Colin Kirby
co201115@ucf.edu

HW #4

Paper #1 : *TensorFlow: A System for Large-Scale Machine Learning*

This lecture explores how deep learning has become a vital component in large-scale applications, highlighting the need for models that learn directly from massive datasets rather than relying on hand-crafted features. Jeff Dean demonstrates how AI techniques, especially those used at Google, can tackle everything from speech recognition to advanced language understanding in ways that traditional algorithms never could in the past. And by showcasing the enormous scope of modern data and the efficiency gained through those distributed systems, Dean paints a vivid picture of how neural networks continue to reshape our interactions with technology in the future.

Personally, I found it interesting to see how AI can dynamically adapt to various tasks like categorizing images or parsing complex text by capitalizing on raw data and minimal feature engineering. However, the talk also made me think about the resources needed to train these large models as smaller teams or institutions might find it challenging to replicate such scale. Still, Dean's insights into parallel computations and Google's practical successes hint at an exciting future where these techniques spread more widely, fueling innovative applications across more industries.

Paper #2 : *DéjàVu: KV-cache Streaming for Fast, Fault-tolerant Generative LLM Serving*

This talk introduces Deja, a system built to address key bottlenecks in large-scale language model (LLM) inference, including pipeline delays, wasted GPU memory, and limited failure handling. By splitting prompt processing from token generation and giving each its own machine, Deja cuts down on pipeline bubbles and boosts resource usage. It also optimizes how KV caches move between CPU and GPU to reduce unused GPU memory. Additionally, it tackles fault tolerance through key-value cache replication, preventing data loss and enabling smoother restarts. Thanks to these strategies, the presentation shows Deja can double throughput compared to baseline models, delivering a clear motivation for integrating this approach into state-of-the-art LLM services.

A standout strength is Deja's focus on flexible workload division and fail-safe strategies. Separating prompt tasks for parallel execution improves throughput, while cache replication enables quick recovery which is vital when downtime can be costly. This design could help future

LLM frameworks better manage their computational resources and avoid interruptions in production environments. However, relying on these specialized processes for prompt handling and token generation may complicate scaling in more varied settings, and maintaining replicated caches can introduce extra overhead. From my perspective, these limitations suggest that Deja's open-source nature is especially important as it allows broader testing and refinement, potentially resulting in a more accessible and resilient solution for large-scale LLM deployments.

Paper #3 : *Fast LLM Serving with vLLM and PagedAttention*

This presentation discusses how Bria 11 aims to overcome the high memory demands and slow serving capabilities faced by many language models (LMs). By introducing a Key-Value (KV) cache management strategy inspired by operating systems' paging concepts, the project significantly reduces memory fragmentation and improves throughput. The team emphasizes that current hardware setups often lead to underutilized memory, slowing down large-scale deployments like ChatGPT or Copilot. Through a combination of dynamic memory allocation and "paged attention," their solution allows bigger batch processing, ultimately delivering faster response times and better scalability. The motivation is if LMs can be served more efficiently, broader applications and cost savings become possible.

A notable advantage is the marked improvement in throughput with up to 24x over naive approaches which directly benefits real-world applications needing quick, large-scale inference. By making VLM open source, the project invites community feedback and collaboration, potentially accelerating further improvements. However, although the results are promising, adopting such a specialized memory management layer may require adaptation to existing infrastructure, posing complexity for smaller teams or legacy systems. In my opinion from this video, the ability to reduce memory waste while handling advanced techniques like beam search is a critical step toward more accessible and efficient AI tools. As the demand for AI continues to grow, the innovations in memory handling found here could pave the way for broader adoption and improved reliability in LM-based applications.

Paper #4 : *Paella: Low-latency Model Serving with Software-defined GPU Scheduling*

This presentation introduces Paella, a system designed to address latency challenges in machine learning serving by changing how GPU resources are shared. As models grow larger with some exceeding hundreds of billions of parameters, current GPU schedulers often struggle with inefficiencies like head of line blocking. Paella focuses on software defined scheduling, dispatching kernels only when they can run immediately. By combining detailed resource monitoring with scheduling techniques like shortest remaining processing time, this approach goes beyond traditional methods such as NVIDIA Triton. Experiments on a range of vision models running on an NVIDIA T4 GPU show that Paella can offer both faster throughput and lower latency, marking a step forward for large-scale AI services.

Despite these gains, adopting Paella may bring challenges for teams with existing setups or limited resources, since software-defined approaches can introduce complexity when balancing small tasks against large ones. Still, the results are convincing. By reducing idle time and synchronizing kernel dispatch more effectively, Paella can handle various high demand scenarios. In my view, it signals how machine learning infrastructures are evolving to be more flexible, suggesting a future where custom scheduling becomes standard practice for serving large models at scale.

Paper #5 : *Zeus: Understanding and Optimizing GPU Energy Consumption of DNN Training*

Zeus tackles the escalating energy cost of Deep Neural Network (DNN) training by jointly optimizing GPU power limits and batch sizes, as detailed in Section 3. This approach, called "energy-to-accuracy (ETA)," identifies configurations that reduce power consumption substantially—even up to 75%—while still maintaining performance objectives (Section 2). The authors demonstrate how conventional practices (e.g., setting maximal batch sizes or power limits) can waste energy, motivating Zeus's online exploration-exploitation procedure to automatically find an optimal trade-off between power draw and training time. By leveraging a Multi-Armed Bandit (MAB) formulation, Zeus measures cost outcomes repeatedly, quickly settling on efficient configurations without significant manual tuning (Section 4). This solution is especially relevant for production clusters where the same DNNs are retrained frequently, making online learning both economical and practical (Section 1).

