Alibaba Cloud

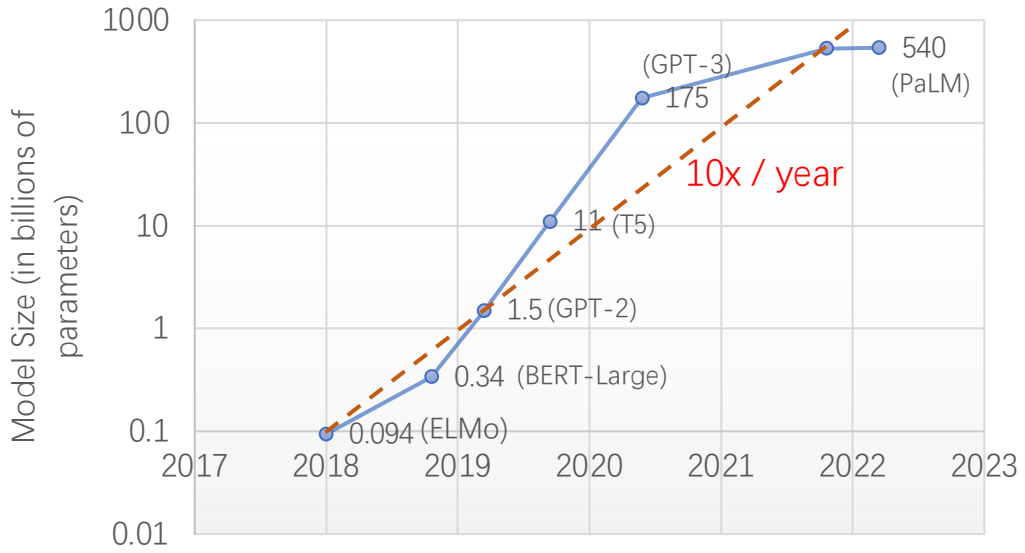
HALO: A Compiler Framework for Chiplet Architectures

