

**PLANIFICAREA LUCRĂRILOR DE LABORATOR LA DISCIPLINA
“ARHITECTURI AVANSATE”(C) / “SISTEME CU PROCESĂRI
PARALEL-SPECULATIVE”(TI)**

L1. INTRODUCERE. Prezentarea metodologiei de simulare și optimizare bazat pe SimpleScalar Tool Set 3. Prezentare generală a conținutului laboratorului axat pe dezvoltarea aplicativă de arhitecturi urmată de simulări software complexe în sistemul de operare Linux (Fedora / Ubuntu). Descriere succintă a celor 8 simulatoare de tip *execution driven*: simulatoare care evidențiază gradele de localitate centrate pe diverse resurse – instrucțiuni, regiștri, predictoare de valori a resurselor (de tip Last Value, Incremental, Contextual, Hibrid), predictoare și metapredictoare centrate pe regiștrii, predictoare de salturi indirecte. Grefarea tehnicilor de reutilizare dinamică și predicție a valorilor pe arhitecturi *SMT*, simularea cache-urilor într-un sistem multicore omogen în Linux. Compilarea și execuția de programe pe arhitectura multicore eterogenă IBM-CELL. [Repartizare teme de proiect](#).

L2. INIȚIERE ÎN SISTEMUL DE OPERARE LINUX. DEZVOLTAREA DE SIMULATOARE SUB MEDIUL SIMPLESCALAR UTILIZÂND BENCHMARK-URILE SPEC.

Aplicarea unor comenzi și utilitare Linux. Analiza rezolvării unei probleme prin compilarea, depanarea și execuția programului C/C++. Ce este "SimpleScalar Tool Set" ? Avantajele/dezavantajele utilizării "SimpleScalar Tool Set". Arhitectura SimpleScalar. Benchmark-urile SPEC'95. Simulatoarele setului de instrumente SimpleScalar 3.0. Ghid de utilizare. Extensii. Evaluarea performanțelor.

L3. EXTINDEREA MEDIULUI SIMPLESCALAR CU MODULUL DE MĂSURARE A GRADELOR DE LOCALITATE (REGIȘTRI / INSTRUCȚIUNI). PREDICȚIA VALORILOR FOLOSIND SCHEMA DE TIP LAST-VALUE.

L4. IMPLEMENTAREA UNOR TEHNICI DE PREDICȚIE DINAMICĂ A VALORILOR. DETERMINAREA ACURATEȚII PREDICȚIEI FOLOSIND PREDICTOARE INCREMENTALE, CONTEXTUALE ȘI HIBRIDE CU PRIORITIZARE STATICĂ. LIMITE. DETERMINAREA CÂȘTIGULUI DE PERFORMANȚĂ (SPEED-UP) OBTINUT PRIN IMPLEMENTAREA ACESTOR STRUCTURI DE PREDICȚIE.

L5. DETECTIA ȘI IZOLAREA SALTURILOR DIFICIL DE PREZIS INTR-UN ANUMIT CONTEXT. PREDICȚIA ACESTORA PRIN INTERMEDIUL UNOR SCHEME AVANSATE (NEURALE ȘI HIBRIDE). Modelare și implementare sub platforma industrială standardizată CBP – campionatul mondial de predicția salturilor. Simulare trace-driven.



L6. DEPĂȘIREA UNOR BARIERE ARHITECTURALE ALE PARADIGMEI ACTUALE A PROCESOARELOR SUPERSICALARE. Dezvoltarea unei arhitecturi supersicalare îmbogățită cu un mecanism complex de anticipare selectivă a valorilor instrucțiunilor cu latență ridicată de execuție, inclusiv pentru a reduce impactul negativ al salturilor nepolarizate asupra performanței de procesare. **SCHEMĂ DE REUTILIZARE** pentru instrucțiunile de înmulțire și împărțire (DIV/MUL) respectiv un **PREDICTOR DE VALORI (LVP)** pentru instrucțiunile *Load* critice. **CUANTIFICAREA IMPACTULUI MECANISMULUI DE ANTICIPARE SELECTIVĂ A VALORILOR INSTRUCȚIUNILOR CU LATENȚĂ RIDICATĂ ÎNTR-O ARHITECTURĂ DE TIP MULTITHREADING SIMULTAN (SMT)**, care implică multiplicări ale *buffer*-elor de reutilizare și ale predictoarelor de valori, pentru fiecare fir în parte.

L7. a) INFLUENȚA UNOR CONCEPTE DE PROGRAMARE PROCEDURALĂ / OBIECTUALĂ ASUPRA GENERĂRII SALTURILOR INDIRECTE PREZENTE LA NIVEL ARHITECTURAL.

b) **NECESITATEA PREDICȚIEI SALTURILOR / APELURILOR INDIRECTE. SOLUȚII – PREDICTOR DE TIP TARGETCACHE, PREDICTOR HIBRID CU SELECȚIE BAZATĂ PE ARITATE. SIMULATORUL JINDIRSIM. SIMULAREA BENCHMARK-URILOR SPEC2000.**

L8. PREDICȚIA VALORILOR CENTRATĂ PE REGIȘTRI PROCESORULUI. DETERMINAREA CÂȘTIGULUI ÎN ACURATEȚE OBTINUT PE ANUMIȚI REGIȘTRI FOLOSIND STRUCTURI DE METAPREDICȚIE CU SELECȚIE BAZATĂ PE COMPORTAMENTUL DINAMIC AL PREDICTOARELOR COMPONENTE (LAST VALUE / INCREMENTAL / CONTEXTUAL).

L9. CACHE-URI ÎN SISTEME MULTIPROCESOR AFLATE PE UN SINGUR CIP (procesoare multicore). INSTALAREA și CONFIGURAREA SIMULATORULUI MULTI2SIM PE S.O. LINUX UBUNTU 8.04 / FEDORA core 9. Familiarizarea cu comenzi / tools-uri în Linux. Obținerea simulatorului. Instalarea librărilor necesare. Simularea unei arhitecturi SMT / multicore pe benchmark-uri paralele – SPLASH2, PARSEC folosind simulatorul Multi2Sim (cu suport pentru fire de execuție). Să se reprezinte grafic variația performanței de procesare în funcție de parametrii de intrare (numărul de nuclee, numărul de linii din cache (L1 / L2), gradul de asociativitate și topologia cache-ului de date).

L10. COMPILAREA ȘI EXECUȚIA APLICAȚIILOR PE ARHITECTURA MULTICORE ETEROGENĂ IBM-CELL. Familiarizarea cu mediul CELL SDK: tools-uri de compilare / dezvoltare. Tipuri diferite de executabile: standalone PPU program, standalone SPU program, (a)synchronous SPU thread called by PPU.

L11 ÷ L13. PROIECT – ANALIZĂ ȘI IMPLEMENTARE.

L14. EVALUARE PROIECT.