A notable advantage is Zeus's capacity to adapt to shifting workloads ("data drift") with minimal overhead, offering strong real-world viability for any environment that retrains models on newly arriving data (Section 6.4). The main drawback is the assumption that many DNN jobs recur often and run on the same GPU type; less frequently repeated jobs or environments with mixed GPU hardware may need additional profiling steps (Section 7). Personally, I see Zeus's online nature as a promising step toward "set-and-forget" energy optimizations that can drive down operational costs for large-scale AI. However, I wonder how quickly it would converge with tasks or architectures that differ drastically from the training data used to form the bandit's initial priors.

Further research extending Zeus to more rapidly changing conditions might deepen its impact across broader AI deployments.

Paper #6 : *A Study of SSD Reliability in Large-Scale Enterprise Storage Deployments*

This paper presents a thorough examination of how NAND-based SSDs perform in large-scale enterprise storage environments, focusing on real-world telemetry data gathered from roughly 1.4 million drives. The authors discuss a variety of factors, such as drive age, capacity, firmware version, and flash technology, all of which can profoundly influence SSD reliability and failure rates (Sections 3.1 and 5.1). In contrast to earlier studies centered on data center deployments, this work specifically explores NetApp's enterprise systems, where high-end SSDs in RAID configurations must maintain strong reliability under different usage patterns. The motivation is clear with SSDs becoming standard in enterprise setups, understanding the nuances of field failures is crucial for designing better RAID policies and streamlining firmware updates (Introduction and Section 2).

Notably, the authors analyze how firmware updates can dramatically affect replacement rates, with some drives seeing order-of-magnitude improvements after switching away from early firmware (Section 5.5). They also highlight that higher-capacity SSDs face disproportionately severe issues, showing both more frequent and more catastrophic failures (Section 5.3). A key strength of the study is its granular look at correlation within RAID groups, where multiple drive failures often occur in tight succession—a scenario that can threaten data availability if not properly addressed (Section 6). On the other hand, the paper primarily examines NetApp deployments, and practitioners with mixed or smaller-scale storage setups might wonder whether these insights generalize. Personally, I find the observation that most drives use only a tiny fraction of their rated program-erase cycles interesting, suggesting that total wear out isn't the main issue (Section 5.1). As SSD capacities continue to climb, I'd be curious to see follow-up research on how differently QLC or even PLC technologies might behave in similar enterprise settings.

Paper #7 : *Tectonic-Shift: A Composite Storage Fabric for Large-Scale ML Training*

This paper presents a new storage system, Tectonic-Shift, which tackles the bandwidth-heavy and large-capacity demands of industrial ML training workloads. The authors note that Meta's existing HDD-based storage, called Tectonic, struggled to meet the continually growing I/O requirements of GPUs, leading to excessive power use from over-provisioning disk capacity. Tectonic-Shift addresses this gap by adding a flash-based "Shift" tier that caches hot data, significantly reducing hard disk reads and total power consumption while remaining transparent to users and requiring no changes to applications. The system is further optimized through caching policies that leverage both historical access patterns and scheduled job metadata to predict future data usage, ensuring high hit rates and minimized writes to flash. Such a composite fabric therefore outperforms single-tier HDD or flash deployments by balancing each medium's strengths (storage efficiency for HDDs, IOPS efficiency for flash) .

A key strength is the application-aware cache logic, which calculates when specific data will be reused by referencing ML job definitions rather than guessing from access logs alone. Because Tectonic-Shift's caching tier stays invisible to ML engineers, it's easily deployed at petabyte scale and adaptable to new training workloads. However, as the authors point out, evolving demands on flash endurance and new patterns in model training will require further enhancements to caching policies, data placement, and job scheduling. In my view, it's compelling how Tectonic-Shift merges domain insights with established caching frameworks to achieve a 29% power reduction for at-scale ML training. Looking ahead, co-designing job routing and data replication policies with Tectonic-Shift may open even bigger efficiency gains, especially as larger and more varied AI workloads continue to push storage fabrics to their limits.

Paper #8 : *Quant-LLM: Accelerating the Serving of Large Language Models via FP6-Centric Algorithm-System Co-Design on Modern GPUs*

Quant-LLM is a system designed to speed up large language model (LLM) inference by shrinking model weights down to six bits (called FP6). This format hits a good middle ground—smaller than the common 8-bit but more accurate than 4-bit. The main challenge is that GPUs usually work best with 8-bit or 16-bit formats, so using 6-bit requires special handling. To solve this, the authors created a system called TC-FPx, which works by combining different parts of the GPU to convert FP6 weights into a usable format (FP16) while the model runs. They also carefully arrange the 6-bit data ahead of time so it fits neatly into memory and flows efficiently through the system. By processing weights in small slices and sending them straight to the GPU's fast matrix math units, the system avoids delays and keeps everything running smoothly. This setup has already been added to DeepSpeed to make it easier to use in real LLM serving.

When tested on models like LLaMA-70B and OPT-30B, Quant-LLM showed strong improvements. It ran faster than both 8-bit and 16-bit versions, especially when handling small batches of data like in real-time text generation. In some cases, it cut execution time by more than half compared to FP16. It also kept accuracy high, unlike some 4-bit methods that can harm model quality. Overall, Quant-LLM gives developers a way to run large models faster and with less memory, without needing major changes to existing systems. And since it's open source, anyone can try it out and build on it.