Weiming Zhao
Weifeng Zhang

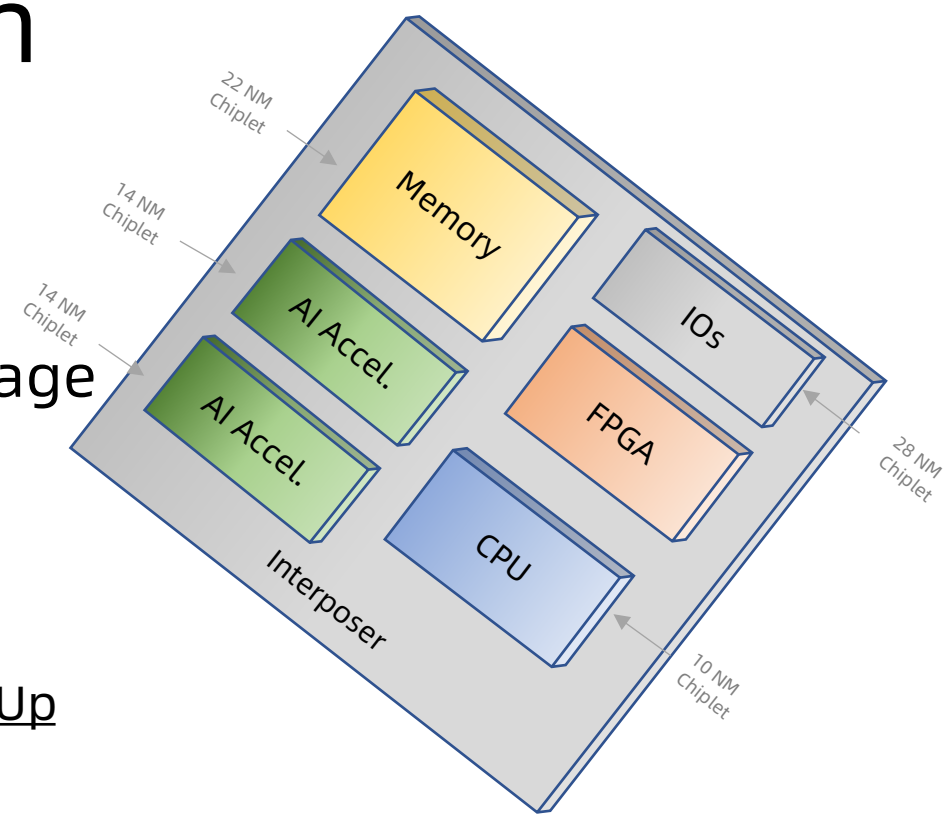
Background

- AI needs more-than-Moore computing power
  - Massive computing power needed
    - Huge dataset for training
    - Massive model parameter
  - Exponential growth
    - 10x/year increase in demand
    - Moore's law: 1.4 x / year

- To tackle the challenge
  - Scale up
    - Domain Specific Architecture (DSA)
  - Scale out
    - More cores / accelerators



Chiplet: A Promising Approach



- Chiplet
  - Integrate multiple dies (chiplets) into one package
    - Allows heterogeneous dies
    - Very low latency on die-to-die communication
- Why Chiplet for AI ?
  - Specific AI accelerator
  - End-to-end acceleration
  - Easy for processors tiling
  - Ultra fast inter-core communication
  - Optimize for various scenarios
    - Power, latency, throughput, storage, cost, security, ...
  - Shorter time-to-market
    - Critical to rapid evolving AI algorithm

Scale Up

Scale Out

Software: The Roadblock

Alibaba Cloud

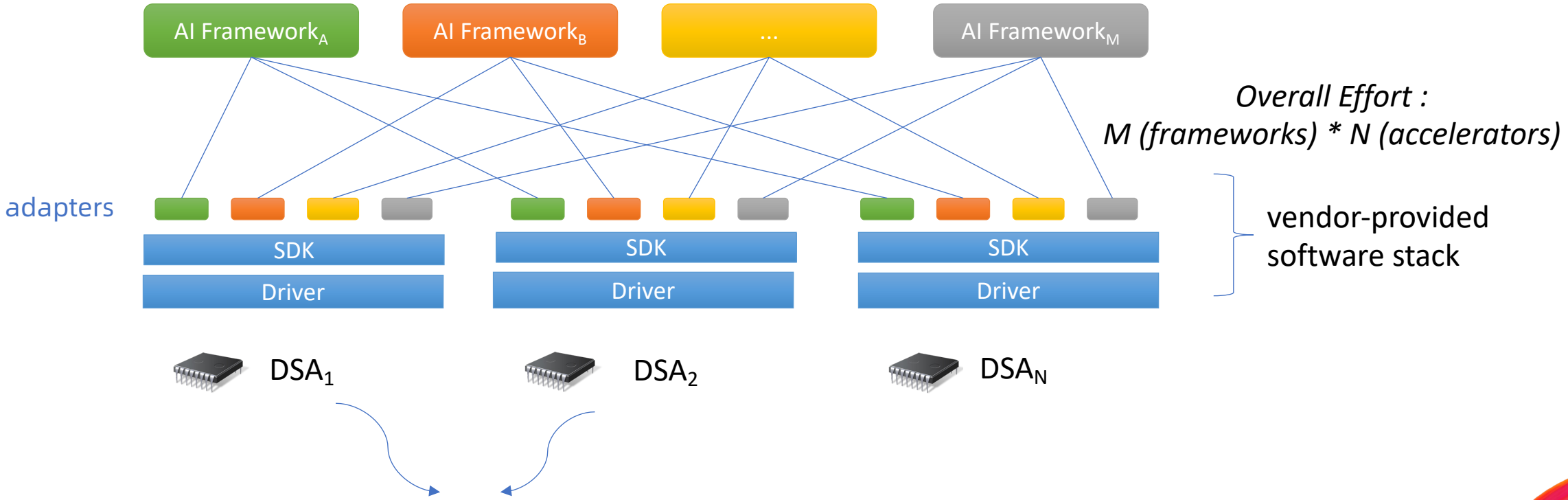
- Scale-Up: Fragmented Software Ecosystem
  - Diversified HW. 😀 → vendor-specific SW. stack ☹
  - Difficult to scale-up
    - Tremendous porting effort
    - Delays time-to-market
  - Lack of interoperability
    - Suboptimal performance

