Are Edge MicroDCs Equipped to Tackle Memory Contention?

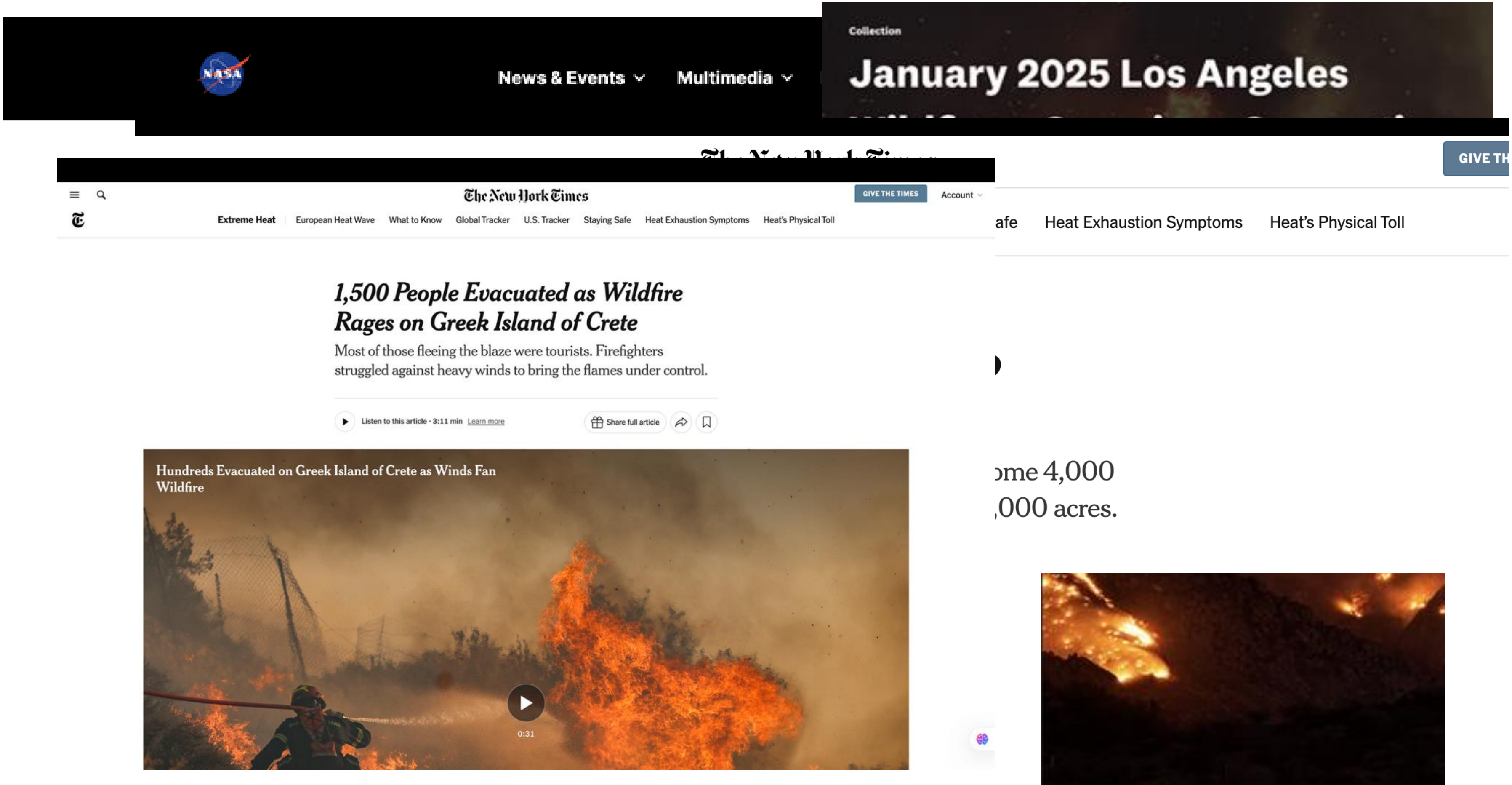
Long Tran*, River Bartz, Ram Durairajan, Ulrich Kremer*, Sudarsun Kannan*
Rutgers University*, University of Oregon

HazardMon
Resiliency, Sustainability, Security

RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

NSF

Wildfire Incidents



Current Solutions

FIRMS
Fire Information for Resource Management System

OroraTech

IQ FireWatch

DRYAD
CONNECTING THE NATURAL WORLD

SAGE
A Software-Defined Sensor Network
Cyberinfrastructure for Edge Computing

Current Solutions

SAGE A Software-Defined Sensor Network
Cyberinfrastructure for Edge Computing

[Waggle IEEE 2016]
[ARMing the Edge]
[Performance Monitoring on Sage Continuum Edge Devices]

- Support real-time environmental monitoring
- Support data analytics and AI solutions at the edge

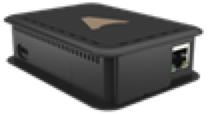
Edge Node

[Waggle Node IEEE 2016]

Edge Node

[Waggle Node IEEE 2016]

Wireless/
Ethernet/
Cellular

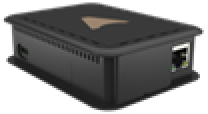


Gateways

Edge Node

[Waggle Node IEEE 2016]

