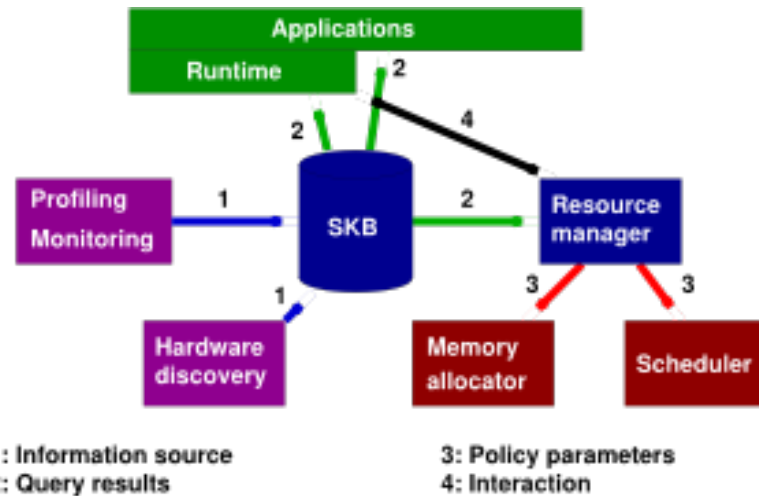
apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARnr, Base, Size, mem|io,
    (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
    
```



-  Includes communication costs
-  Performance predictions

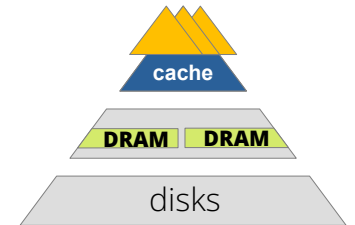
Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.




# System models: System Knowledge Base



```

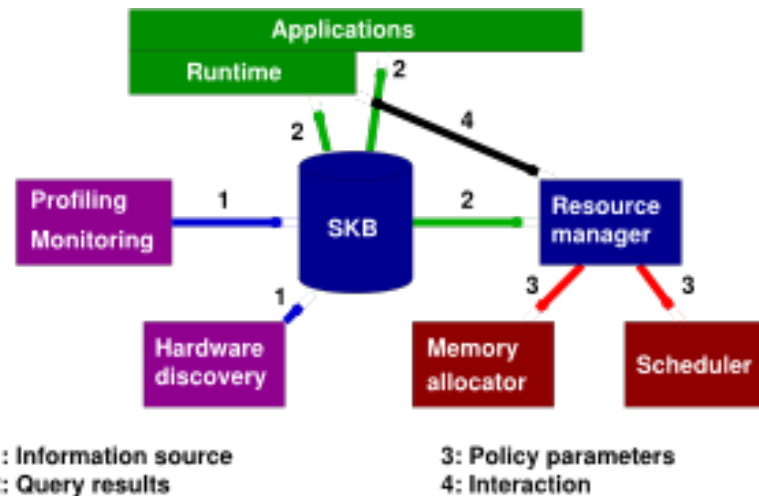
apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARnr, Base, Size, mem|io,
     (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
    
```



-  Includes communication costs
-  Performance predictions
-  Optimized mapping decisions

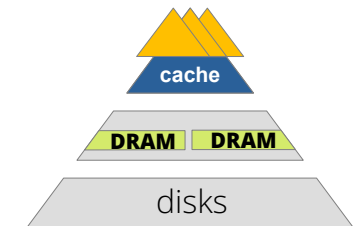
Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.





# System models: System Knowledge Base



```

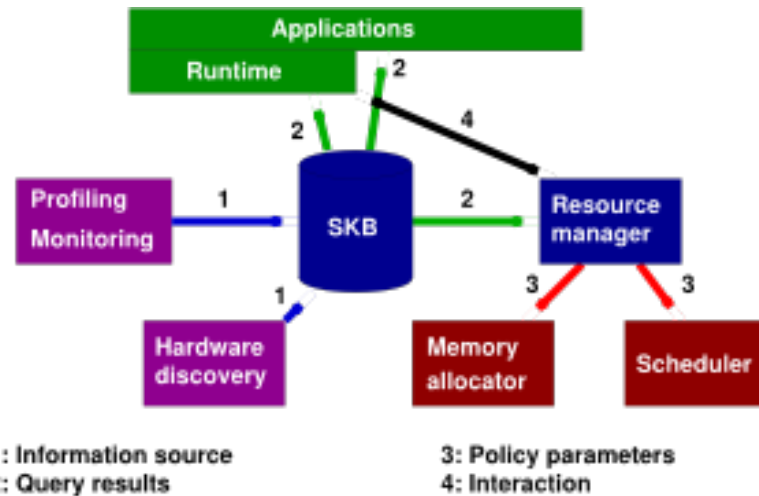
apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARnr, Base, Size, mem|io,
     (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
    
```



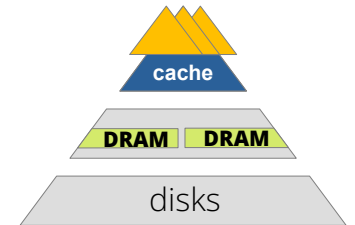
-  Includes communication costs
-  Performance predictions
-  Optimized mapping decisions
-  Updates at runtime






Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.