- Scale-out: More of a Software Challenge
  - Workload parallelization
  - Distributed computing system (esp. for chiplets)

Fragmented Software Ecosystem

Alibaba Cloud

Frameworks may not fit for some use cases



Overall Effort :
*M (frameworks) * N (accelerators)*

vendor-provided
software stack

No interoperability, no collaboration

Motivation & Our Philosophy

Alibaba Cloud

Motivation

Make software an enabler, not a blocker

- Significantly reduce overall effort
- Fully unleash the power of chiplets for AI
  - Easy to scale-up with more powerful HW
  - Easy to orchestrate different chiplets
  - Easy to scale-out with many processors
- Efficient and Flexible
  - No dependency on specific AI framework
  - Low memory resource footprint
  - Low runtime overhead

Philosophy

1. An unified SW layer to isolate HW discrepancy
2. Make AI algorithms run efficiently on it

Proposed Solutions (1/2)

1. A Unifying Programming Model for AI Computing
   - Open Deep Learning API (ODLA)
     - APIs for:
       (1) AI operations and pre/post-processing
         - Extensible for user-defined ops & tasks

       (2) Runtime Management
         - Device management
           - Capabilities, topologies, etc.
         - Execution management
           - Context management
           - Data movement
           - Synchronized/Asynchronized execution

     - Abstract representations for devices, computations, data, etc.
     - Implementation agnostic
       - HW specific implementation