Wireless/
Ethernet/
Cellular



Gateways

Solar-powered/
Intermittent grid
MicroDCs

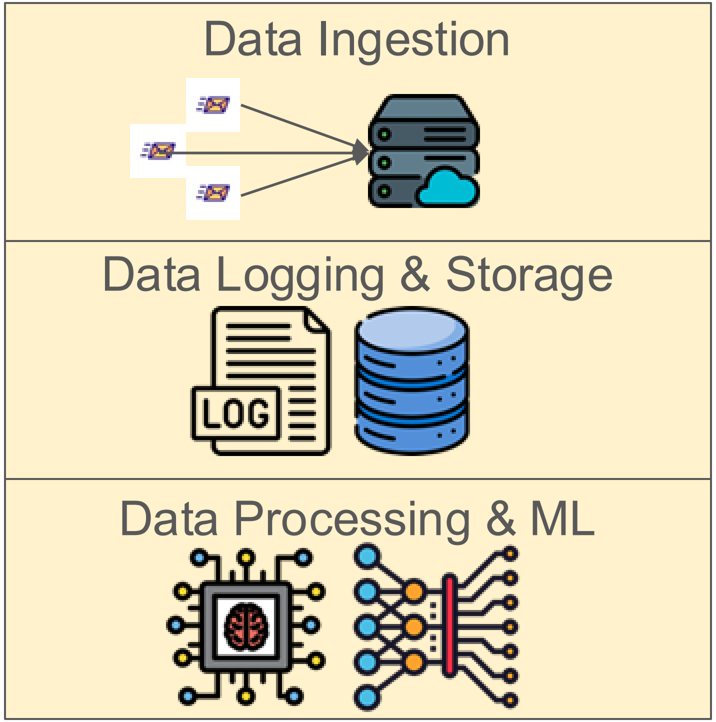
Limited compute
Limited memory

Application Deployment

- Popular option – Containerization
- Portable deployment
- Isolate application resources

Dynamic Workloads

In MicroDCs, multiple applications running have diverse memory allocation and access patterns



RabbitMQ

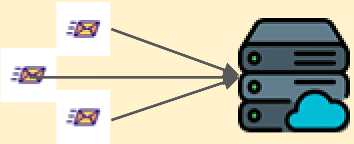
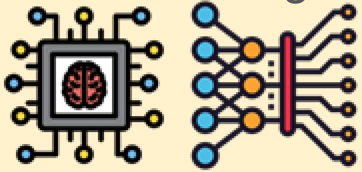
RocksDB

PyTorch

OpenCV

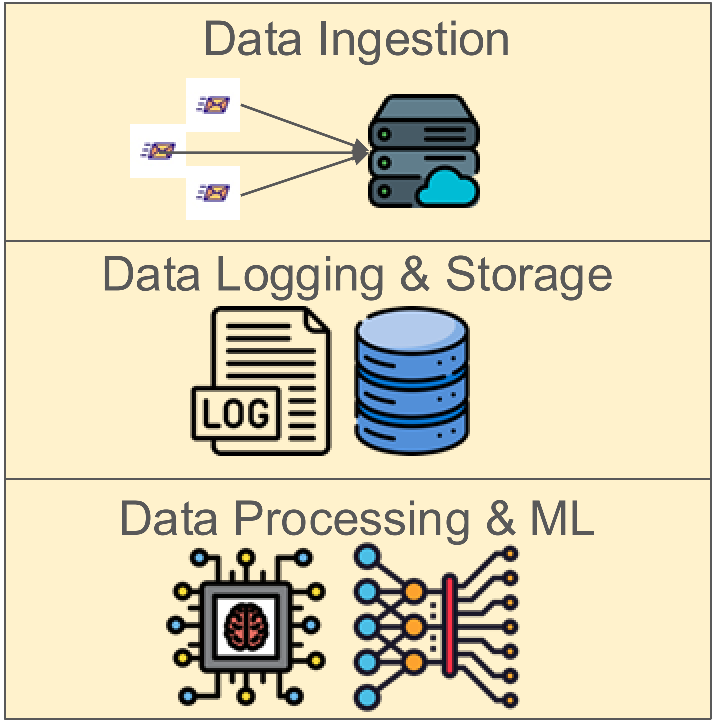
Dynamic Workloads

In MicroDCs, multiple applications running have diverse memory allocation and access patterns

Data Ingestion 	Kernel heap buffers User-space buffers
Data Logging & Storage  	Look-up Tables File-backed Page Cache
Data Processing & ML 	Heap allocated Tensors Data Transfer with GPU

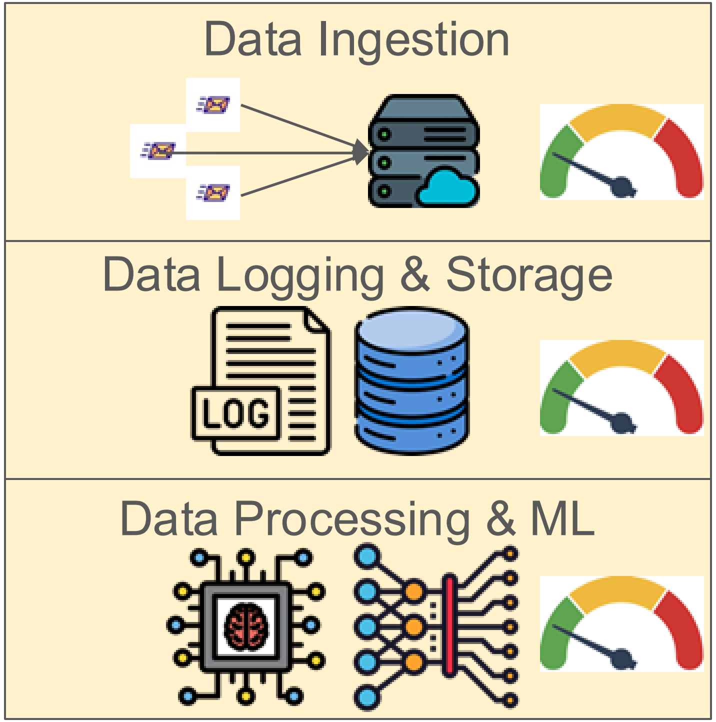
Dynamic Workloads

In MicroDCs, multiple applications running have diverse memory allocation and access patterns



