

Teme Orientative pentru Proiecte

la disciplina *Tehnologii Cloud Computing pentru Machine Learning*

I. Sisteme Distribuite, Consens și Arhitecturi Cloud

1. **Arhitectura de Consens Hibrid pentru LLM-uri Distribuite (C1):** Proiectarea și implementarea unui protocol de consens dinamic care să optimizeze echilibrul dintre Consistență (C) și Toleranță la Partiționare (P) (**Teorema CAP**) în medii de antrenare a modelelor LLM la scară exascale, utilizând tehnologii de tip **Service Fabric** și **RDMA**(**Remote Direct Memory Access**).
2. **Modelarea Toleranței la Eșec în Data Lake-uri Geografice (C3):** Dezvoltarea unui cadru de simulare și evaluare a rezilienței arhitecturilor de stocare de tip **Delta Lake** sub scenarii de eșec geografic și partiționare extinsă, analizând impactul asupra **Controlului Concurenței Optimsite (OCC)**.
3. **Optimizarea Workload-Aware Orchestration în Cloud-ul Hibrid (C2, C4):** Crearea unui sistem de orchestrare bazat pe **Kubernetes/Service Fabric** care să aloce dinamic resursele de calcul (**IaaS**) pe baza cerințelor semantice ale sarcinilor de lucru (de exemplu, diferențierea între joburile **Spark** și serviciile **Actor**), integrând concepte de *Confidential Computing*.
4. **Consistență Edge-to-Cloud (C2, C3):** Propunerea unui nou model de consistență la nivel de sistem care să gestioneze replicarea și sincronizarea datelor între mediile de **Edge Computing** și serviciile de stocare centralizate (**Cosmos DB Eventual/Strong mode**), minimizând latența.

II. Învățare Automată (ML) Distribuită și MLOps

5. **Un Framework de *Data Provenance*¹ Bazat pe Delta Lake și MLflow (C3, C4):** Dezvoltarea unui sistem automatizat de urmărire a trasabilității (provenance) datelor și modelelor ML, utilizând registrul tranzacțional **Delta Lake** și integrând **MLflow** pentru a garanta reproductibilitatea și consistența semantică a experimentelor distribuite.
6. **Scalabilitatea Antrenării LLM prin Optimizarea Rețelei (C1):** Cercetarea algoritmilor de scheduling și a topologiilor de rețea optimizate pentru **RDMA**, cu scopul de a reduce *overhead-ul* de sincronizare în antrenarea distribuită a LLM-urilor (utilizând **PyTorch Operator** pe AKS).
7. **Sisteme de Fișiere pentru HPC/AI (C1):** Analiza performanței **Azure Managed Lustre (AMLFS)** versus alte soluții de stocare distribuite pentru scenarii de citire/scriere

¹ Istoricul complet al datelor la nivel Enterprise

intensivă, specifice antrenării LLM-urilor, și propunerea de îmbunătățiri la nivel de arhitectură I/O.

III. Interogare Semantică și LLM-uri Avansate (RAG/Ontologii)

8. **RAG Agentic Fundamentat pe Ontologii (C6):** Proiectarea unui cadru de agenți autonomi bazat pe **Azure Semantic Kernel/LangChain**, care utilizează **Ontologii** ca mecanism de **Grounding** și validare a faptelor extrase, îmbunătățind precizia și reducând halucinațiile în **RAG (Retrieval-Augmented Generation)**.
9. **Generarea de Strat Semantic pentru Data Warehouse (C5, C6):** O metodologie pentru generarea automată și întreținerea dinamică a unui strat semantic (sub forma unei **Ontologii**) peste baze de date relaționale (**Azure SQL Database**), cu scopul de a facilita interogarea precisă în limbaj natural (**Text2SQL**) prin **Vector Search**.
10. **Hibrid RAG și Fine-Tuning (C5, C6):** Dezvoltarea și evaluarea unei metode hibride care combină *Fine-Tuning*-ul unui LLM pe date operaționale (pentru stil și aliniere) cu un pipeline **RAG** bazat pe date relaționale (pentru cunoștințe factuale), îmbunătățind atât coerența, cât și precizia.

IV. Baze de Date și Stocare Distribuită

11. **Protocol OCC Bazat pe Grafuri de Cunoștințe pentru Delta Lake (C3, C6):** Cercetarea unui nou protocol de **Optimistic Concurrency Control** (OCC) pentru arhivele **Delta Lake**, unde validarea tranzacțiilor este asistată semantic de **Ontologii** (asigurând consistența logică a schemelor pe lângă cea fizică).
12. **Model de Cost Dinamic pentru BDaaS (C5):** Propunerea unui model predictiv care să orienteze migrarea automată a workload-urilor de baze de date între diferitele niveluri de serviciu (**PaaS** - Azure SQL, **IaaS** - VM, **Serverless**) pe baza metricilor de elasticitate și cost-eficiență.
13. **Vector Search Avansat cu Structură Ontologică (C5, C6):** Crearea de algoritmi de indexare și recuperare vectorială în pipeline-urile **RAG**, care să utilizeze ierarhia și relațiile din **Ontologii** pentru a îmbunătăți relevanța semantică și *grounding*-ul fragmentelor de text extrase.

V. Concepte Filosofice și Arhitecturale

15. **Simularea Sensului în Arhitectura Transformer (C6):** O analiză computațională profundă a modului în care **Arhitectura Transformer** (prin intermediul *embeddings*-

urilor) reușește să simuleze sensul ca utilizare contextuală, comparând rezultatele cu viziunea pragmatică a lui **Wittgenstein**.

16. **Securitate Semantică în Data Lake-uri (C3, C6):** Dezvoltarea unui model de control al accesului bazat pe politici (Policy-Driven Access Control) pentru **Data Lake Gen2**, unde permisiunile sunt definite nu doar pe baza structurii de fișiere, ci și pe **Ontologia** datelor (ex: măști de date bazate pe concepte).

VI. Proiecte de Convergență și Hibridizare

17. **Convergență IaaS/PaaS/Serverless pentru Aplicații AI (C2, C5):** Un studiu de caz privind implementarea și performanța unei aplicații complete de **RAG** (incluzând indexarea **Azure AI Search** și interogarea **Azure SQL**) într-o arhitectură care combină în mod optim **IaaS, PaaS** și **Serverless**.
18. **Scheduler Spark Sensibil la Consistență (C3, C4):** Dezvoltarea unui scheduler alternativ pentru **Apache Spark** care să integreze cerințele de consistență ale protocolului **Delta Lake (OCC)** direct în planificarea joburilor, minimizând tranzacțiile eșuate în scenarii de scriere concurrentă.
19. **Metodologie CAG (Context-Augmented Generation) vs. RAG (C6):** O cercetare comparativă extinsă între **RAG** și **CAG (Context-Augmented Generation)** pe baza unor metriki de precizie, latență și consum de resurse, aplicată pe seturi de date din domenii tehnice.
20. **Arhitecturi pentru Confidential Computing și LLM-uri (C1, C2):** Proiectarea unei arhitecturi sigure pentru antrenarea și inferența LLM-urilor care utilizează **Confidential Computing** la nivel de **IaaS** pentru a proteja atât modelul, cât și datele de antrenare în mediul multi-tenant al cloud-ului public.