CS3006 - Parallel and Distributed Computing

Lecture 01: Introduction to Parallel and Distributed Computing

Syed Faisal Ali

email: faisal.ali@nu.edu.pk

National University of Computer and Emerging Sciences - FAST, Karachi Campus

Monday, Aug 21, FALL - 2023

About the Course

- Learn about parallel and distributed architectures.
- Implement different parallel and distributed programming paradigms and algorithms using Message-Passing Interface (MPI) and OpenMP.
- Perform analytical modelling, dependence, and performance analysis of parallel algorithms and programs.
- Use Hadoop or MapReduce programming model to write bigdata applications.

About the Course

Information.	Details
Contact Hours	3+0.
Lectures	Follow my time table shared on GCR.
Consultancy Hours	From next week check GCR.
Interactions	GCR or email.
Course Pre-requisites	PF, DS, Operatig System, CAO.

Evaluation Criteria

- Assignments (15%) - Lab based.
- Quiz (5%).
- Mid Term (I , II) ($15 + 15 = 30\%$).
- Final Exam (50%).

Remember

- Active reading of textbook required.
- Class notes.
- Late submissions are not allowed.
- Plagiarism will be marked as ZERO.
- Make sure that you have 80% attendance in course.

Course Outline

- Introduction: (Introduction: Flynn's Taxonomy, Granularity: fine and coarse grained, some general terms parallel execution, parallel overhead, scalability, Shared Memory paradigm, Distributed memory paradigm, Hybrid – shared and distributed memory.)
- Parallel Programming Platforms: (Scope of Parallelism, Implicit Parallelism, Trends in Microprocessor Architectures, Pipelining and Superscalar Execution, Issue Mechanisms , Efficiency Considerations, Very Long Instruction Word (VLIW) Processors, Limitations of Memory System Performance, Memory System Performance: Bandwidth and Latency, Improving Effective Memory Latency Using Caches.)

Course Outline

- Parallel Programming Platforms: (Storage Impact of Caches , impact of bandwidth, Alternate Approaches for Hiding Memory Latency, Multithreading and prefetching, Interconnection Networks for Parallel Computers, static and dynamic network, Network Topologies: Buses and Crossbar.)
- Principles of Parallel Algorithm Design: (Preliminaries: Decomposition, Tasks, and Dependency Graphs, Multiplying a Dense Matrix with a Vector, Database Query Processing, Granularity of Task Decompositions, Degree of Concurrency, Critical Path Length, Limits on Parallel Performance, Task Interaction Graphs, sparse matrix example, Processes and Mapping.)

Course Outline

- Principles of Parallel Algorithm Design: (Decomposition Techniques: recursive decomposition, Array example, data decomposition, matrix example, itemset frequencies, exploratory decomposition, 15-puzzle example,speculative decomposition, simulation of network nodes. Parallel Algorithm Models: Data Parallel Model, Task Graph Model, Master-Slave Model, Pipeline / Producer-Comsumer Model,Hybrid Models.)

Course Outline

- Programming Shared Address Space (OpenMP):
  (CProgramming Shared Address Space (OpenMP): OpenMP programming model, parallel directive, reduction clause, for loop, nowait clause, scheduling clause: static, dynamic, guided, Data sharing attribute clauses: shared, private, default, reduction, Synchronization clauses: critical, atomic, barrier, ordered. Parallel For Loops, sections directive, OpenML Libarary Functions.)
- Mid-I Exams.

Course Outline

- Programming Using the Message Passing Paradigm:
  (Principles of MPI, The Building Blocks: Send and Receive Operations, Buffered and non bufferedMP, MPI interface, Starting and Terminating the MPI Library, Communicators, Querying Information, Sending and Receiving Messages, overlapping communication with computation, collective communication and computation opertaions: barrier, broadcast, reduction, prefix, scatter, gather.)

Course Outline

- Distributed File System (HDFS): (Client-server file systems, What is a parallel file system, Google File System (GFS), Design Assumptions and design principles. File System Interface, GFS Master and Chunkservers, files, chunks , Master, Client Interaction Model, Reading and writing to files, namespace, Core Part of Google Cluster Environment, HDFS design goals, architecture, namenode, datanode, rackawareness. Hadoop: Distributed file system.)
- Paper: Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. ("The hadoop distributed file system." In 2010 IEEE 26th symposium on mass storage systems and technologies (MSST), pp. 1-10. Ieee, 2010.)

Course Outline

- Map Reduce Framework: (Map, Partition, shuffle, sort, reduce. Example: word count, URL access count, reverse web link graph, inverted index, stock summary.)
- Mid-II Exams.
- Introduction to Data Parallelism and CUDA C: (Data Parallelism, CUDA program structure, Vector Addition Kernel, Device global memory and data transfer, Kernel functions and threading. CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types.)