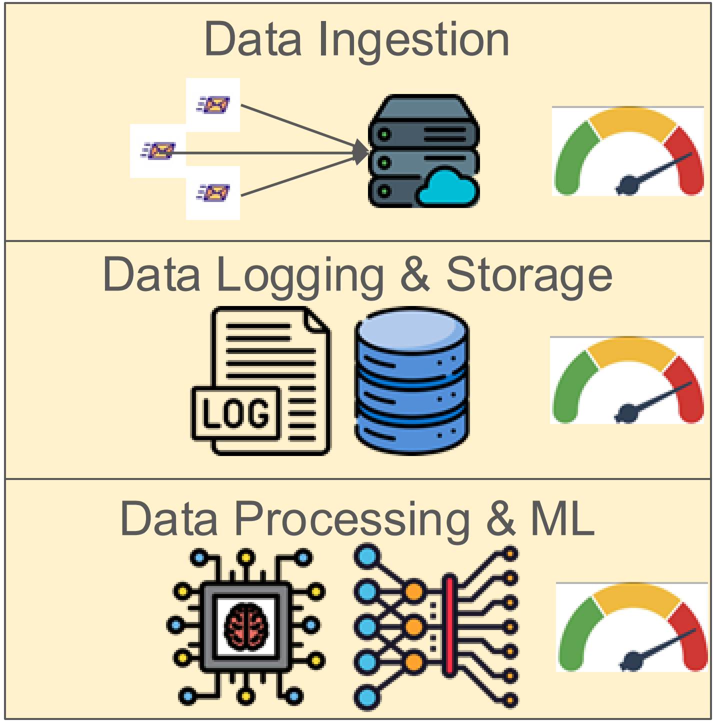
Dynamic Workloads

In MicroDCs, multiple applications running have diverse memory allocation and access patterns



Dynamic Workloads

In MicroDCs, multiple applications running have diverse memory allocation and access patterns



Memory contention!!!

Increasing Memory

- MicroDCs are deployed at the remote environment across the U.S
- Cost constraints
- Increasing in energy consumption
  - Static power - unnecessary energy when memory is not intensive
- Contributing to the embodied carbon footprint
  - ~48g / GB for LPDDR4 SK Hynix
  - Estimation: each 16GB DDR4 produces 768g CO_2

Conflicts with sustainability objectives of hazard monitoring infrastructure.

Outline

- Background
- Motivation
- Design
- Conclusion

Multitenancy

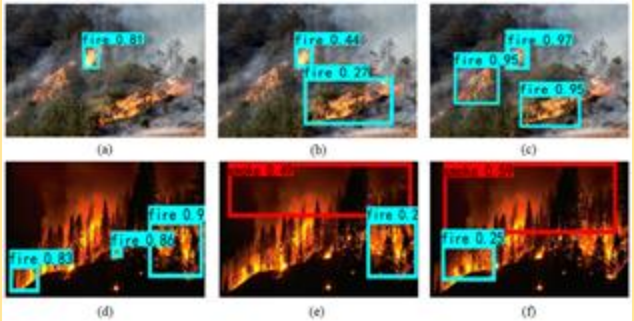
- MicroDCs are designed for near-sensor processing
- MicroDCs are the last layers of computation in SAGE
- MicroDCs are also used for ML training
- In emergencies, edge node off-loads computing tasks (e.g., wildfire detection) to MicroDCs

Multitenancy is a requirement, not a design choice

Experiment

Representative applications:

YOLO



- Data processing & ML-based computer vision on MicroDCs
- Real-time inference over wildfire data
- Heavy heap memory

[You Only Look Once, ArXiv 2015]

Experiment

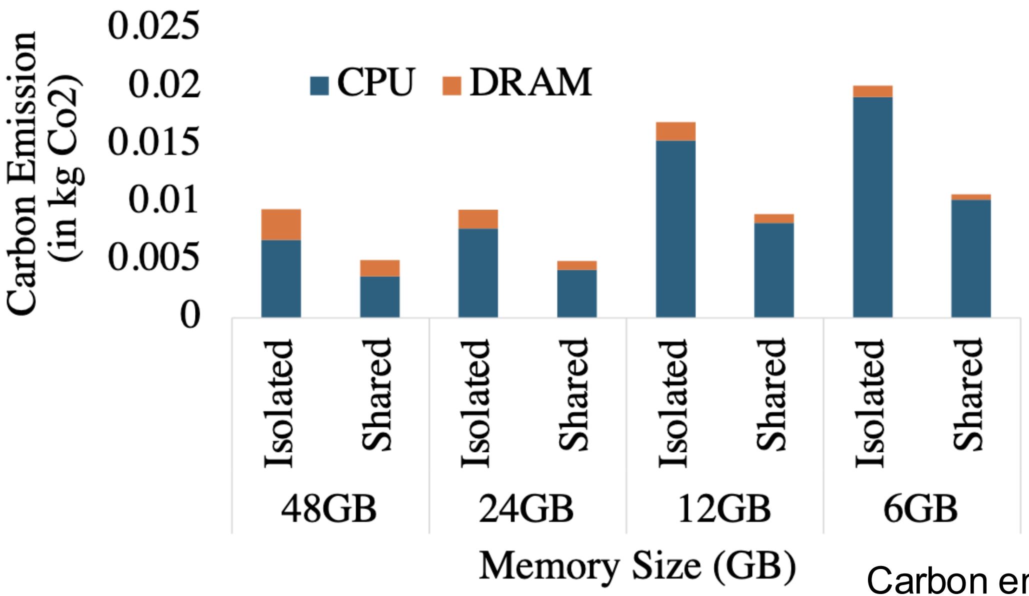
Representative applications:

RocksDB

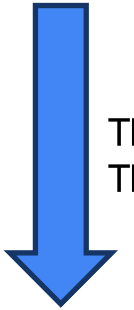
- Data logging & storage on MicroDCs
- Manages large volumes of sensor data, wildfire imagery and even logs.
- Relies on page cache

[Facebook, 2012]

Multitenancy – Impact on Carbon Emission



(c) Total Carbon Emission.