# System models: System Knowledge Base



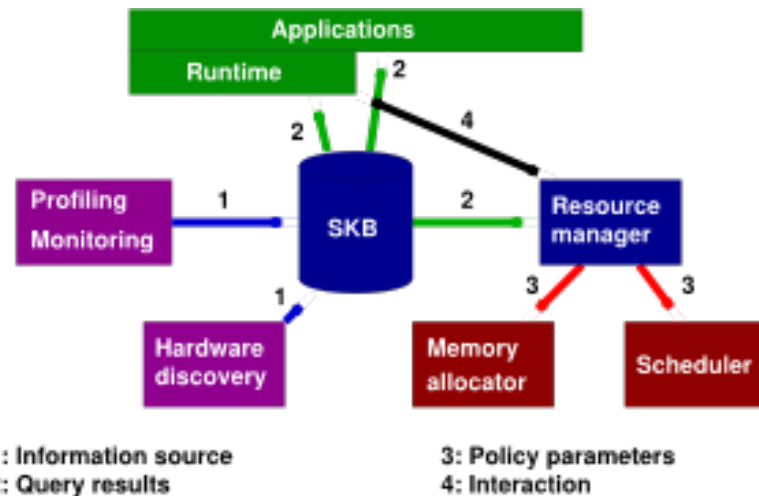
```
apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARnr, Base, Size, mem|io,
    (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
```



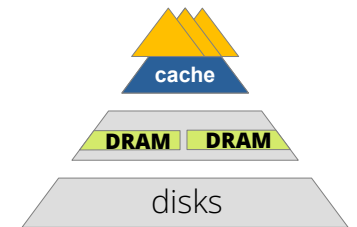
-  Includes communication costs
-  Performance predictions
-  Optimized mapping decisions
-  Updates at runtime
-  Whole system view







Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.

# System models: System Knowledge Base



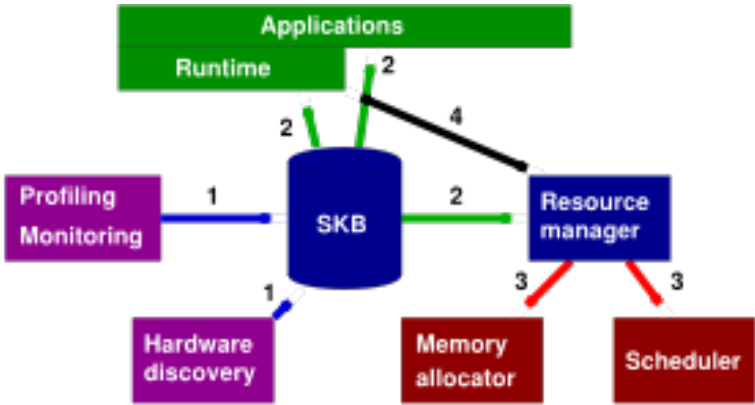
```
apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARnr, Base, Size, mem|io,
     (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
```



-  Includes communication costs
-  Performance predictions
-  Optimized mapping decisions
-  Updates at runtime
-  Whole system view
-  Application model

Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.

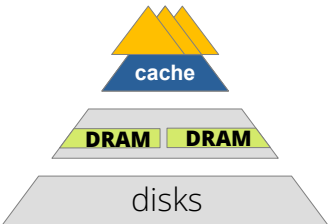
# System models: System Knowledge Base










1: Information source  
2: Query results  
3: Policy parameters  
4: Interaction

```

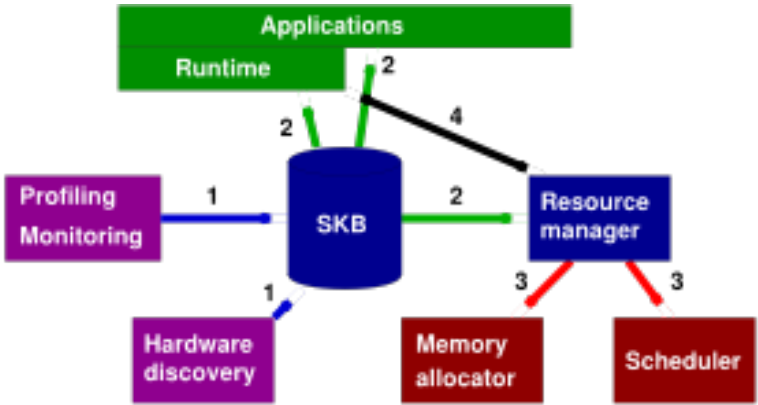
apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARnr, Base, Size, mem|io,
     (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
    
```



-  Includes communication costs
-  Performance predictions
-  Optimized mapping decisions
-  Updates at runtime
-  Whole system view
-  Application model
-  No support for DMT

Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.