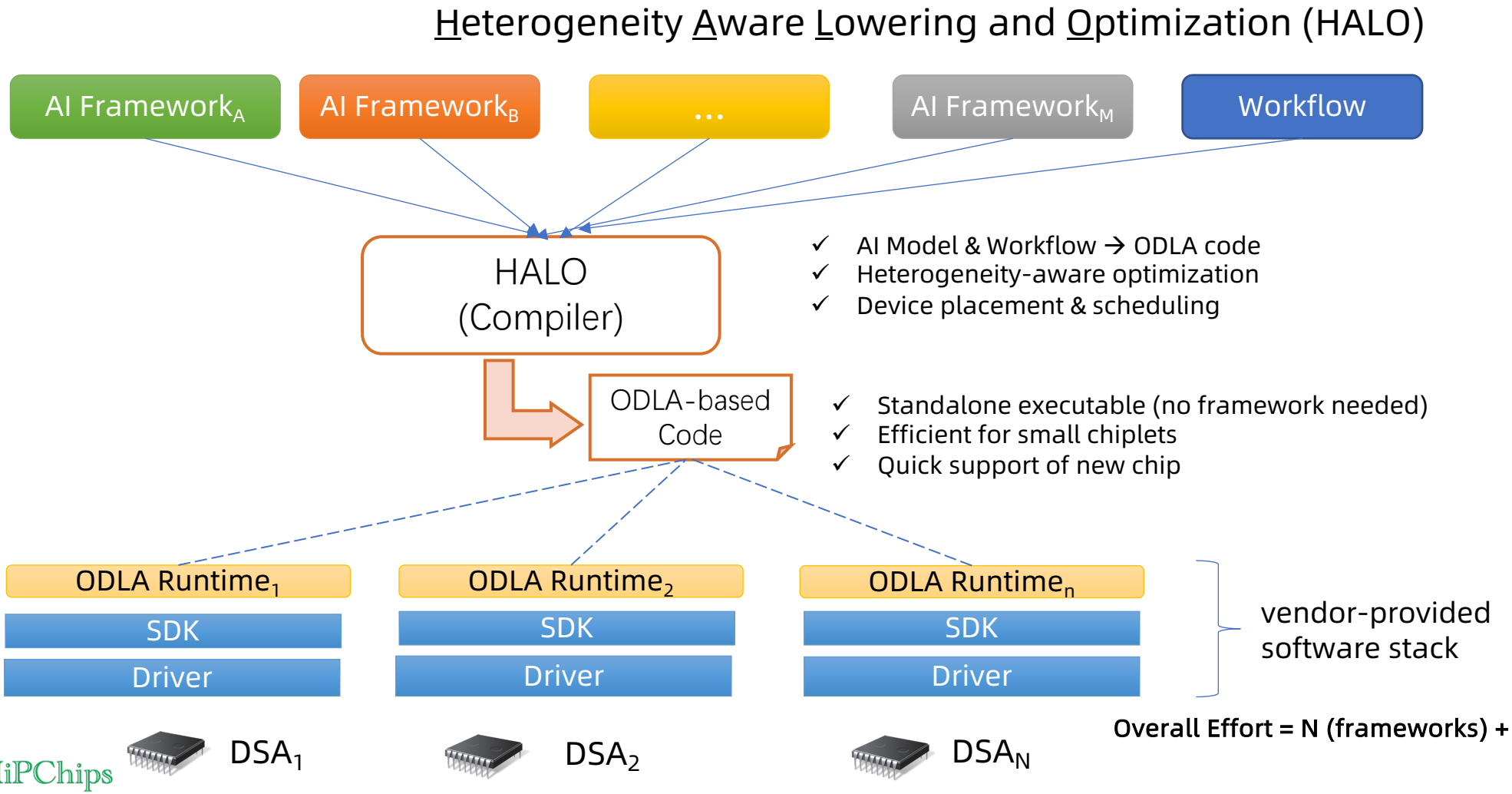
     - Simple C99-based API

Proposed Solutions (2/2)

