Networking and Information Technology
Research and Development

***The government seeks individual input; attendees/participants may provide individual advice only*.**

Middleware and Grid Interagency Coordination (MAGIC) Meeting Minutes
May 4, 2022, 12-2 pm ET

Virtual

Participants

Aditya Akella (University of Texas)	Mallory Hinks (NCO)
Bogdan Mihaila (NSF)	Marcy Collinson (Oracle)
Cheryl Martin	Miron Livny (Wisconsin)
David Martin (ANL)	Olga Kuchar (ORNL)
Dhruva K Chakravorty (Texas A&M)	Rich Carlson (DOE/SC)
Donald Petravick (University of Illinois)	Robert Schreiber (Cerebras Systems)
Eric Lancon (BNL)	Sarp Oral (ORNL)
H Birali Runesha (University of Chicago)	Seung-Jong Park (NSF)
Hall Finkel (DOE/SC)	Sharon Broude Geva (University of Michigan)
Jeff Conklin (NCO)	Tevfik Kosar (NSF)
Juan Jenny Li (NSF/OAC)	Tom Gulbransen (NSF)
Keith Beattie (LBL)	Val Anantharaj (ORNL)
	Varun Chandola (NSF)

Introductions: This meeting was chaired by Rich Carlson (DOE/SC) and Tevfik Kosar (NSF)

A Case for Building AI-Native Systems

Aditya Akella, Computer Scientist, Professor and Regents Chair Professor at the University of Texas at Austin

- Aditya provided an overview of the presentation.
- Aditya discussed using AI in systems
    - Growing evidence that learned approaches >> manually crafted heuristics
    - Most are point solutions that target specific design components
    - Much excitement, but also trepidation in taking ideas to production
    - At least four stumbling blocks or blind spots:
        1. System layering: many approaches don’t result in end-to-end benefits
        2. Safety: learning often conflicts with correctness and performance guarantees
        3. Management: code + models –how do we debug? Troubleshoot? Manage? Update?
        4. Substrate: conflict with learning and system resources
- How do we bring the power of AI to the automation of production-grade computer systems and networks, vastly improving their performance, efficiency, and robustness, while preserving dependability and manageability?

    - Eschew point solutions and take a principled approach rethinking system design with AI as a first-class citizen, resulting in AI-Native systems
    - Fundamental AI advances that account for: the scale, heterogeneity, and coupled nature of learned system components; provably correctness, performance; and end-to-end understanding
    - Aditya stated this presentation will ask the fundamental questions, discuss at a high level some promising approaches, and discuss some ways or opportunities for collaboration.
    - He provided an example of an AI-Native system, describing a tiered approach with data/database solutions in the cloud, Application and network transport control on the edge connecting to users via 5G and beyond cellular configurations. The example system would provide Automatic choice of execution plan in retrieving data, automatic rate adaptation at the network transport layer, and automatic selection of hundreds of attributes at the interface layer.
- A Holistic Approach to Layering, Interactions, Operationality
    - Challenge - Each system is deeply layered (tens of components), co-dependent on other systems (tens of them).
    - Millions of interacting requests.
    - AI-Nativeness calls for: Holistic at-scale AI-driven optimization across boundaries of many diverse systems, networks
    - Holistic approach to operations, ensuring correctness, dependability, and understandability/manageability
- Where AI-Nativeness can Make a Difference
    - Service providers networks -ESnet, ATT
    - Cloud services, infrastructure -CloudLab, Micro Focus, Microsoft, VMware
    - Cyber-infrastructure
    - – HTCondor
    - And many more... Mobile, edge settings; Robot systems; Real-time disaster response
- Fundamental Issues
    - AI-Native Systems:AI, as a first-class citizen, automatically selects system designs and configurations that optimize cross-layer cross-system performance while ensuring operationality.
    - Effectively using AI in production computer systems and networks: maximizing flexibility to improve performance while constraining designs to improve operations
    - Explore cross-system AI-based optimizations
    - Select optimal run-time data and resources
    - Ensure average or worst-case behavior meets objectives
    - Identify designs that are easy to understand and manage
- End-to-end Example: Databases
    - Challenges:
        - Hand-tuned components: Fail to perform under dynamic workloads or learn from past optimization actions
        - Heterogenous agents: require coordination for end-to-end performance benefits
- AI-Native Performance Optimization