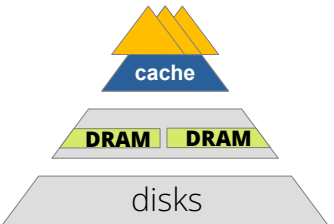
# System models: System Knowledge Base











1: Information source  
2: Query results  
3: Policy parameters  
4: Interaction

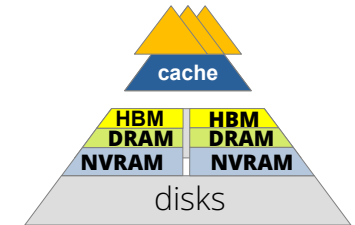
```
apic(ACPI_ProcessorID, APICID, Availability). % 1 = Yes, 0 = no
bridge(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, secondary(Sec)).
device(pcie|pci, addr(Bus, Dev, Fun), VendorID, DeviceID,
      Class, SubClass, ProgIf, IntPin).
interrupt_override(Bus, SourceIRQ, GlobalIRQ, IntiFlags).
rootbridge_address_window(addr(Bus, Dev, Fun), mem(Min, Max)).
bar(addr(Bus, Dev, Fun), BARnr, Base, Size, mem|io,
     (non)prefetchable, Bits (64|32)).
fixed_memory(Base, Limit).
apic_nmi(ACPI_ProcessorID, IntiFlags, Lint).
memory_region(Base, SzBits, SzBytes, RegionType, Data).
```

Schüpbach, A.L. 2012. *Tackling OS Complexity with Declarative Techniques*. ETH Zurich.



-  Includes communication costs
-  Performance predictions
-  Optimized mapping decisions
-  Updates at runtime
-  Whole system view
-  Application model
-  No support for DMT
-  Slow and resource heavy

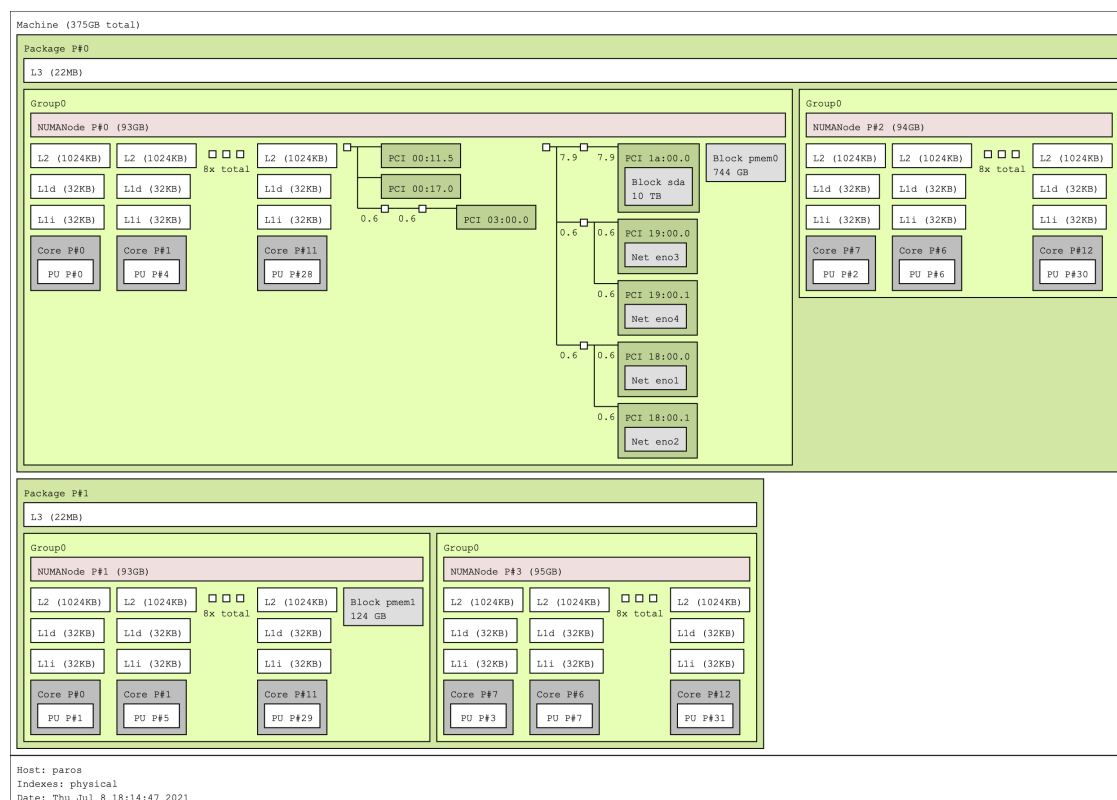
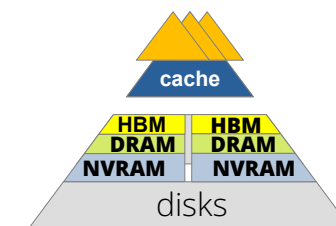
# System models: hwloc



Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.