The lower
The better

Carbon emission model from [Architectural carbon modeling tool, ISCA'22]

Better utilization of hardware
Avoid wasted static power
Saving operational & embodied carbon emissions

Towards application centric carbon emission management, Sudarsun Kannan, Ulrich Kremer [HotCarbon '24]

Multitenancy – Impact on Memory Contention

- YOLO:
  - Dataset ~38k images – 7.7GB
  - 8 processes do inference in parallel

- RocksDB:
  - Dataset: 20 million key-value pairs – Value size 4KB – 43 GB
  - Read 10 million random keys in database

1) Evaluate the performance of isolation execution
2) Evaluate the performance of shared multitenant execution

Multitenancy – Impact on Memory Contention

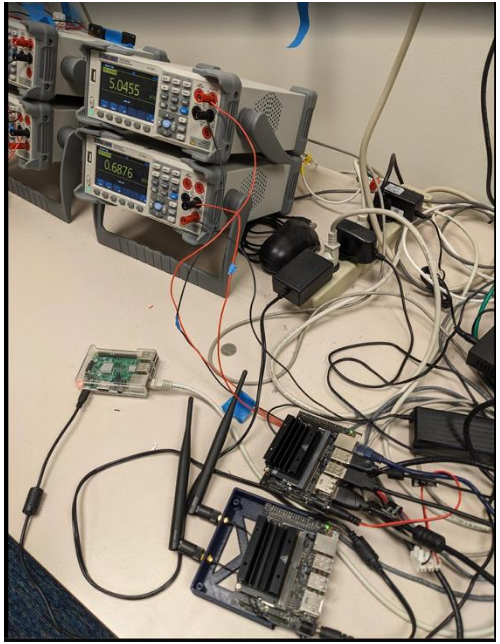
Hardware:

- Single-socket system
- a 16-core 2.1GHz Intel(R) Xeon(R) Silver 4110 processor
- 48GB DRAM
- 512GB NVMe SSD
- Nvidia Quadro P5000 16GB GPU

OS:

- Linux 5.15
- Supports training & inference pipelines

Multitenancy Impact on the Edge (Future Work)

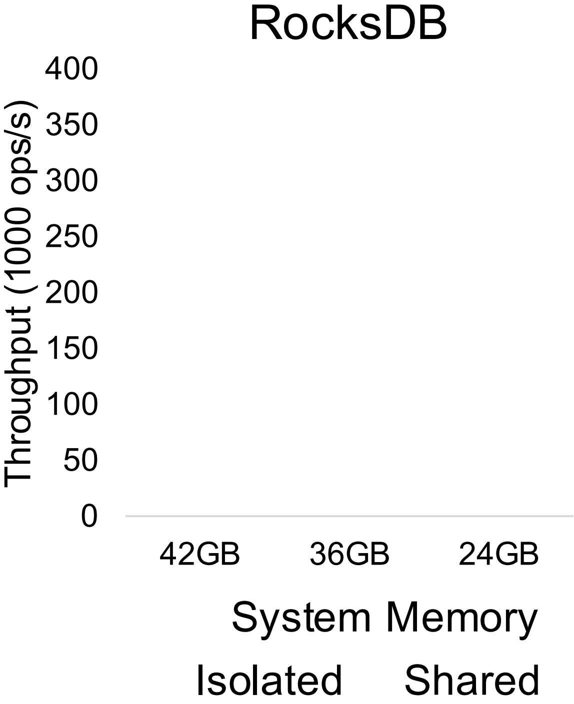


Our ongoing work explores multitenancy impact on both local edge setup and real SAGE deployments

Not discussed in this talk

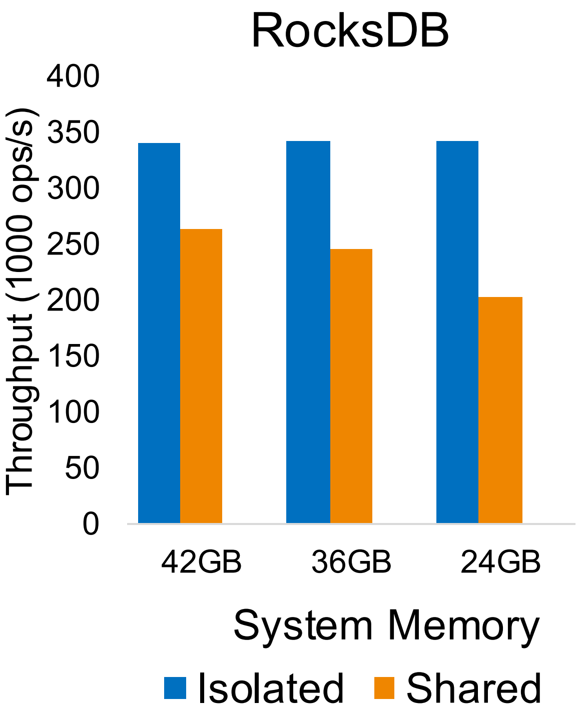
Multitenancy – Impact on Memory Contention

