

# Паралелно програмиране

доц. д-р А. Пенев

Анотация: Целта на учебната дисциплина е придобиване на теоретични знания и практически умения в областта на паралелното програмиране. Това включва – съвременни паралелни изчислителни архитектури (CPU, GPGPU и др.), йерархия на паметта и архитектури на паметта на паралелните компютри (споделена, разпределена и разпределена-споделена памет; UMA, NUMA, COMA и др.), различни видове паралелизъм (ILP, Data parallelism, SMT, SpMT, Preemptive и Cooperative многозадачност, нишки, задачи и др.), и др. Разглеждат се някои важни теоретични аспекти на паралелните алгоритми, техният анализ, даннови и контролни зависимости, възможни проблеми (мъртва хватка, жива хватка, трудна скалируемост, глад за ресурси, съперничество и др.). Разгледани са също и подходи за дизайн и анализ на паралелни програми (Шаблони, Fork-Join, Map-Reduce, Неблокиращи алгоритми и структури данни и др.). Лекциите включват множество примери и дават някои основни сведения за езиците и библиотеките (API) за паралелно програмиране (например C++11 STL, C# TPL, PLINQ, Java Streams, Posix Threads, Boost.Thread, TBB, CUDA, OpenCL, OpenMP, MPI/MPI-2, C++ AMP, OpenHMPP, OpenACC и др.)

Упражненията са проектно базирани и дават възможност на студентите да се запознаят с практическите проблеми и начините им на решаване чрез средствата на паралелното програмиране.

## Лекции:

### **1. Въведение в паралелното програмиране. (2 часа)**

Пример. Кратка история. Базови понятия и концепции. Паралелни архитектури. Модели за паралелно програмиране. Възможни проблеми. Пример – критична секция. Практически проект.

### **2. Съвременни паралелни архитектури. Класификация на Флин (Flynn). CPU, GPGPU, HSA, Високо производителни компютри (HPC). (2 часа)**

Класификация на Flynn (SISD SIMD SIMT MISD MIMD). Скаларни/Pipelined и Суперскаларни процесори. SIMD инструкции. Dataflow architecture. Vector processor. Multiprocessor (symmetric/asymmetric). CPU/GPGPU. HSA. HPC.

### **3. Памет. Йерархия на паметта. Архитектури на паметта на паралелните компютри. (2 часа)**

Памет. Кеш. Архитектури (shared, distributed, distributed shared, UMA, NUMA, COMA), Shared-Nothing Architecture и др. Massively parallel computer (GRID Системи, Компютърни Клъстъри, Силно паралелни процесорни масиви).

### **4. Видове паралелизъм. Векторизация. Задачи – фибри, нишки, процеси. Видове многозадачност. (2 часа)**

Видове паралелизъм: Bit Level; Instruction (ILP); Task; Data; Memory. Векторизация. Векторизация на цикли. Задачи – фибри, нишки, процеси. Видове многозадачност: Temporal; Simultaneous (SMT); Speculative (SpMT); Preemptive; Cooperative; Clustered Multi-Thread (CMT).

### **5. Теоретични аспекти на паралелните алгоритми. Анализ на паралелни алгоритми. Зависимости на данните, структурата и контрола (Data dependency, Data, Structural and Control Hazards). (2 часа)**

PRAM модел. PEM модел. Разпаралеляване. Анализ на паралелни алгоритми. Критичен път. Закони на Amdahl. Закон на Gustafson. Метрики на Karp–Flatt. Забавяне и Ускорение. Коефициенти и метрики. Granularity...

**6. Класически алгоритми и проблеми. Алгоритъм на Декер. Проблеми при паралелните алгоритми: Мъртва хватка, жива хватка, трудна скалируемост, глад за ресурси, съперничество и др. Producer-Consumer. (2 часа)**

Задача за „Вечерящите философи“ и др. Алгоритъм на Декер. Задача за „Тримата пушачи“. Задача за „Спящият бръснар“. Deadlock, Livelock, Parallel slowdown, Race condition, Software lockout, Scalability, Starvation, Convoying, Contention; Deterministic algorithms; Embarrassingly parallel; Producer-Consumer.

**7. Модели за паралелно програмиране. Координация в паралелните алгоритми. (2 часа)**

Модели за паралелно програмиране: Shared Memory model; Threads model; Message Passing model; Implicit interaction model; Data Parallel model. Видове координация/синхронизация в паралелните алгоритми: Barrier; Locks; Semaphores; Mutexes, ...