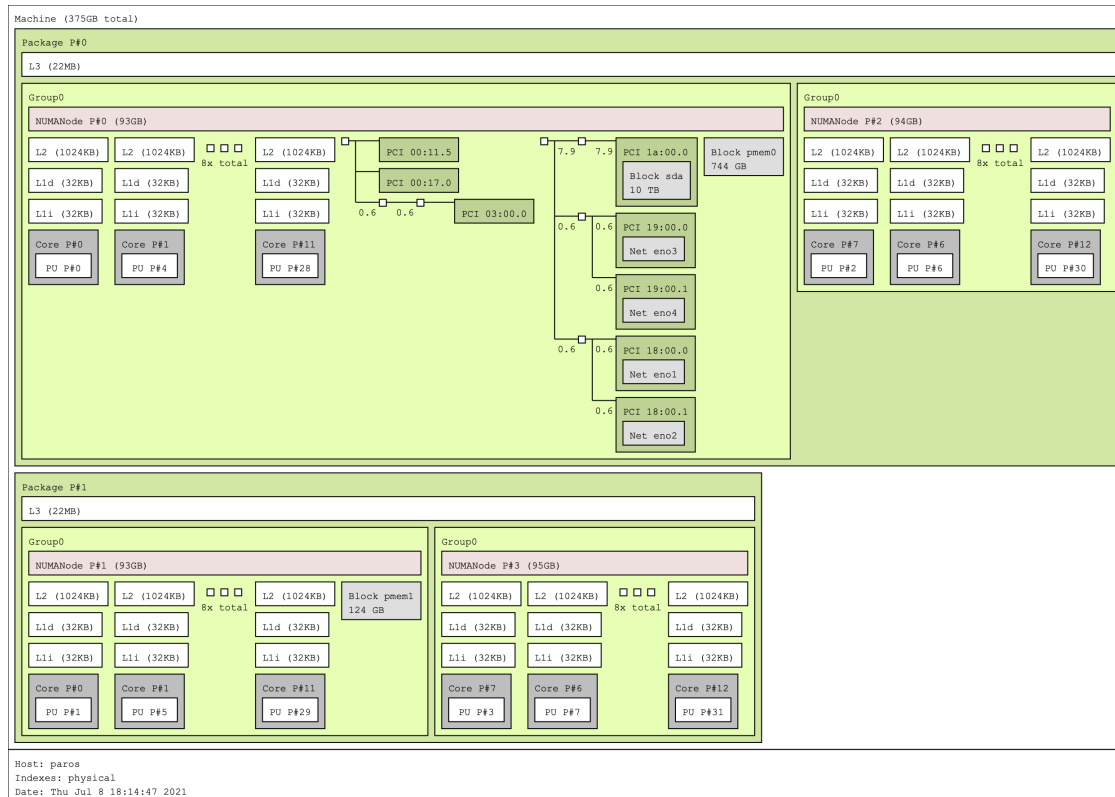


# System models: hwloc

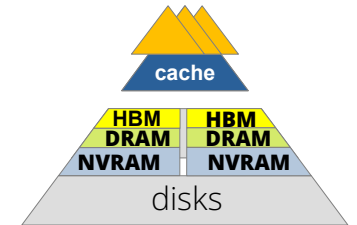


Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.