  - What we need: Data-driven, online optimization of large-scale multi-agent systems; learning to cooperate in dynamic settings with heterogenous components operating at different time-scales
    - A: Design Time
      - Cross-component API exploration
      - Low-cost, high-quality feature discovery
    - B: Run Time
      - Dynamic allocation of system resources to learning
      - Leveraging domain knowledge
      - Simulation of runtime decisions
    - C: Coordination
      - Async learning and credit assignment across timescales
      - Heterogeneous agents with limited information sharing
- Cross-Layer Coordination Using MARL
  - Using AI as a design tool for neural branch predictors
  - End-to-end Example: Databases
    - Challenge
      - Learning does not guarantee correctness: E.g., hard constraints on equivalence
      - ○Learning does not guarantee performance: E.g., when data distribution shifts
- Guardrails for AI-Native Systems
  - Goal: Install guardrails in AI-Native computing: operationalization and deployability in real-world arbitrary environmental conditions.
  - What we need: AI systems with correctness (e.g., non-differentiable constraints), performance dependability in multi-agent settings; distributed safety.
  - A: Guardrails for Correctness
    - Guardrail specification language - safety and liveness properties
    - Enforcing hard, non-differentiable constraints in online and offline learning - safe exploration in RL
  - B: Guardrails for Performance
    - Establishing, maintaining high performance in online learning
    - Detect, mitigate distribution shifts in non-iid, non-stationary, multi-agent context
    - Semantically robust learning for performance
- Learning with Safety Constraints: Differentiable Symbolic Execution
  - Estimate volumes of individual control paths
  - Approximate the program's worst-case safety loss by an integral over paths
  - Compute the gradients of this integral using the classic REINFORCE estimator, balancing two goals:
    - Ensure program trajectories are safer
    - Learn parameters of conditionals so that program avoids unsafe paths.
  - Challenges
    - Management: many interacting layered systems (AI and non-AI components); CI/CD issues; Security
    - Debugging: adoption of learned components hindered by lack of model explainability

- Managing AI-Native Systems
    - Goal: Foresee and address manageability issues in AI-Native systems
    - Innovations: Effective cross-component management, explainability and security across AI and non-AI components.
    - A: Debugging, SE Methods
        - MLOps+ DevOps
        - Scalable root-cause analysis (RCA)
        - Cross-component observability, explanations building on DSE
        - Human-guided patching
    - B: Security and Privacy
        - AI-Native threat model (experiment-guided, red teaming)
        - Guaranteed composability
        - Private training and inference with operational utility
- Manageability and Resilience in Shared CI
    - Goal: Troubleshooting and debugging in large-scale workloads on shared distributed resources
    - Challenges: Planet-scale shared infrastructure; large, complex apps
- Next Steps for AI-Native Systems
    - Maximizing flexibility
        - Explore cross-system AI-based optimizations
        - Select optimal run-time data and resources
    - Constraining design
        - Ensure average or worst-case behavior meets objectives
        - Identify designs that are easy to understand and manage
    - We urgently need a firm understanding of AI-Native foundations, and concrete realizations of AI-Native concepts to guide the community and show promise.
        - -Working with ESNeton an AI-Native network substrate, and with CloudLabon AI Native experiment management platforms
        - -Collaborations and workshops with industry
        - -New courses that introduce AI-Native system design, and manageability

Questions:

- Tevfik wanted to know about the personal challenges of bringing AI Native to the different sets of applications.
    - Aditya answered that when you do learning of routing and traffic engineering in isolation, you get good performance relative to handcrafted heuristics. But when you have applications that are running across the routed network that are making learning-based decisions in terms of where to move their workloads (congestion control) the network makes certain assumptions and learns a new set of routes, then the application makes decisions that throws this assumption completely out of the picture. Aditya stated they are trying to collaborate, to focus on the workload decisions at the different layers and at different time scales.
- Dhruva wanted to know with all the interaction and latency how does adding the security layer affect what was presented.
    - Aditya talked about a particular situation focusing on autonomous driving that required safety factors and if for a moment the processor stalled it might influence a decision

whether to stop the car or make a right turn. As for security Aditya requested they get together and have a discussion. He stated that these learning systems are vulnerable to all of the security threats are experienced by regular systems.

Wafer-scale AI computing

Rob Schreiber, Cerebras Systems

- Overview
  - AI has transformative potential in health, science, engineering, and security.
  - AI is compute-constrained today. Experiments (training a network) can take weeks or months.
  - Long training time limits researchers' ability to test new ideas.
  - A much faster AI platform enables interactive science, cuts time to insight, leads to new discoveries.
- Training needs faster computers
  - 1800x more compute in just 2 years
  - Future, multi-trillion parameter models
  - Distributed training does not scale well
- High Performance Computing (HPC) and Artificial Intelligence (AI)
  - AI: image processing, text, and sequence modeling
  - HPC: physics-based modeling and simulation, chemistry, signal processing
  - Converged AI+HPC: surrogate AI physics models, AI-augmented HPC
  - Common demands:
    - sparse data structures, graphs
    - strong scaling, time to solution
    - High communication bandwidth
    - High memory bandwidth
  - Where and How to Break Through?
- Cerebras Wafer-Scale Engine (WSE-2)
  - Cluster-scale performance in a single chip
  - The largest chip in the world
    - 850,000 cores optimized for sparse linear algebra
    - 46,225 mm2silicon
    - 2.6 trillion transistors
    - 40 Gigabytes of on-chip memory
    - 20 PByte/s memory bandwidth
    - 220 Pbit/s fabric bandwidth
    - 7nm process technology
- Cerebras CS-2 System
  - The world's most powerful AI computer
  - Deploy easily into standard racks
  - Standards-based integration
  - Available on-prem or remote / cloud
- Cerebras software and programming model
  - Program a cluster-scale resource with the ease of a single node
- Cerebras advantages for AI and HPC
  - Architecture tuned for AI and HPC
    - Massive, fine-grained parallel engine