**8. Дизайн на паралелни програми. (2 часа)**

Проблем и решения. Декомпозиция на данни и алгоритми. Видове комуникация между подзадачите. В/И – проблеми и решения. Fork-Join. Map-Reduce. Hotspots и Bottlenecks. Не блокиращи алгоритми и структури данни. Zero-Copy. Read-Copy-Update (RCU/COW). Транзакционна памет.

**9. Езици и Библиотеки (API) за паралелно програмиране. (2 часа)**

Езици за ПП. Примери. Библиотеки: C++11 STL (Futures, Promises, Threads, ...); C# (Threads, TPL, Tasks, Futures, PLINQ, async, await, yield, ...); Java (Threads, Locks, Atomics, Futures, Streams, ...); OpenMP (Fork-Join, ...); MPI/MPI-2 (Комуникация, синхронизация, паралелен В/И, редукции, ...); POSIX Threads, Boost.Thread, TBB; CUDA (Примери, PyCUDA, ...); OpenCL (); OpenHMPP, OpenACC, C++ AMP (Кодолети, кернели, #pragma, GPU, ...).

**10. Бъдеще на паралелните архитектури и програмиране. (2 часа)**

GPGPU, TPU, FPGA, ASIC, NPU, QPU – Квантов паралелизъм и др.

**Забележка: Материалът в лекцията няма да влиза във въпросите за теста.**

Упражнения:

**1. Избор на тема на проект. Изисквания. (2 часа)**

**2. Езици и библиотеки за паралелно програмиране. (2 часа)**

**3. Работа по проект. (2 часа)**

**4. Памет. Даннови зависимости. (2 часа)**

**5. Работа по проект. (2 часа)**

**6. Видове паралелизъм. Синхронизация и др. (2 часа)**

**7. Класически паралелни алгоритми. (2 часа)**

**8. Работа по проект. (2 часа)**

**9. Програмиране на високо паралелни архитектури (GPGPU и др.). (2 часа)**

**10. Работа по проект. (2 часа)**

Литература:

- Peter Pacheco, *An Introduction to Parallel Programming*, Morgan Kaufmann, 2011, 392 pages, ISBN: 0-123-74260-9.
- Victor Pankratius, Ali-Reza Adl-Tabatabai, Walter Tichy, *Fundamentals of Multicore Software Development*, CRC Press, 2011, 330 pages, ISBN 1-439-81273-X.
- David B. Kirk, Wen-mei W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach (Third Edition)*, Morgan Kaufmann, 2016, 576 pages, ISBN: 0-128-11986-1.
- Brian Goetz, *Java Concurrency in Practice*, Addison-Wesley Professional, 2006, 432 pages, ISBN: 0-321-34960-1.
- Thomas Rauber, Gudula Rünger, *Parallel Programming: for Multicore and Cluster Systems (Second Edition)*, Springer, 2013, ISBN: 3-642-43806-7.
- Bertil Schmidt, Jorge Gonzalez-Dominguez, Christian Hundt, Moritz Schlarb, *Parallel Programming: Concepts and Practice*, Morgan Kaufmann, 2017, ISBN: 0-128-49890-0.
- Duane Storti, Mete Yurtoglu, *CUDA for Engineers: An Introduction to High-Performance Parallel Computing*, Addison-Wesley Professional, 2015, 352 pages, ISBN: 0-134-17741-X.
- Michael McCool, James Reinders, Arch Robison, *Structured Parallel Programming: Patterns for Efficient Computation*, Morgan Kaufmann, 2012, 432 pages, ISBN: 0-124-15993-1.
- Mihcael J. Quinn, *Parallel programming in C with MPI and OpenMP*, McGraw Hill Education, 2017, 544 pages, ISBN: 0-070-58201-7.
- ED. Pavan Balaji, *Programming Models for Parallel Computing (Scientific and Engineering Computation)*, The MIT Press, 2015, 488 pages, ISBN: 0-262-52881-9.
- Stephen Cleary, *Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming*, O'Reilly Media, 2014, 208 pages, ISBN: 1-449-36756-9.
- Riccardo Terrell, *Concurrency in .NET: Modern patterns of concurrent and parallel programming*, Manning Publications, 2018, 568 pages, ISBN: 978-1617292996.