Towards Special-Purpose Edge Computing

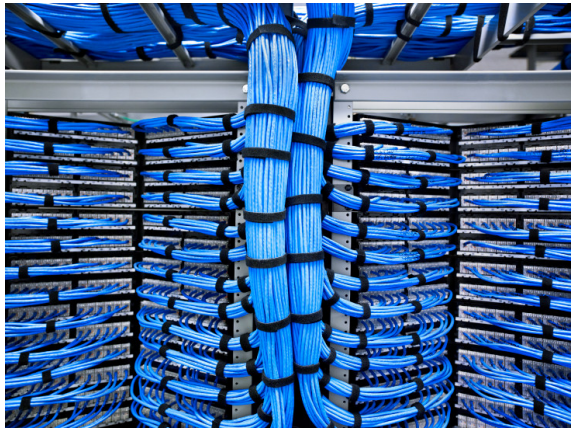
Prashant Shenoy

University of Massachusetts

UMassAmherst

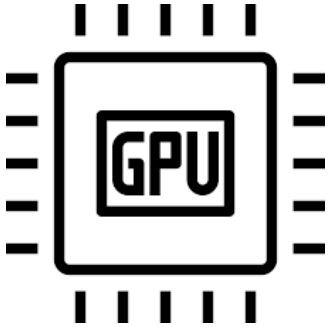
Rise of Edge Computing

- Cloud computing popular for running diverse applications
- IoT and mobile applications increasingly rely on the cloud
- Edge computing bridges the gap between:
  - **Resource-constrained** IoT/mobile devices
  - **Distant** cloud servers
- Beneficial for apps that are
  - Latency-sensitive
  - Bandwidth-intensive
  - Privacy-sensitive



Special-purpose Computing Trends

- Traditional general-purpose computing
  - Server and client machines designed for general-purpose use
  - Supports diverse applications (interactive, batch, parallel, distributed)
  - OS resource management designed to support diverse needs
- **Technology trend**: hardware continues to get cheaper
- Era of special-purpose computing
  - GPUs were the first special-purpose hardware
  - Now: Intel CPU with built-in FGPAs, network accelerator, …
  - Cloud offerings reflect this trend
    - GPU-optimized, FPGA, SSD-optimized, CPU (arm, x86), memory-optimized servers



Specializing Edge Resources

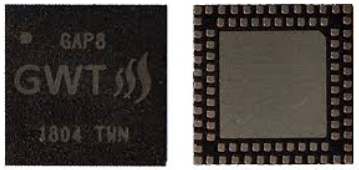
- Edge computing resources are increasingly specialized
- Common use case: **AI at the Edge**

Intel Movidius VPU

Google Edge TPU

Nvidia Jetson Nano GPU

Apple Neural Engine

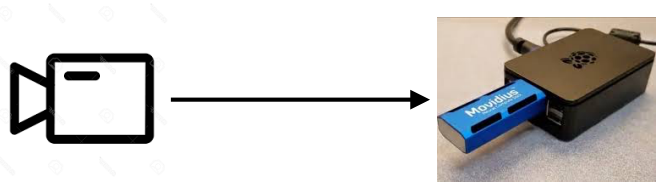


GAP8 IoT Processor

- Cost O($10-100), Power ~ few watts, accelerate specific workloads

Specialized Edge Computing

- Cheap hardware: specialize each edge deployment for workload
- Example: IoT-driven Machine Learning Analytics on the Edge
  - "Server-class" performance on low-cost hardware



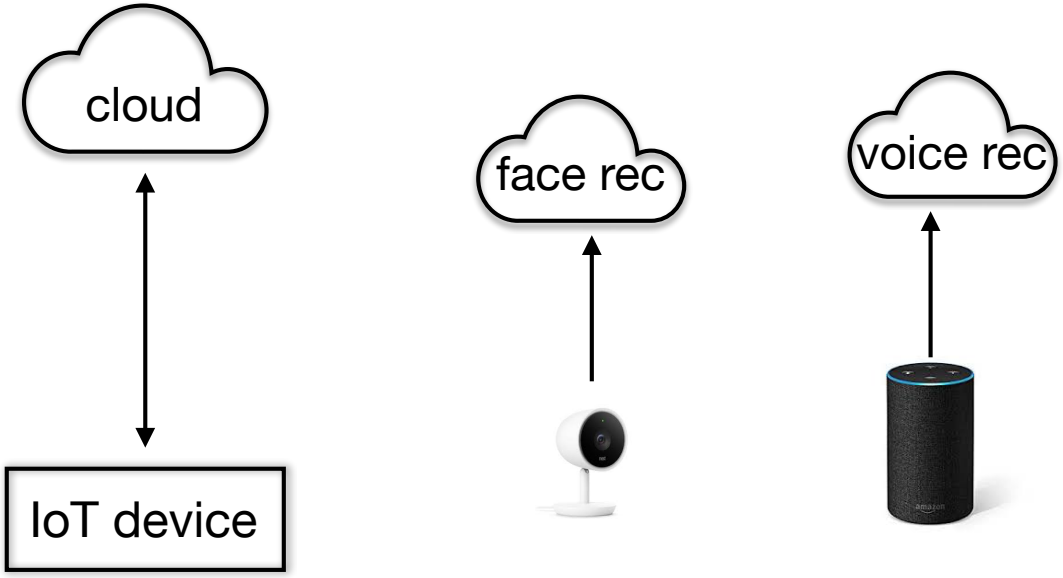
	Movidius	x86 server
Imagenet	46.3ms	24.7ms
Squzzeznet	56.5ms	22.9ms