Course Outline

- Data-Parallel Execution Model: (CUDA thread organization,mapping threads to multidimensional, synchronization and transparent scalability, assigning resources to blocks, query device properties.)
- Final Exams.

Labs in PDC

Lab No.	Lab Tasks
1.	Parallel Execution in OpenMP.
2.	Scheduling in OpenMP.
3.	Critical in OpenMP.
4.	Independent Parallel Task using Sections in OpeMP.
5.	Communication rank and size in MPI.
6.	MPI send and MPI received.
7.	MPI Scatter, Gather, Bcast.
8.	Map Reduce.
9.	CPU/ GPU transfer in CUDA C.

Text Books

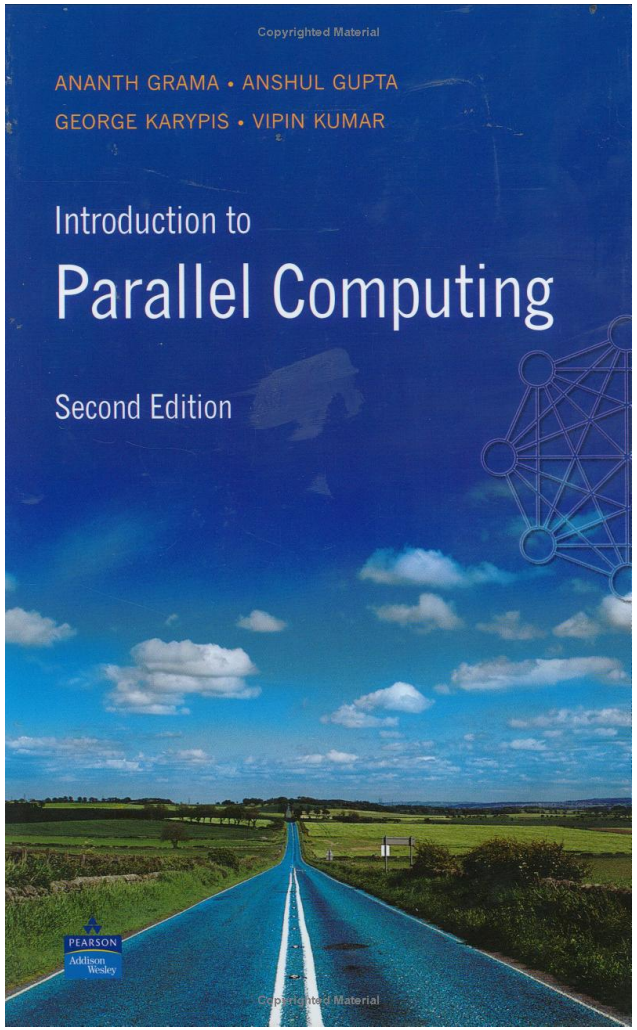


Figure: Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, 2nd Edition

Text Books

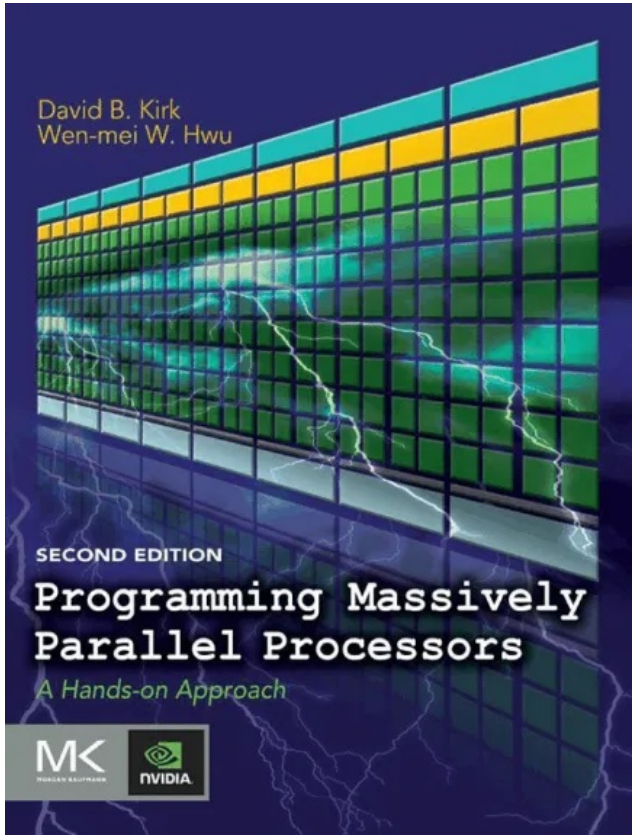


Figure: Programming Massively Parallel Processors by David B. Kirk

Reference Book

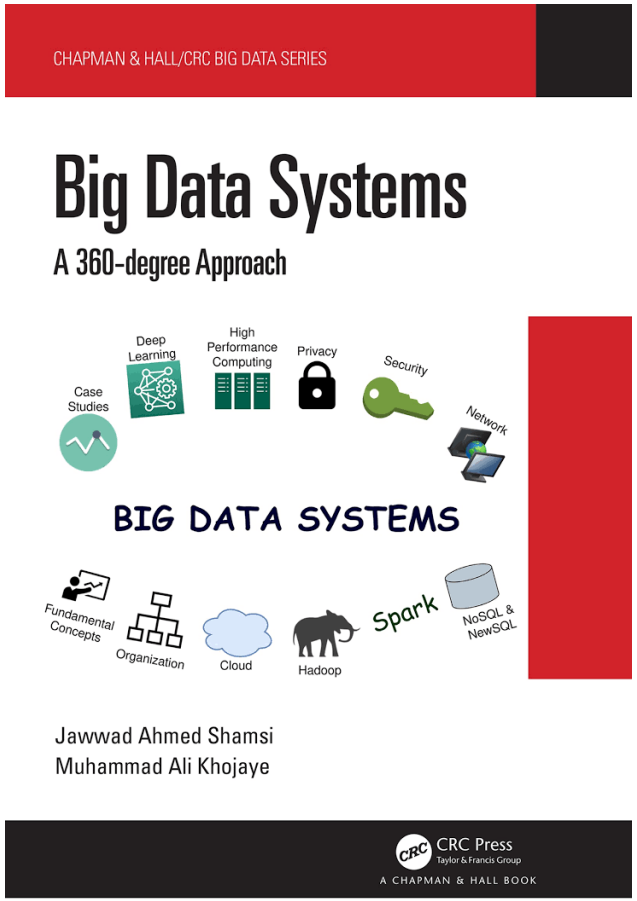


Figure: Big Data Systems: A 360 degree Approach by Jawwad Shamsi

Course Supporting Material on Google Class Room

Ebooks, Lectures, Labs, Software, Handouts

For all the material in this course.

Join the GCR

buufa5y

Programming Assignments

Assignment - 1

Task : OpenMP Assignment (Lab 1, 2,3,4).

Assignment - 2

Task : MPI Assignment (Lab 5,6,7).

Programming Assignments

Assignment - 3

Task 1: Hadoop installation on a single node (screen shot submission of jps) on VM. Task 2: Map Reduce Assignment (Lab 8).

Assignment - 4

(Note: Student list need to be given to IT to make account on GPU server so they can ssh remotely through MobaXterm) Task: CUDA C Assignment (Lab 9)

Why Study Parallel and Distributed Computing?

Reason 1: Performance Improvement

- Parallel and distributed computing techniques allow you to solve complex problems faster and more efficiently by harnessing the power of multiple processors or computers.
- This is especially important for applications that require high computational power, such as scientific simulations, data analysis, and 3D rendering.