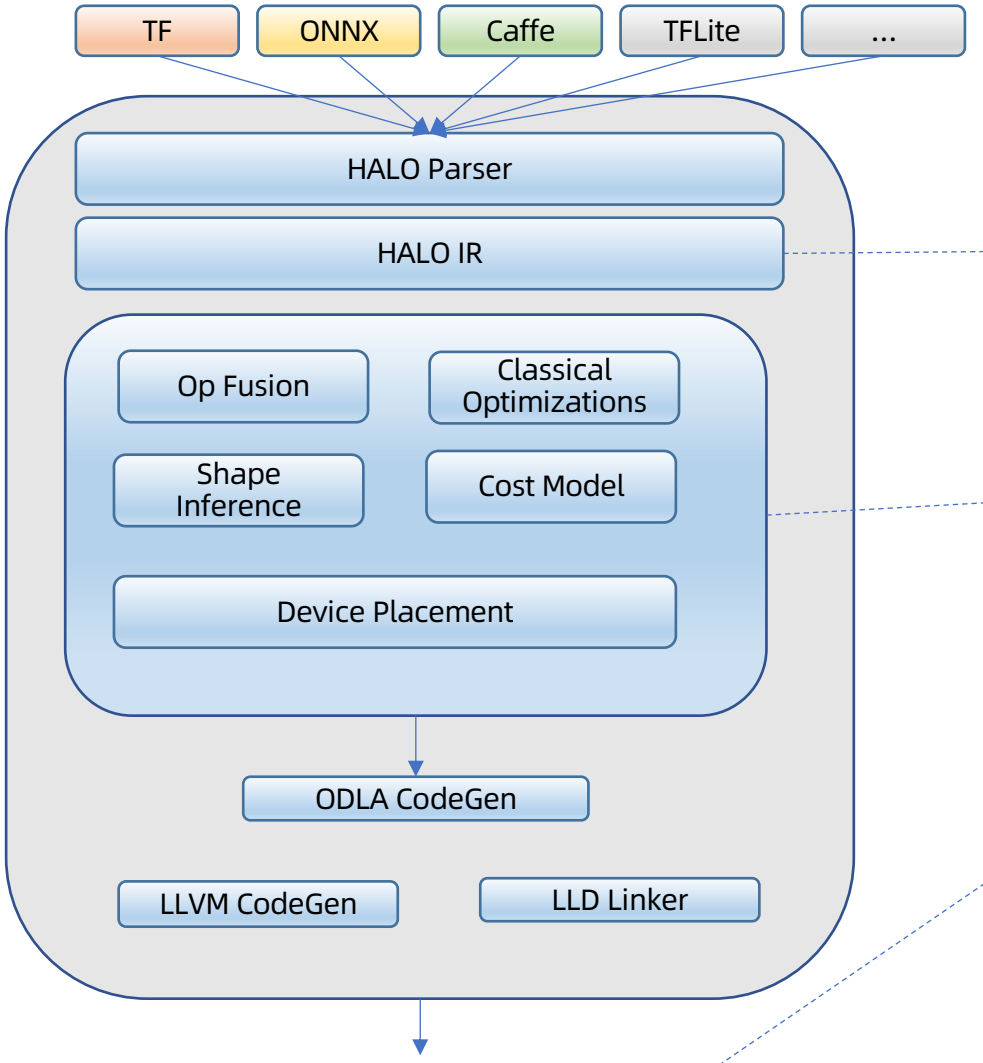
Alibaba Cloud

2. An Optimizing Compiler Framework
   - Heterogeneity Aware Lowering & Optimization (HALO)
     - Compiles AI algorithms & workflows → ODLA-based code
     - Optimizations
       - Classical compiler optimizations
       - AI optimizations
       - Heterogeneous devices support
         - Device placement
       - Parallelization & sharding