- **Question**: *What are architectural and research challenges for realizing specialized edge computing?*

Talk Outline

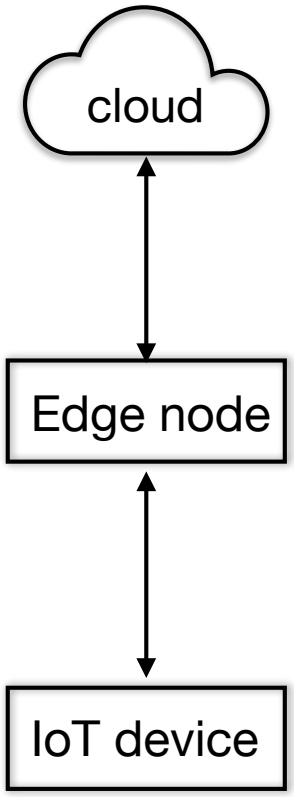
- Motivation
- **Architectural Implications**
- Research Challenges
- Conclusions

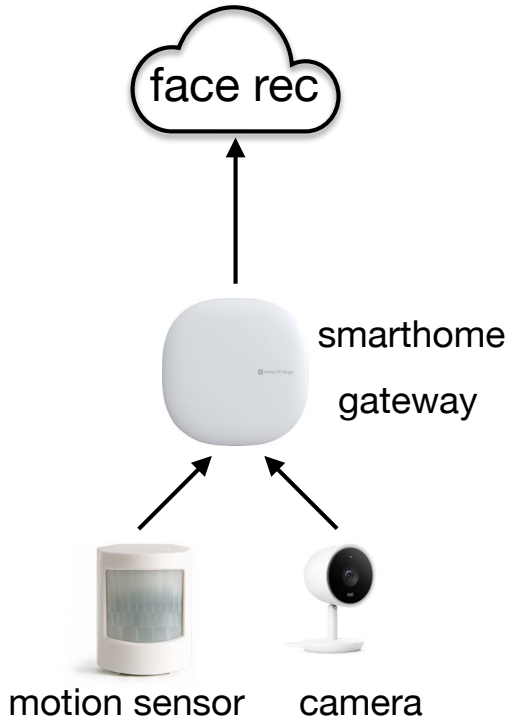
Two-tier Cloud-enabled IoT Architecture



- Computational offloading to cloud
- Two tier model
- Defacto for current IoT products