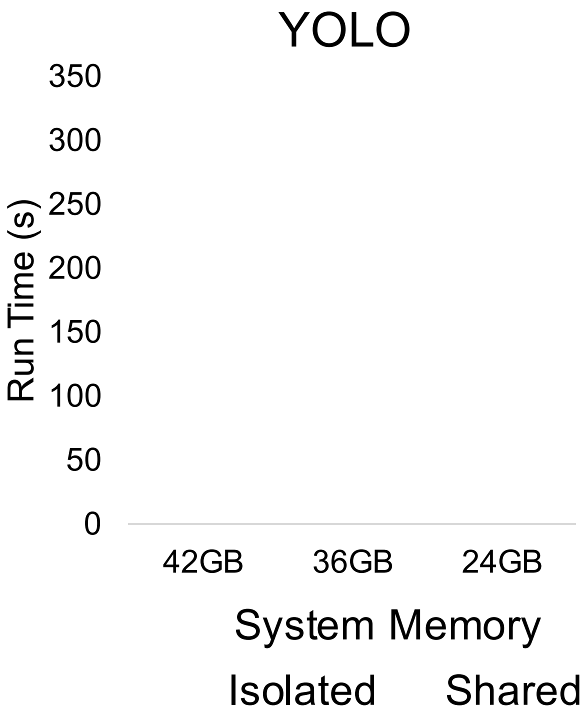
The higher
The better



Multitenancy – Impact on Memory Contention (

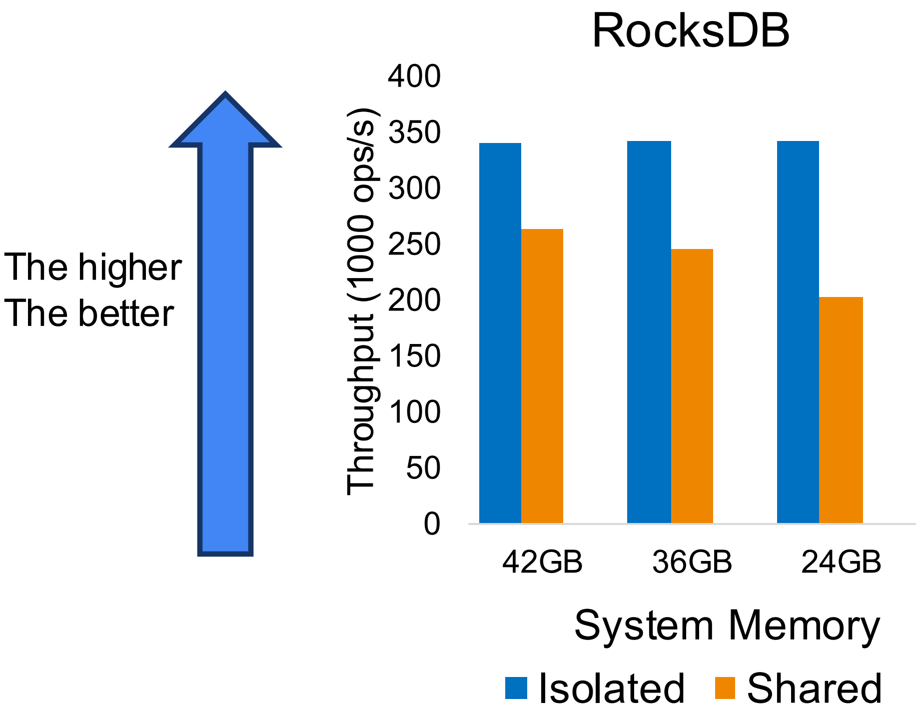
The higher
The better

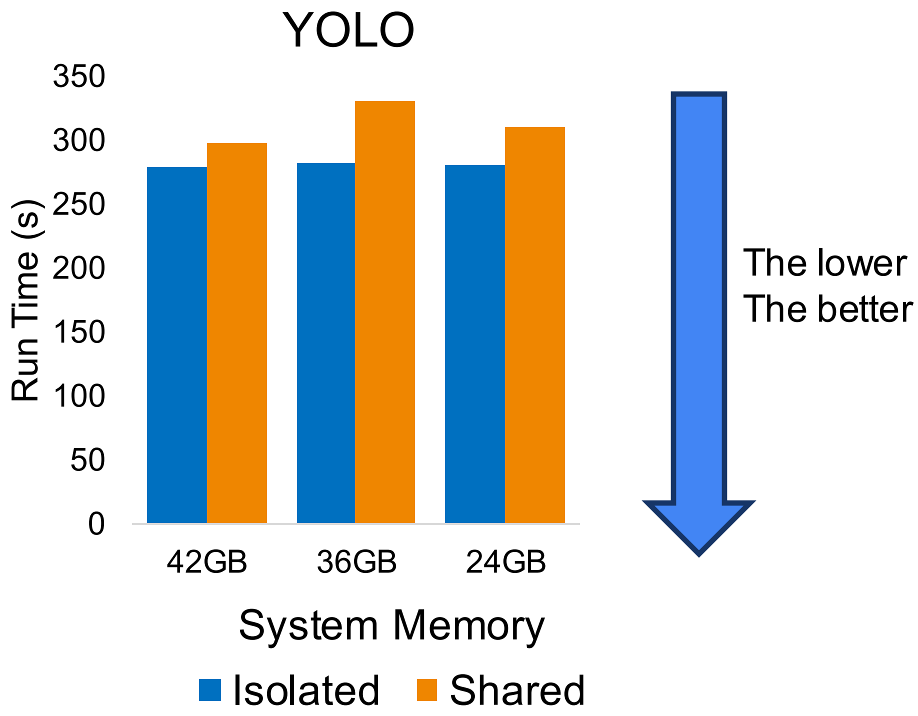




The lower
The better

Multitenancy – Impact on Memory Contention





RocksDB performance reduces significantly due to reliance on page cache

Multitenancy – Impact on Memory Contention in MicroDCs

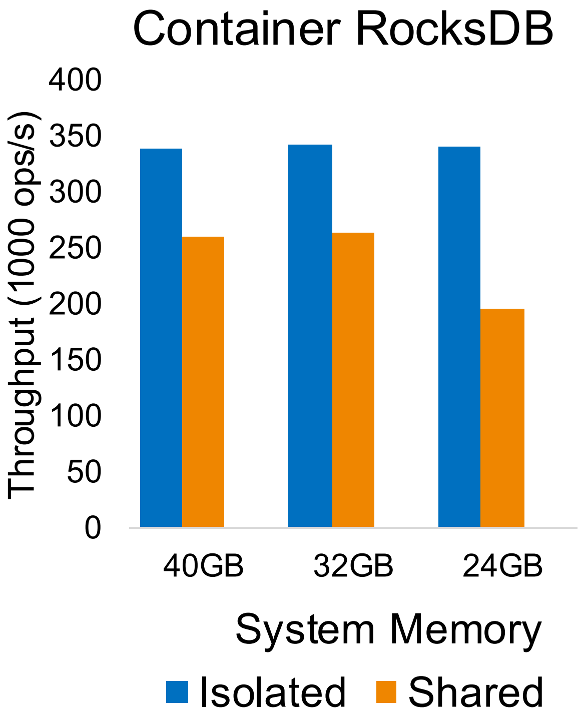
- YOLO:
  - Dataset ~38k images – 7.7GB
  - 8 processes do inference in parallel
  - Containerized using cgroups
  - 8GB memory limits – best isolated performance