        - Programmable core optimized for sparse, tensor-based linear algebra
        - High bandwidth, low latency, local SRAM memory, on chip
        - High bandwidth, low latency, 2D interconnect mesh
        - Programmable for AI easily with standard ML frameworks
        - Customizable for HPC and other applications with lower-level SDK
- Examples of real-world applications:
    - Financial Services: Using Domain-Specific Datasets to Improve ML Model Accuracy
        - Objective: Improve BERTLARGE model accuracy by training from scratch using domain-specific datasets for Financial Services applications
        - Challenge: Training from scratch was intractable using conventional hardware, making experimentation impractical
        - Outcome: CS-2 reduced training time 15X, enabling demonstration of dramatic improvement in model prediction confidence using Thomson Reuters TRC2 dataset
    - Novel AI epigenomic model from GlaxoSmithKline
        - Objective: Accelerate genetic validation of drug targets using novel technique that includes epigenomic data in NLP models, rather than genome-only models
        - Challenge: Training this complex model with massive datasets would take several weeks on a 16-GPU cluster, making rapid experimentation impractical
        - Outcome: ~10X training speedup empowered researchers to experiment with epigenomic data and demonstrate superior results to DNA-only datasets
    - Accelerated energy research at Total Energies
        - Objective: Enable order-of-magnitude speedups on a wide range of simulations: batteries, biofuels, wind flows, drillings, and CO2 storage
        - Challenge: Participate in Total study to evaluate hardware architectures, using finite difference seismic modelling code as a benchmark
        - Outcome: Cerebras CS-2 system outperformed a A100 AI GPU by >200X using code written in the Cerebras Software Language (CSL). System now installed and running at customer facility in Houston, TX
    - Toward real-time computational fluid dynamics: NETL
        - CS-1 used to accelerate physics-based CFD
        - CS-1 system solves sparse linear equations 200x faster than Joule 2.0 supercomputer1
        - Sparse GEMM performance enabled by massive memory bandwidth
    - AI-augmented MD for CoVID-19 research at ANL
        - Task: Steer numerical simulations by learning behavior of previous runs
        - Challenge: CVAE is quadratic in time and space complexity and can be prohibitive to train.
        - Outcome: Support out of box throughput comparable with 100 GPUs
- Conclusions
    - Convergence of compute demands across AI and HPC
    - Exponential growth in datasets, model sizes
    - Wafer-scale compute: a radical and now successful hardware innovation for AI + HPC
    - The Cerebras CS-2 delivers
        - Orders of magnitude greater performance
        - Time to solution from days-weeks to minutes-hours
        - Simple, standards-based programming model

- Recently launched software supporting more PyTorch and billion-parameter models
- But this is just the beginning. Near-term future:
    - Support for 10s-100s billion parameter models, multi-CS2 clusters

Questions

- Rich Carlson wanted to know what specific challenges that had to be overcome to get to present state.
    - Rob stated that there are problems in the wafers due to manufacturing or breaks etc. The build in redundancy in the processors on the wafers, extra links, etc. So he stated they diagnose the ones that are working after manufacturing, and set up a satisfiability problem, what is the biggest working machine they can create by routing around the bad processors. Other problems include bringing in enough power to the wafer and removing the heat, though the heat density is probably less than the common gpu/cpu.
- Bodgan Mihaila asked how many of the processors and links (on the wafer) actually work.
    - Rob stated that it's about 80% but implied that the wafers are made with more than enough processors. He stated they are over the 500K range and approaching the 700K to 800K range.
- Bodgan also asked how many layers do the wafers have.
    - Rob stated there are a lot of metal layers but a lot of them are for the redistribution of power. The communication is mesh, all short wires.
- Seung-Jong Park asked if he could use multiple systems together.
    - Rob stated that they were right at the threshold for multi-wafer systems for training one network. They are currently doing this in their labs. They also go the other way and train small networks by making multiple copies of the network and train it in a data parallel way like you would split across multiple GPUs. They don't timeshare the processor, but they do space share the wafer. No one is doing that now, but it is conceivable that everybody gets a quarter of a wafer for some period until a big job comes and takes the whole wafer.
    - Seung-Jong stated that it seemed that there was only 40 gigabits of SRAM per GB.
    - Rob responded saying it was all SRAM on the Wafer, and that there was no attached DRAM or DRAM on the wafer. In the future they will be making some changes to reduce IO bottlenecks.
- Dhruva. asked if Rob could provide any insight into the future of the software to be used with this.
    - Rob provided some details about one interesting case at Livermore. There is a middle level SDK that they have put out there and are getting good feedback.

Roundtable

- Dhruva shared a number of initiatives he is participating in from hosting sessions at different conferences to working with high school students. He will be sending these to Mallory to share with this group.

Next Meeting

June 1st (12 pm ET)