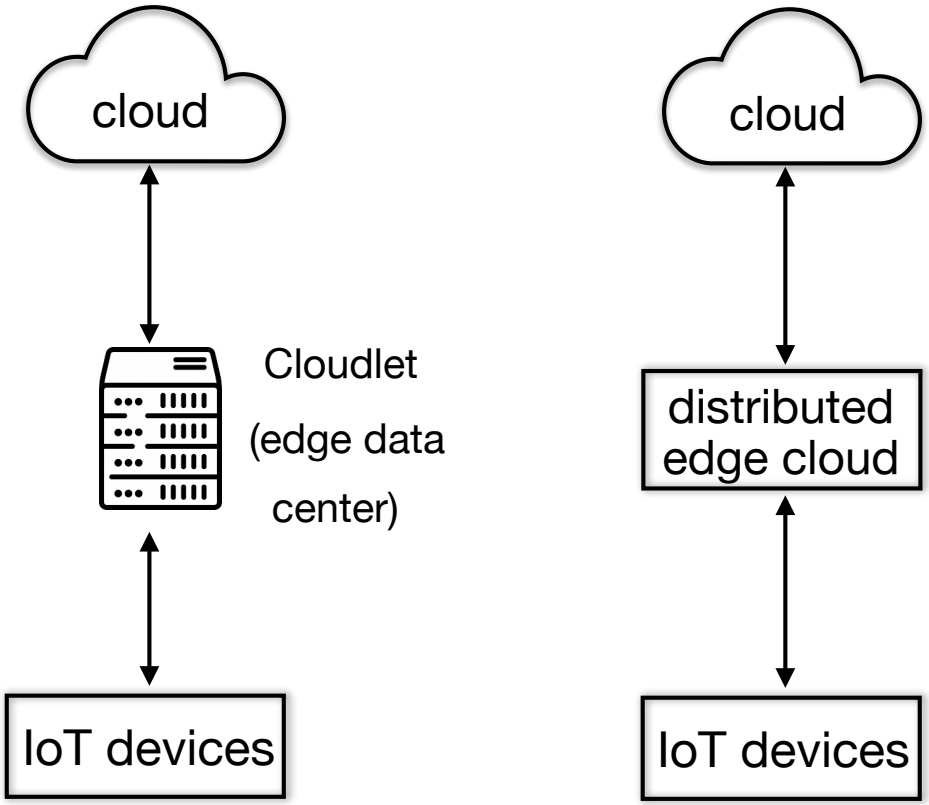
Three-tier Edge-assisted Architecture





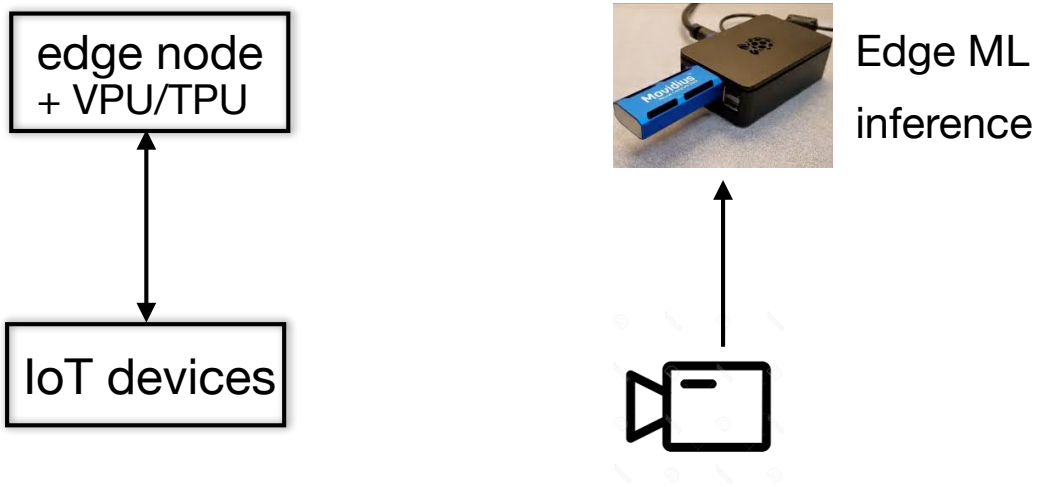
- Offload to edge+cloud
- Three-tier IoT architecture
- Edge is still resource-constrained
  - Split processing
  - Edge aggregation

Three-tier Architecture Variants



- General-purpose edge servers
- More capable edge
- Use case: AR/VR offloading
- Latency and bandwidth benefits

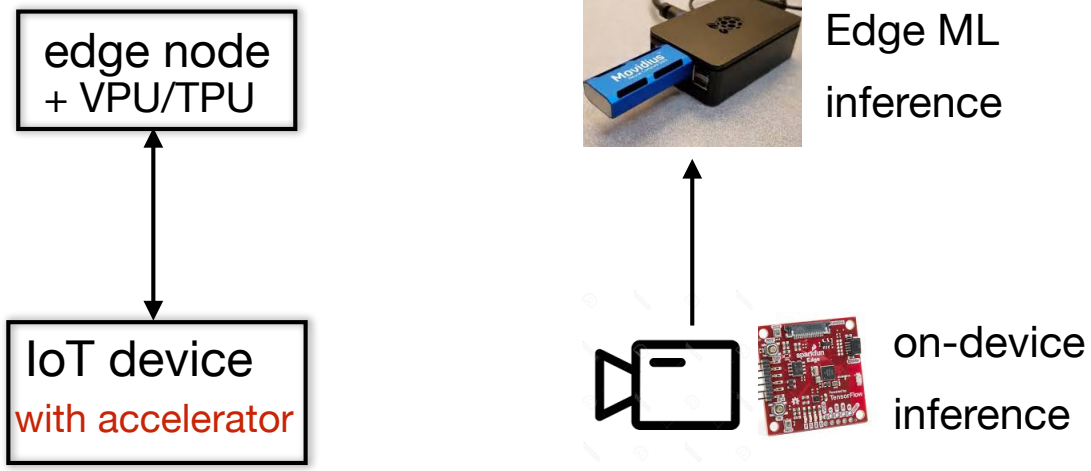
Two-tier Specialized Edge



- Specialized edge nodes
- Accelerate specific workloads
- “Server-class” performance
- Little/no cloud reliance