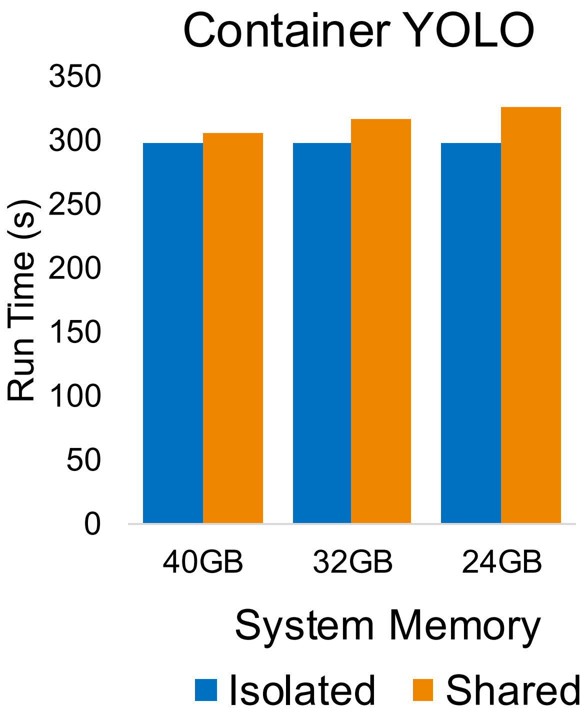
- RocksDB:
  - Dataset: 20 million key-value pairs – Value size 4KB – 43 GB
  - Read 10 million random keys in database
  - Containerized using cgroups
  - Assigned the rest of system memory

1) Evaluate the performance of isolation execution
2) Evaluate the performance of shared multitenant execution

Memory Contention Experiment

The higher
The better





The lower
The better

Same pattern with the previous experiment.

Multitenancy – Impact on Memory Contention

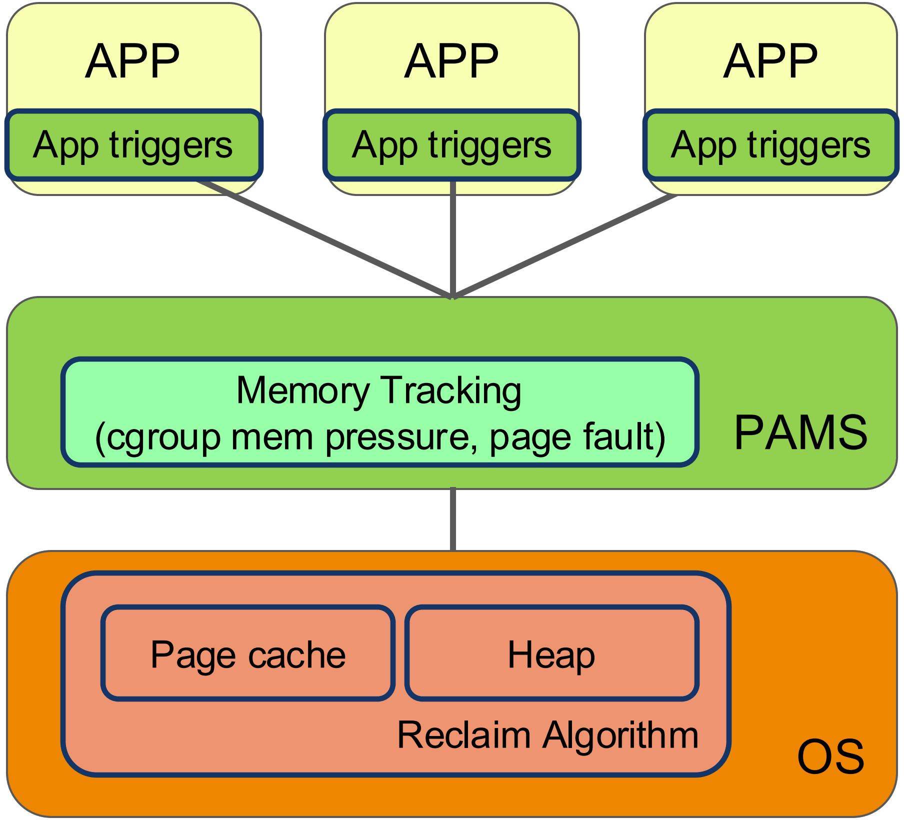
- What did we learn?
  - Asymmetric performance degradation under contention
  - Static partitioning fails to consider various memory types

Initial Design

PAMS: Multitenancy Resource Management Framework

- Unified Treatment of Memory Types
- Application-level Triggers
- Predictive Resource Allocation