# System models: hwloc

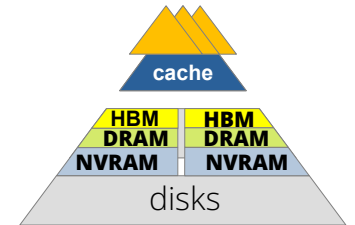
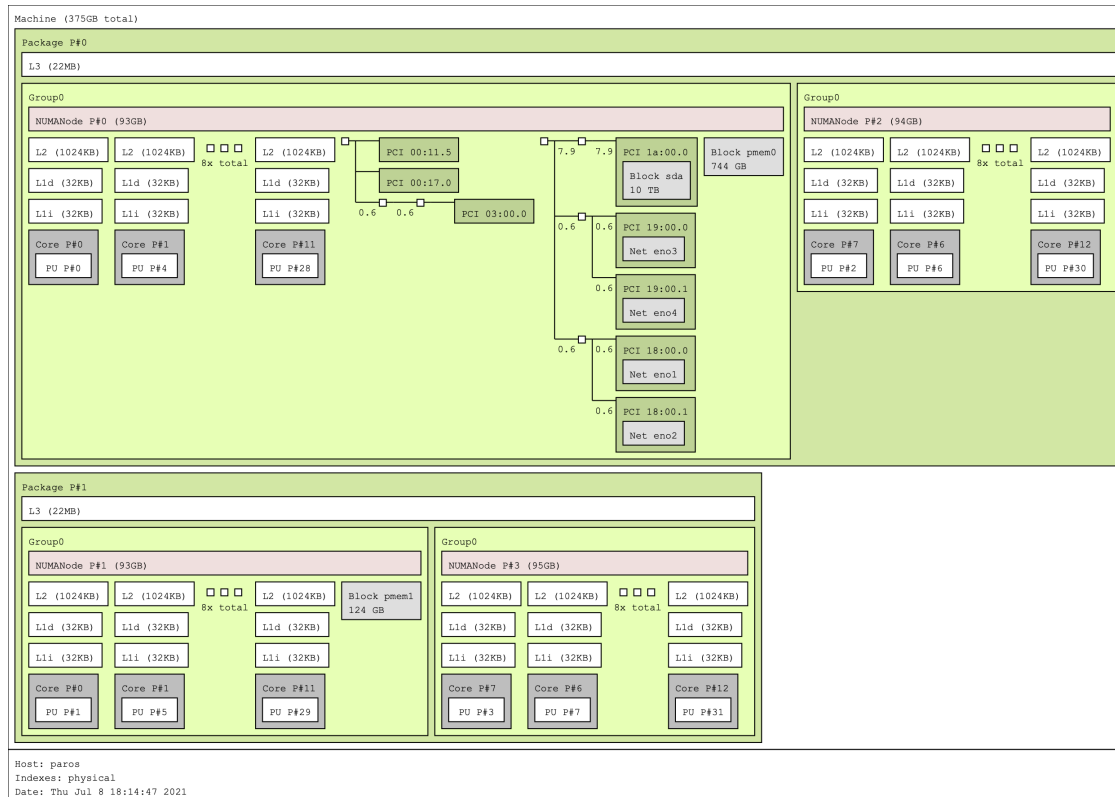


Support for DMT



Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.

# System models: hwloc



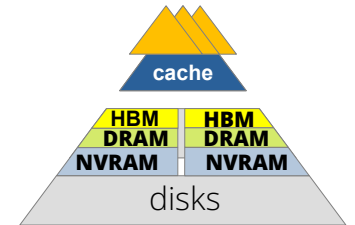
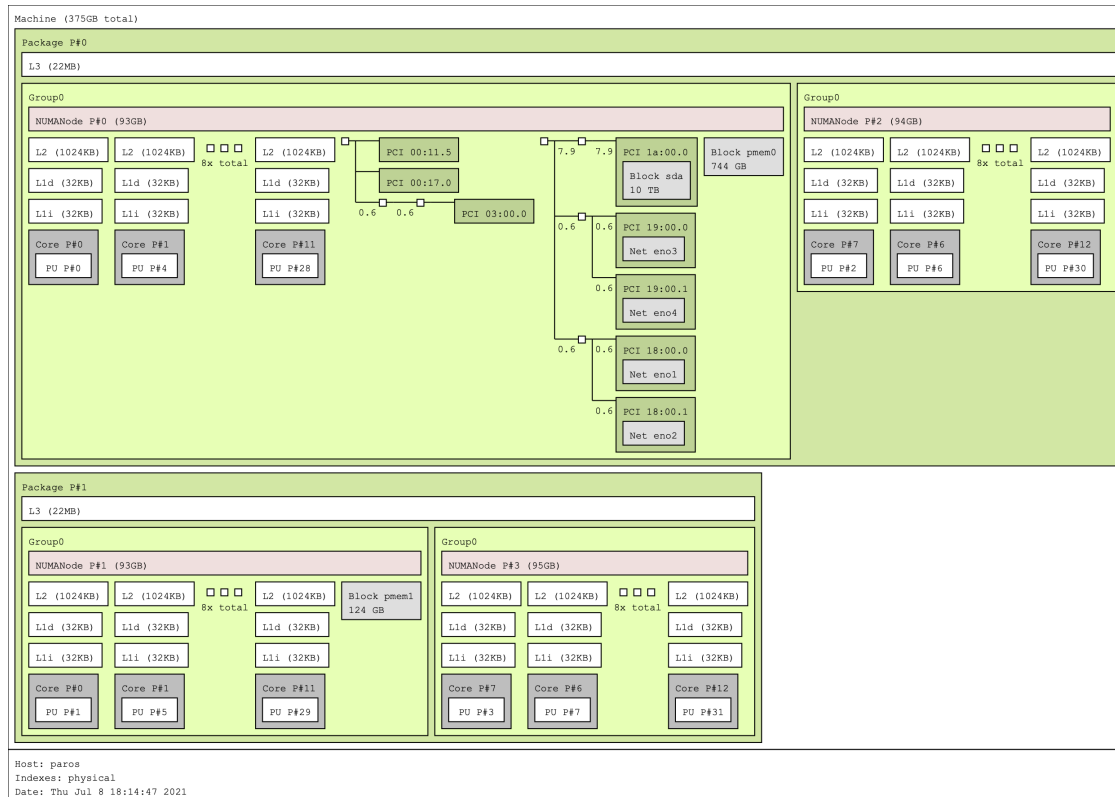
Support for DMT



User-space library

Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.

