Resource disaggregation for the 99%

Irina Calciu, Aasheesh Kolli, Jayneel Gandhi,
Stanko Novakovic, Marcos K. Aguilera,
Rajesh Venkatasubramanian, Pratap Subrahmanyam

WAMS @ ASPLOS 2018
Ratio of network to memory latency

log scale

1000
100
10
1

1997 2007 2017
Resource disaggregation

Requires new/expensive hw

Logical disaggregation

Benefits
(1) More resources w/ less overprovisioning
(2) Better utilization and sharing
(3) Decoupling of memory and compute
Logical resource disaggregation

1. Programming model
2. Resource management
3. Security
4. Performance
Programming model

Simplicity

Complete transparency

Performance

App makes all decisions
Programming model

- Shared pool of memory
  - Coherent within host
  - Non-coherent across hosts
- Threads local and remote
  - Hierarchical rack-level scheduler
- Some failures exposed to the application
Failure management (component, host)

- What failures can we solve transparently?
- What failures can we ignore?
- What failures should we report to the application?
- How to contain failures?
Logical resource disaggregation

1. Programming model
2. Resource management
3. Security
4. Performance
Logical resource disaggregation

2. Resource management

- Memory management
- Compute management
- Devices
- Isolation
Remote memory [SoCC 2017]

Benefits
Better memory utilization
Big memory
Efficient sharing

Fast network
How does remote memory work?

Application host

RAM

CPU+NIC

virtual memory functionality

cache remote pages

Application host

RAM

CPU+NIC

remote pages

Rack resource manager

RDMA
How does remote memory work?

Application host

RAM

CPU+NIC

flush dirty pages

remote pages

Rack resource manager

virtual memory functionality

RDMA
Remote memory

- Cache remote pages:
  - page fault + RDMA

- Flush dirty pages:
  - Write protect page
  - Page fault on write
  - Dirty data amplification
Granularity

Data movement
- RDMA latency;
- Dirty data amplification

Memory management (Local and remote)
- Metadata size;
- Translation overheads
Do we need new hardware?

• Large software overhead
• Fix system software first

• Possible: Cache-coherent FPGAs
• Network attached
• Could enable more efficient data movement

• Good platform for experimenting with new hw
Technical challenges

1. Virtual memory overheads
2. Dirty data amplification
3. Local cache management
4. Remote memory allocation
5. Sharing model
6. API & transparency level
7. Remote host crashes
8. Network slow or congested
9. Non-uniform latency
10. Virtual machine indirection
11. Remote host compromised
Logical resource disaggregation

2. Resource management

- Memory management
- Compute management
- Devices
- Isolation
Scaling compute

- Create local thread
- Create remote thread
- Shared memory across remote threads
Heterogeneous compute

- Many heterogeneous compute elements
- Heterogeneous cores, FPGAs, GPUs, etc.
- Accessing accelerators across the rack
- Scheduling across heterogeneous compute elements
Conclusion

Resource disaggregation is becoming feasible, but still risky and expensive for most companies.

Logical resource disaggregation presents most of the benefits, at a lower cost.

Still, many technical challenges for the community.
Thank you!

We’re hiring!
We are looking for students with FPGA expertise for full-time/internship roles

icalciu@vmware.com
https://research.vmware.com/