System Overview

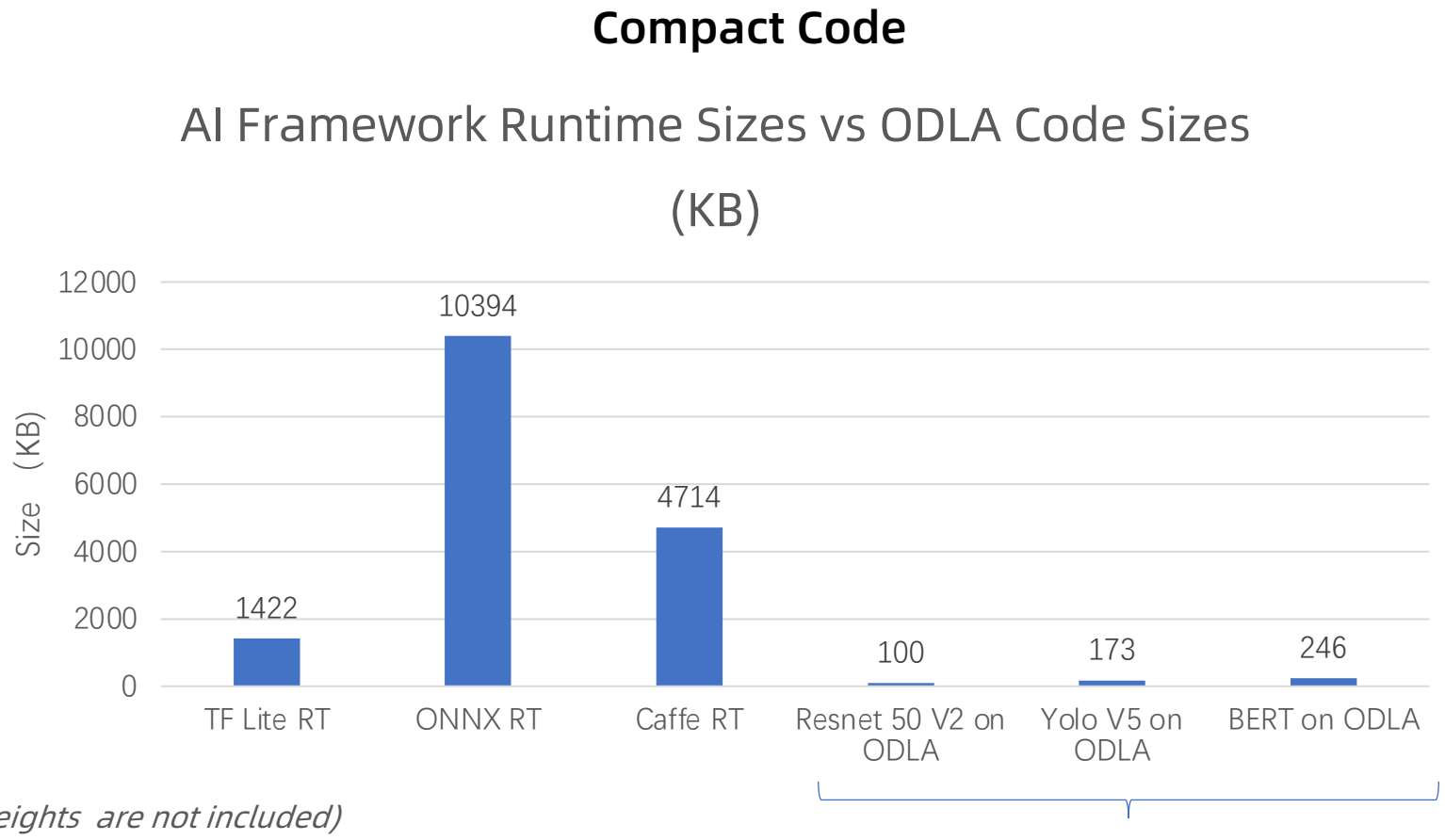


HALO Overview



- Multi-grained IR
  - Multi-level optimization
  - Easy to extend & customize
- Optimizations
  - Device-independent optimization
  - Device-dependent optimization
  - Partitioning & placement
- Generated code
  - Free of AI framework
  - Memory efficient
  - Low runtime overhead

Evaluations



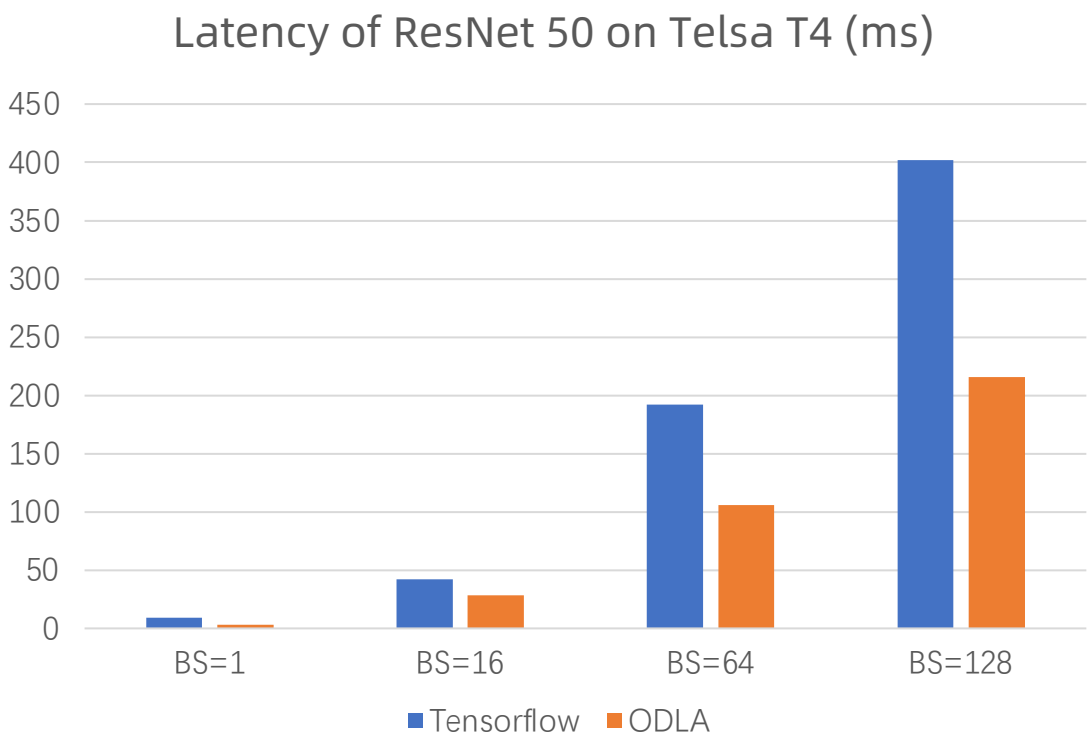
(weights are not included)

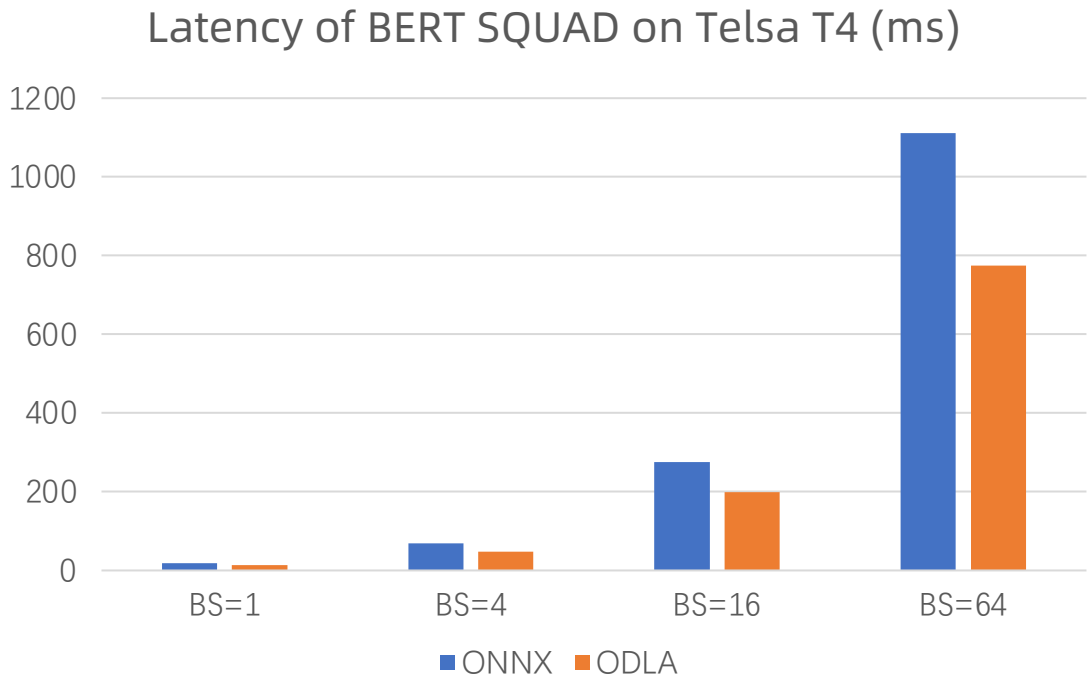
Best fit for small chiplets
(IoT, edge, etc.)

Evaluations

Alibaba Cloud

Efficient Execution





Case Study

- Smart Factory
  - Analog instrument reading, fire detection, security, …
  - Mixed AI algorithms
    - Caffe, TensorFlow
  - 3 HW platforms
    - Restrictions on storage & memory
  - Planned as a 6-month project for a team
  - With HALO/ODLA
    - For each HW, 1 engineer, ~2 weeks to implement ODLA RT lib
      - no prior experience on ODLA API
      - no need to access actual AI model
    - Finished in 2 months!
    - Recently migrated to new HW (~3 man-week)

From Wikipedia

Call for Actions

Alibaba Cloud

- Build unified software ecosystem
  - Open sourced @ *github.com/alibaba/heterogeneity-aware-lowering-and-optimization*)
    - Partnered with Intel, Graphcore, Qualcomm, Nvidia, Cambrian, …
    - Supports 10+ HW platforms
    - Please join us!
- New programming paradigm for heterogeneous chiplets?
  - HBM, D2D interface (BoW)
    - new programming model ? new language?
    - new distributed heterogeneous computing system?

Alibaba Cloud

Thanks