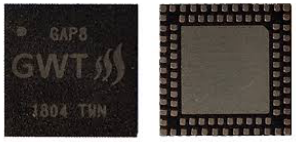
- Computation has moved from cloud back to edge!

Two-tier Specialized Edge Variants



- Devices also specialized
- Specialized edge ***and*** specialized device
- Split processing
- Further reduces cloud reliance

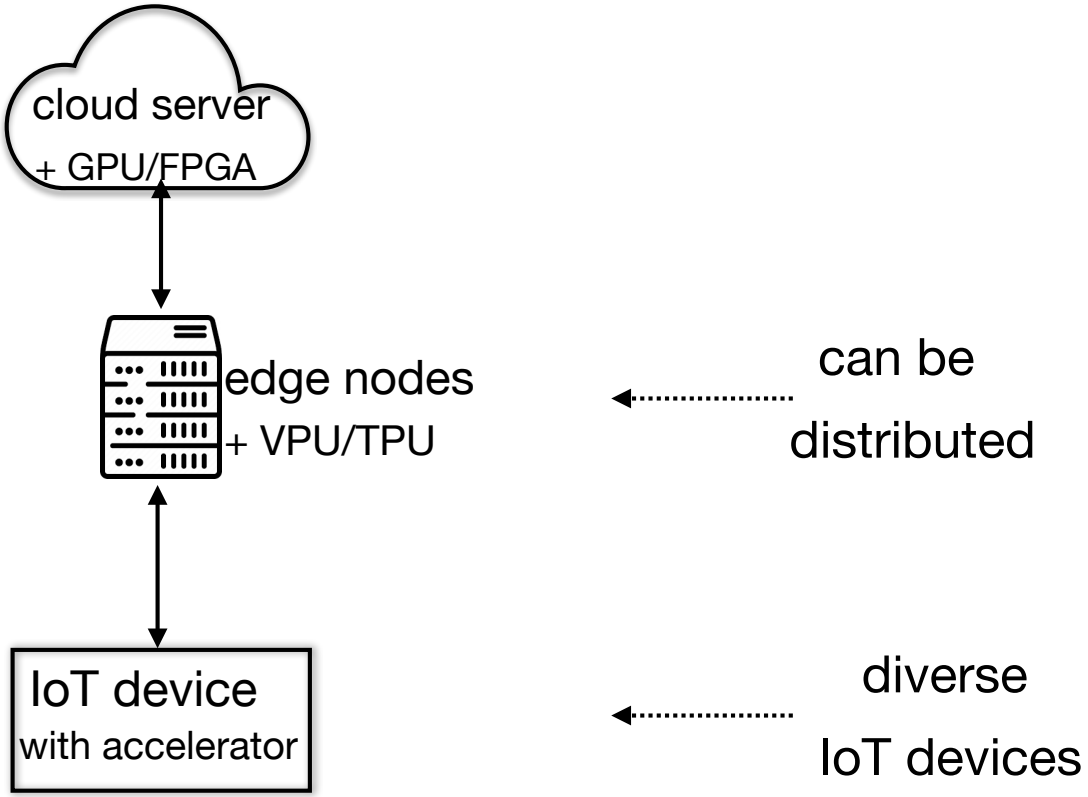
tensorflow board



GAP8 IoT processor

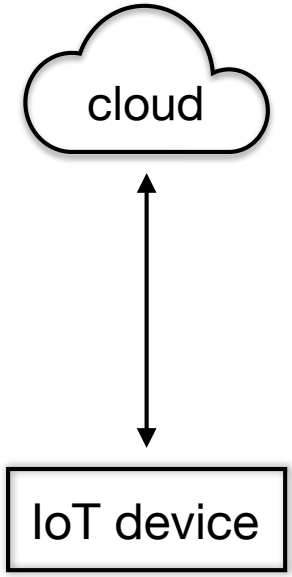
Three-tier Architecture Revisited

- Three-tier architecture in an era of specialization

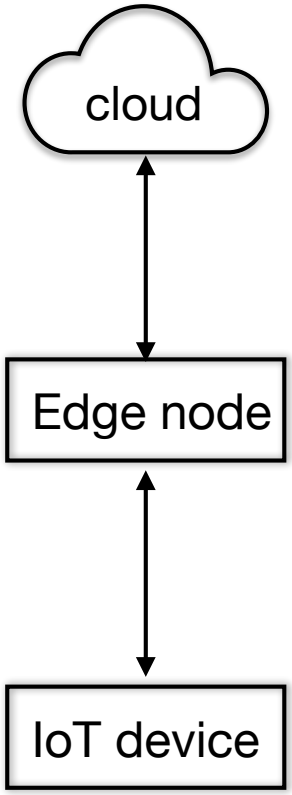


Architectural Summary

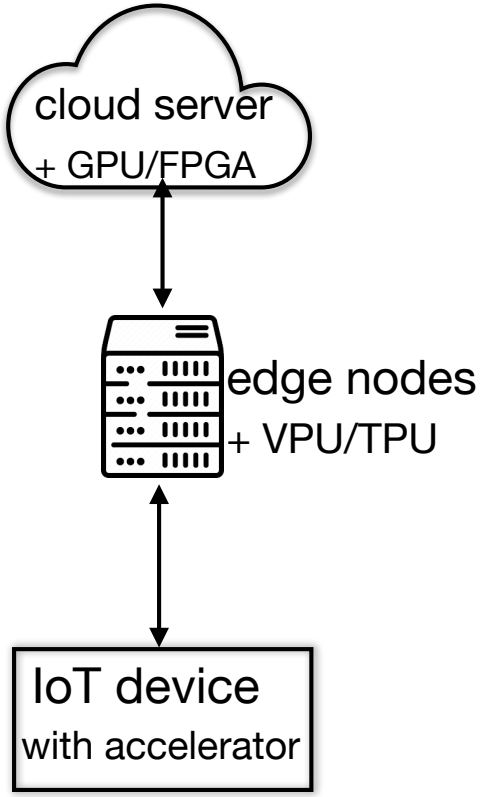
- From two-tier to three-tier to specialized three-tier architecture



Traditional cloud
(2-tier)



Traditional edge
(3-tier)



Specialized
(3-tier)

Talk Outline

- Motivation
- Architectural Implications
- **Research Challenges**
- Conclusions

Lack of Generality

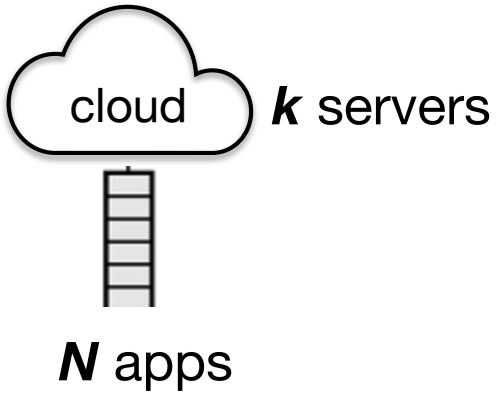
- General-purpose hardware can run a **diverse** set of applications
- Specialized edge can only run a **single** class of application
  - Lower hardware **reuse** across application classes (no multi-tenancy)
- Multiple specialized hardware configurations needed to support different application classes
  - Potentially one config per application class
  - Increases hardware costs and management complexity

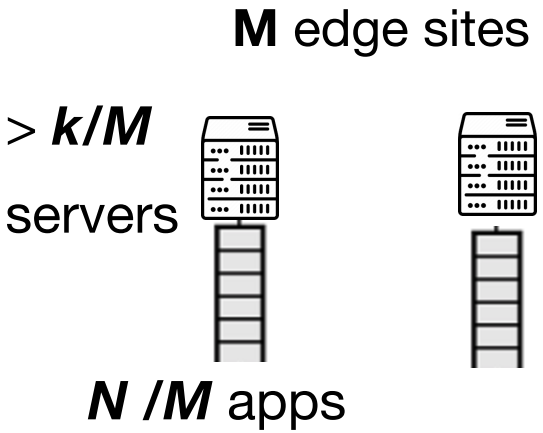
Resource Multiplexing

- General purpose nodes are time shared
  - Multiple applications can run concurrently [OS scheduling]
  - Increases system utilization and lower idling
- Specialized hardware not amenable to time sharing
  - Multiple application of same type can not share FPGA/TPU etc
  - Hardware resource idle if application can not maximize utilization
- One app to one node mapping: lower multiplexing, higher costs
- Counter-argument: hardware is cheap, deploy “lots” of cheap nodes

Cloud versus Edge Economics

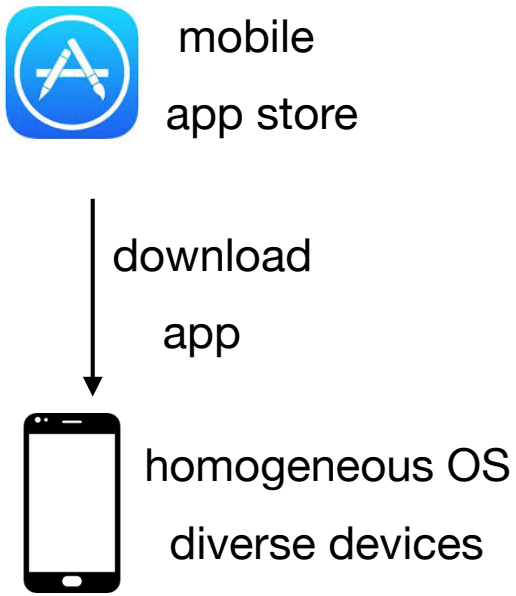
- **Cloud**: greater multiplexing benefits
  - Can host larger number of "bursty" applications
  - Multiplex app that peak at different times
- **Edge**: smaller number of servers per site
  - Lower smoothing : reduces multiplexing benefits
- Lower economy of scale for edge clouds
  - More costs to host same number of apps
  - Benefits need to outweigh costs.
  - *Specialization exacerbates these issues!*



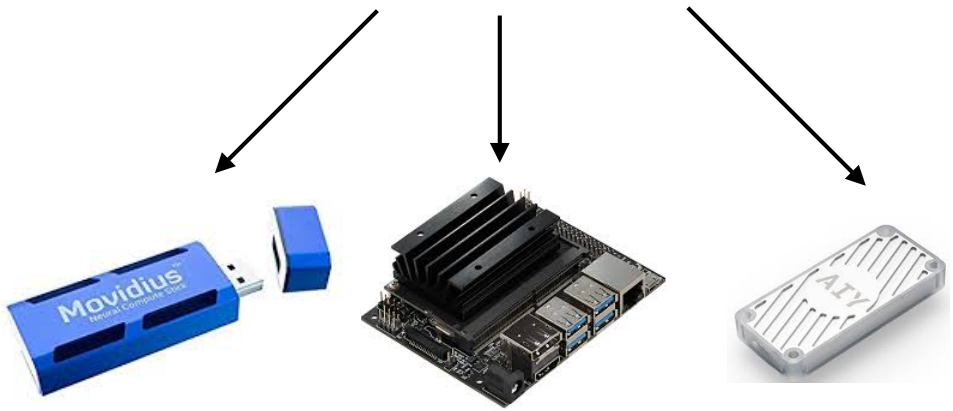


Hardware Heterogeneity

- Specialization increases hardware heterogeneity
  - Different hardware for each application class
  - Different choices in different deployments (VPU vs TPU)
- Complicates application programming

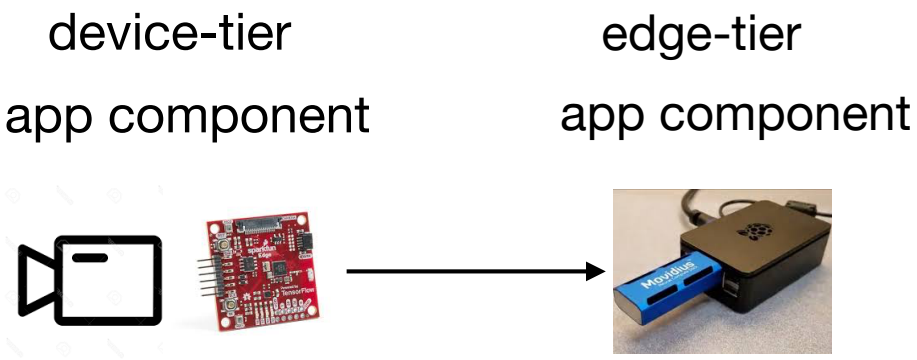


single edge/ IOT program?



Split Application Processing

- Application needs to be distributed across tiers
  - What function to put where?
- Complicates application programming



Research Challenges Summary

- Lack of generality
- Lower Resource Multiplexing
- Worse Economics for Edge

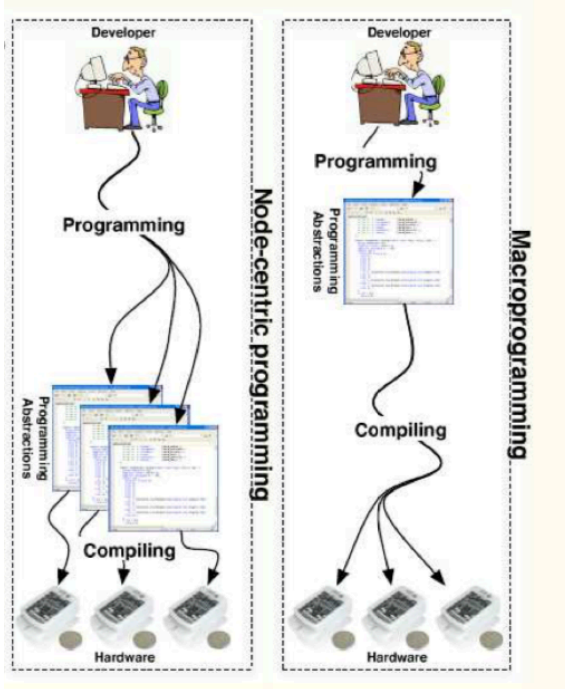
} Greater hardware complexity

- Hardware Heterogeneity
- Split applications

} Greater application complexity

Macroprogramming

- what is macroprogramming
  - Origins in sensor networks (circa 2005)
  - Specify aggregate system behavior rather than device behavior
  - Run-time application instantiation

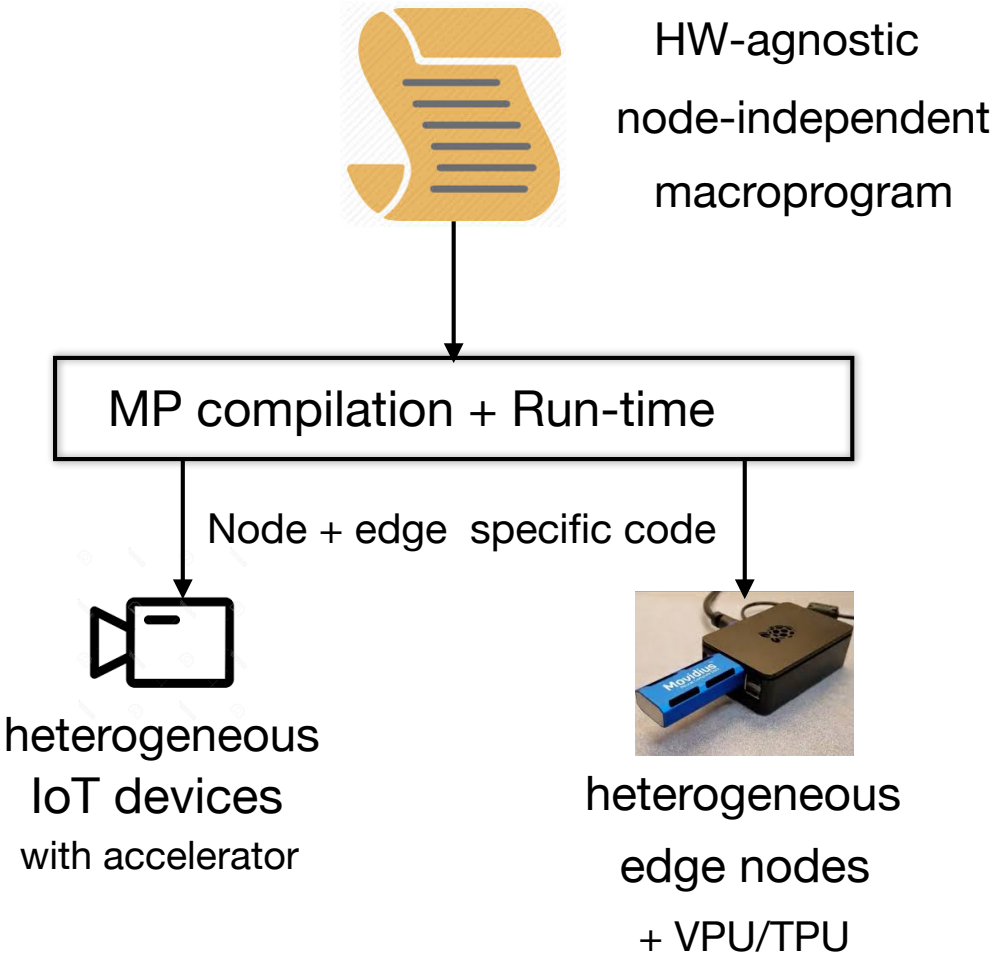


- Macroprogramming benefits for specialized edge computing
  - Hides hardware diversity from programmers
  - Write once, run anywhere
  - Program the collective, not the nodes

figure courtesy of G. P. Picco

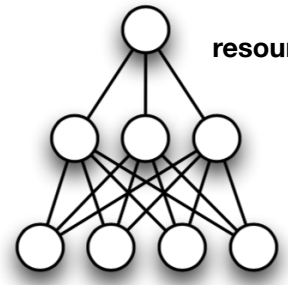
Macroprogramming Platform

- Macroprogramming run-time platform

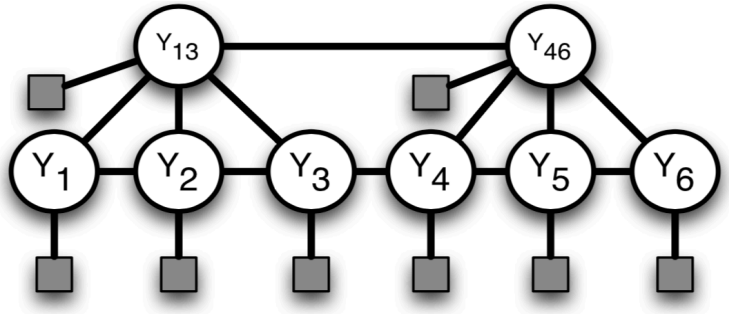


Model Compression and Splitting

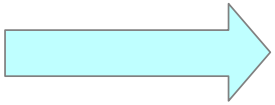
- Machine learning analytics for specialized edge
- Split machine learning inference models across multiple tiers
  - Compact coarse-grain model on device
  - Larger model on specialized edge node
- Cloud used for training the model: download, split, and deploy



resource-optimized model



wearable device



edge device / edge accelerator

figure courtesy of D. Ganesan

Automated Model Splitting

- Automatically and adaptively split the model between tiers
  - how/where to partition the network between tiers to optimize energy-accuracy-latency tradeoffs

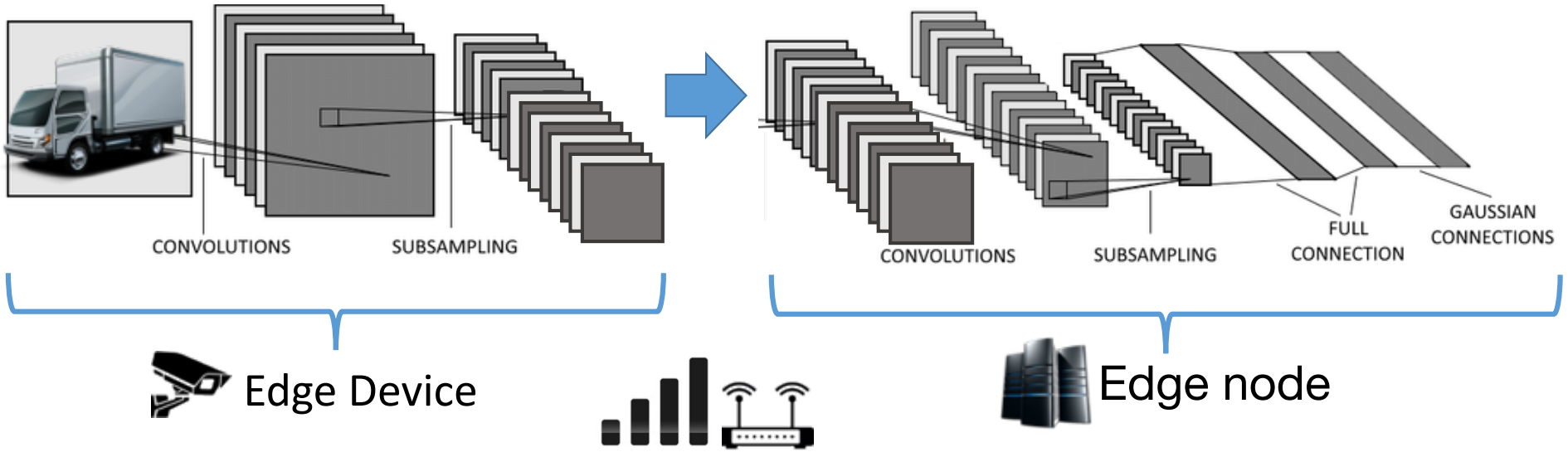


figure courtesy of Ben Marlin

Concluding Remarks

- Edge computing: significant benefits from specialization
- Many open challenges due to
  - Hardware complexity
  - Application complexity
- Need better software, run-time, and language tools to realize full potential

Thanks

- Questions?
- More at: **http://lass.cs.umass.edu**
- Acknowledgments: M. Srivastava, D. Ganesan, B. Marlin, D. Irwin