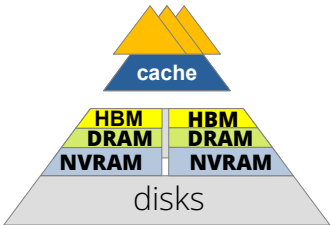
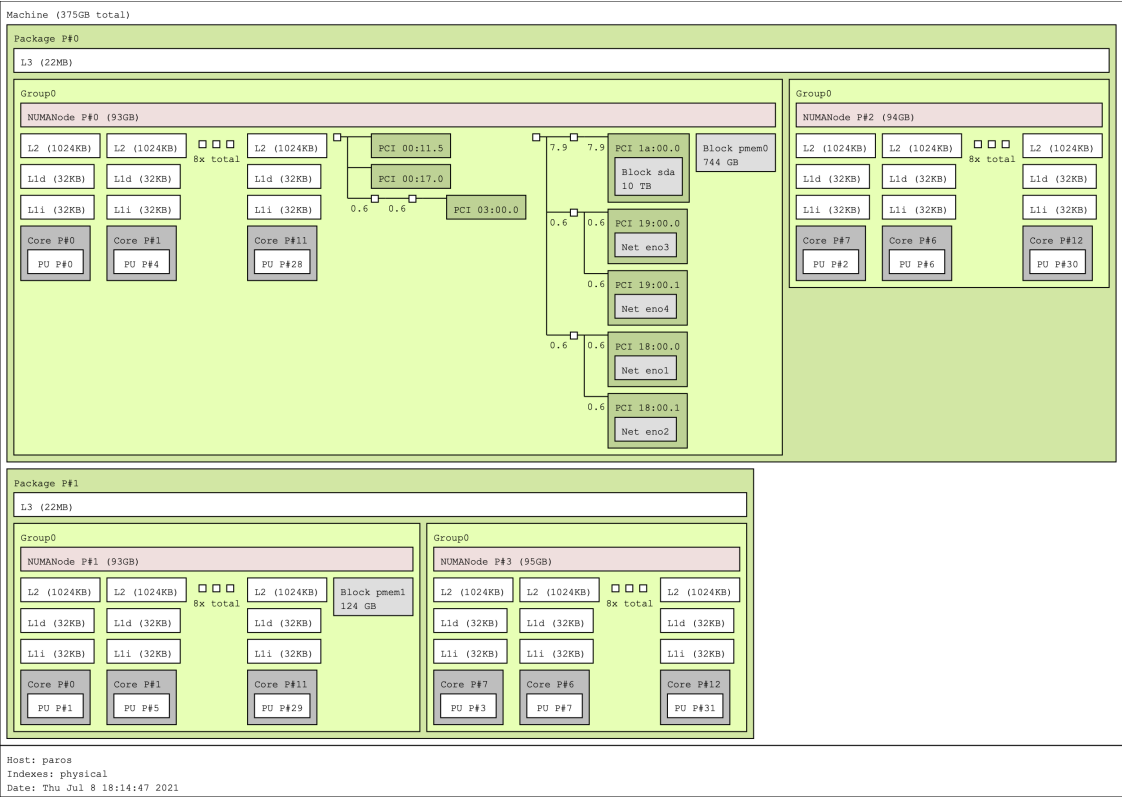
# System models: hwloc







- Support for DMT
- User-space library
- No Performance predictions

Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.