Initial Design

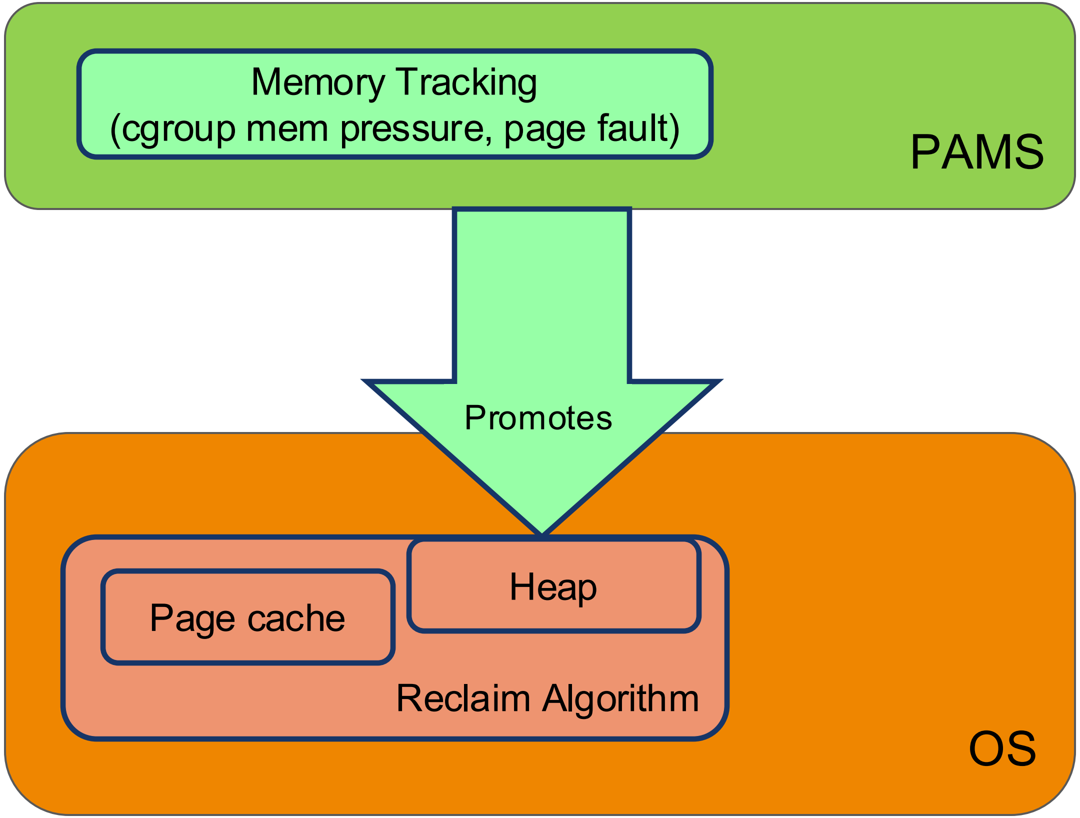


Application-level Triggers

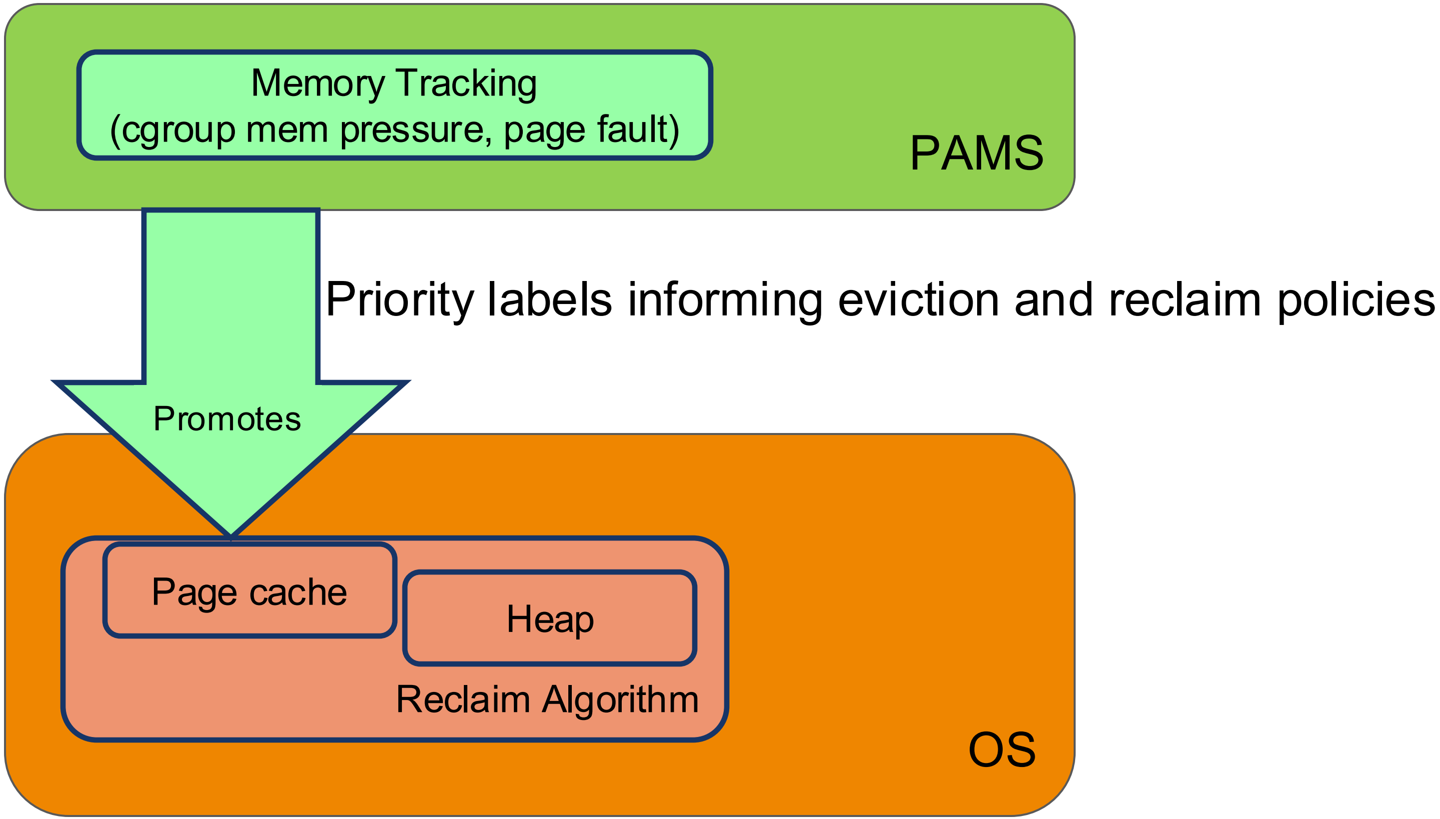
Predictive Resource Allocation

Unified Treatment of Memory Types

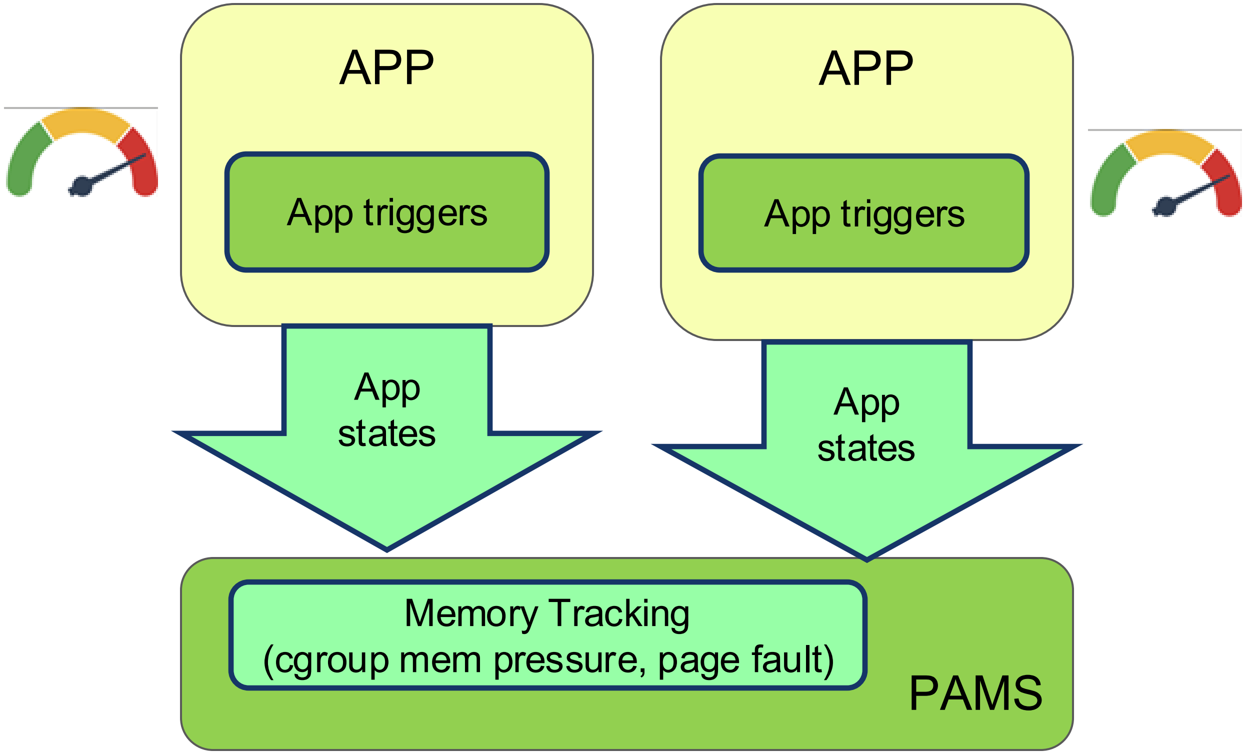
Unified Treatment of Memory Types



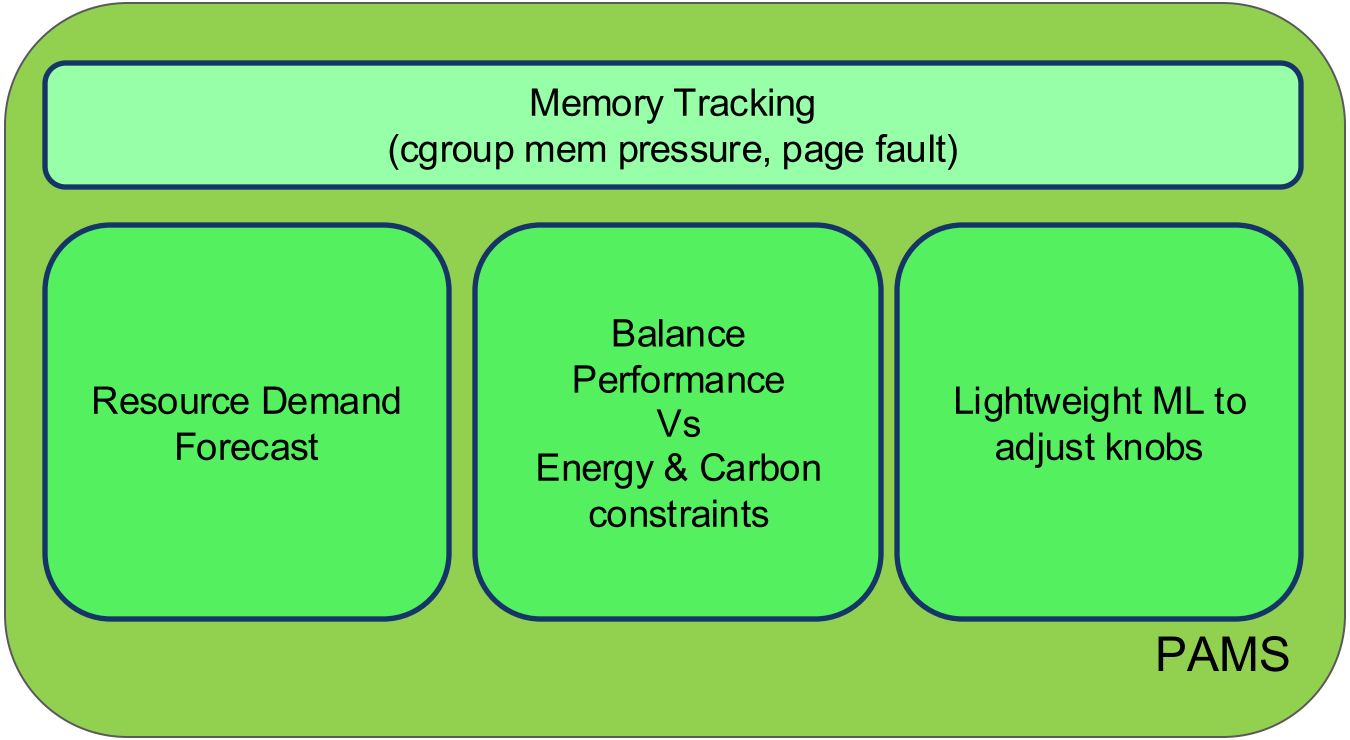
Unified Treatment of Memory Types



Application-level Triggers



Predictive Resource Allocation



Summary and Future Work

- Identifies challenges of multi-tenancy on MicroDCs for hazard monitoring.
- Motivates the need of cross-layered approach to manage memory dynamically.
- Explore whether PAMS can be implemented without significant changes to OS.

Thanks!
Long Tran (lht21@scarletmail.rutgers.edu)

HazardMon
Resiliency, Sustainability, Security

RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY