- 4 Framework
  - 4.1 Frameworks
    - 4.1.1 Mainstream Frameworks
      - Framework, Backends, Device, Model Family, Model Size, Highlight Features
    - 4.1.2 Instruction & Compilation
      - Instruction Sets (ISA): Computing Capability
      - Cross-platform Compatibility
    - 4.1.3 Model Export Format
      - Security, Performance, Compatibility, Ecosystem & Toolchains
  - 4.2 Kernels
    - 4.2.1 Quantization
      - 4.2.1.1 Quantization Strategies
      - 4.2.1.2 Customized Kernels
    - 4.2.2 Sparsification
      - 4.2.2.1 Sparse Storage as Dense
      - 4.2.2.2 Load as Sparse and Compute as Dense
  - 4.3 Graphs
    - 4.3.1 Atomic Operators Fusion
      - 4.3.1.1 What to Fuse
      - 4.3.1.2 How to Fuse
    - 4.3.2 Reuse and Sharing
      - 4.3.2.1 Chunk-Sharing
      - 4.3.2.2 CUDA Graph Optimizations
    - 4.3.3 Automatic Graph Generation
  - 4.4 Memory
    - 4.4.1 Memory Reuse
      - 4.4.1.1 Weight Sharing
      - 4.4.1.2 Activation Reuse
    - 4.4.2 Data Locality & Consecutive Access
      - 4.4.2.1 Tensor Reorder
      - 4.4.2.2 Memory Fragments Elimination
    - 4.4.3 Storage Hierarchy & Parameter Offloading
      - 4.4.3.1 Cost Model
      - 4.4.3.2 KV Cache Offloading
      - 4.4.3.3 Layer-Level Offloading
      - 4.4.3.4 MoE Offloading
  - 4.5 Pipeline
    - 4.5.1 Double Buffering
      - 4.5.1.1 Overlapping
      - 4.5.1.2 Minimizing Data Stall Time
    - 4.5.2 Multi-core Workload Balancing
  - 4.6 Collaboration
    - 4.6.1 Existing Heterogeneous Platforms
      - 4.6.1.1 Memory Architectures
      - 4.6.1.2 High Bandwidth Interconnection
      - 4.6.1.3 Inner Communication
    - 4.6.2 Heterogeneous Computing
      - 4.6.2.1 Drove by Data Movement Burden
      - 4.6.2.2 Drove by Arithmetic Intensity
      - 4.6.2.3 Case Study: Dataflow in Attention Mechanism
    - 4.6.3 Cloud-Edge Collboration
      - Draft-Verify; Easy-Hard; Prompt-Supplement; Abstract-Details