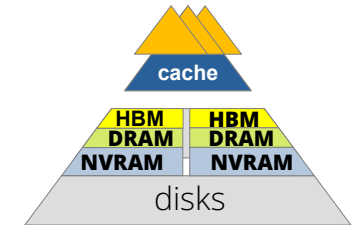
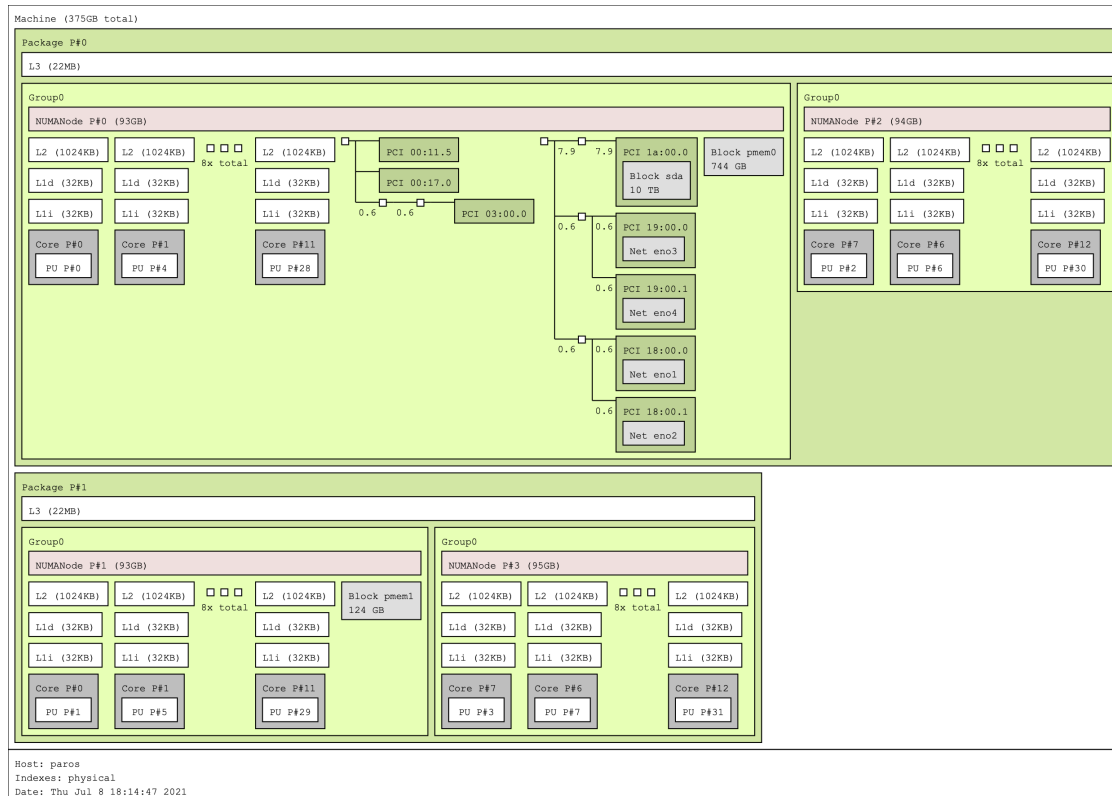
# System models: hwloc



-  Support for DMT
-  User-space library
-  No Performance predictions
-  No Optimized mapping decisions

Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.

## System models: hwloc



## Support for DMT

## User-space library

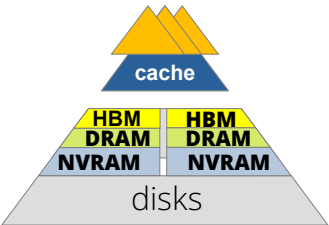
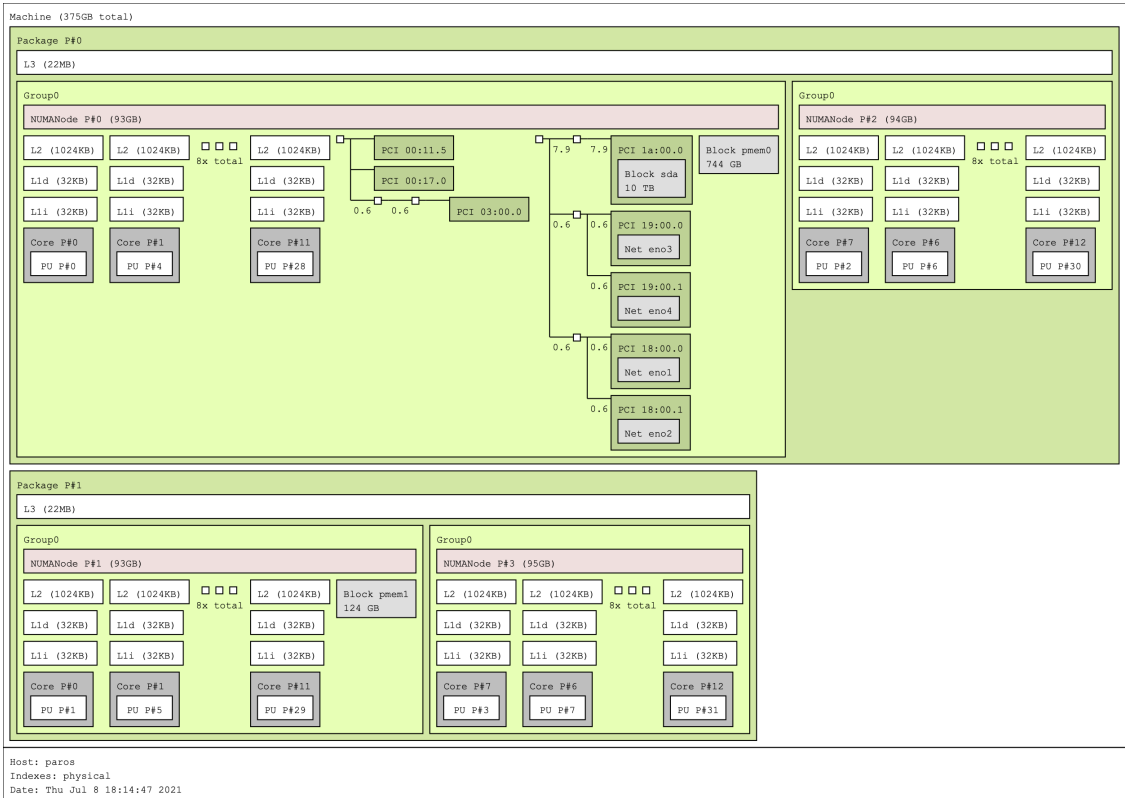
## No Performance predictions

## No Optimized mapping decisions

## No costs model

Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.

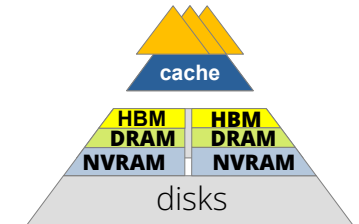
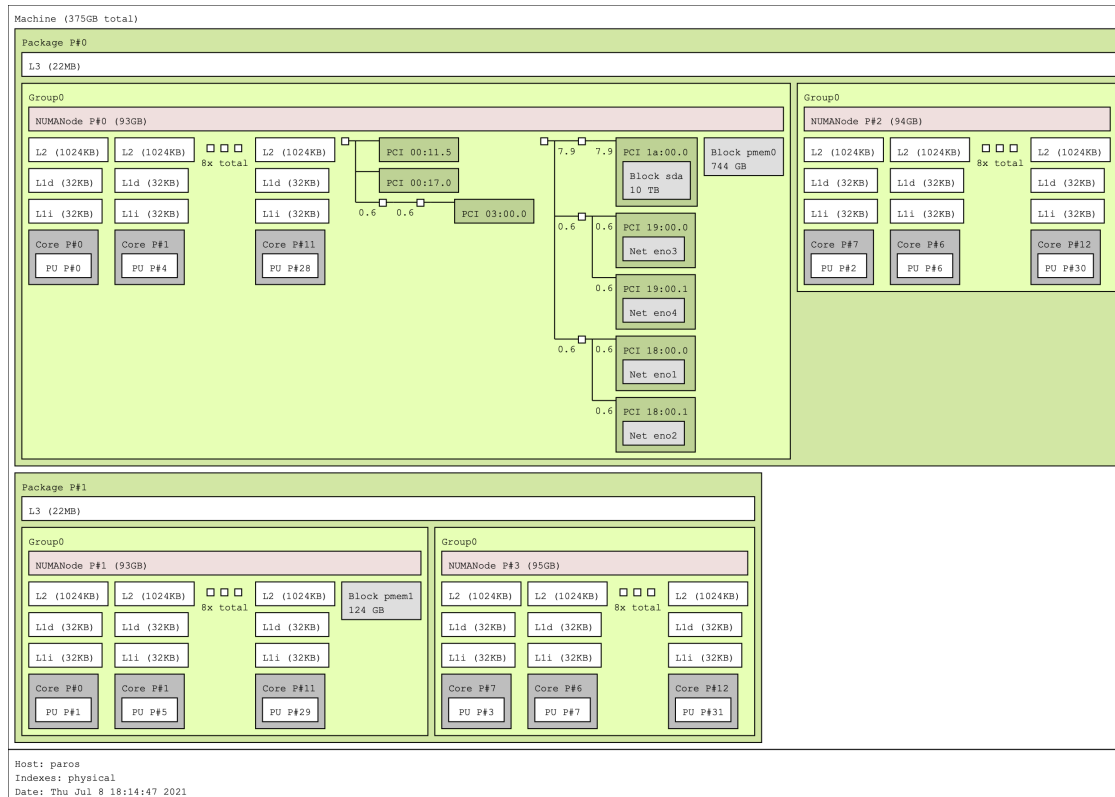
# System models: hwloc



- Support for DMT
- User-space library
- No Performance predictions
- No Optimized mapping decisions
- No costs model
- No updates at runtime

Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.

# System models: hwloc



- Support for DMT
- User-space library
- No Performance predictions
- No Optimized mapping decisions
- No costs model
- No updates at runtime
- Only application view
- No application model

Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S. and Namyst, R. 2010. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. *18th Euromicro Conference on Parallel, Distributed and Network-based Processing* (Feb. 2010), 180–186.



## Conclusion

- Increasing complexity of memory hierarchies calls for sophisticated system models
  - Only device-specific models and performance studies for DMTs
  - Most existing system models, do not model DMTs; And the ones that do are insufficient
- Research on holistic models for systems with DMTs needed