Why Study Parallel and Distributed Computing?

Reason 2: Scalability

- As computing demands grow, the ability to scale your applications is critical.
- Parallel and distributed systems can easily scale by adding more processors or nodes, making them suitable for handling large workloads and big data applications.

Why Study Parallel and Distributed Computing?

Reason 3: Cost Efficiency

- Distributing computational tasks across multiple machines can be more cost-effective than investing in a single, high-end machine. Cloud computing platforms also offer scalable and cost-efficient distributed computing resources.

Why Study Parallel and Distributed Computing?

Reason 4: Fault Tolerance

- Distributed systems can be designed to be fault-tolerant, meaning they can continue to function even if some components fail.
- This resilience is crucial for mission-critical applications, such as financial systems or telecommunications networks.

Why Study Parallel and Distributed Computing?

Reason 5: Machine Learning and AI

- Training machine learning models often requires substantial computational resources. Distributed computing accelerates the training process by distributing it across multiple GPUs or servers.

Why Study Parallel and Distributed Computing?

Reason 6: Cloud Computing

- Understanding parallel and distributed computing is essential for utilizing cloud computing services effectively. Major cloud providers offer scalable and distributed infrastructure for various applications.

What is CPU?

Central Processing Unit

- Constructed from millions of transistors, the CPU can have multiple processing cores and is commonly referred to as the brain of the computer. It is essential to all modern computing systems as it executes the commands and processes needed for your computer and operating system. The CPU is also important in determining how fast programs can run, from surfing the web to building spreadsheets.

What is GPU?

Graphics Processing Unit

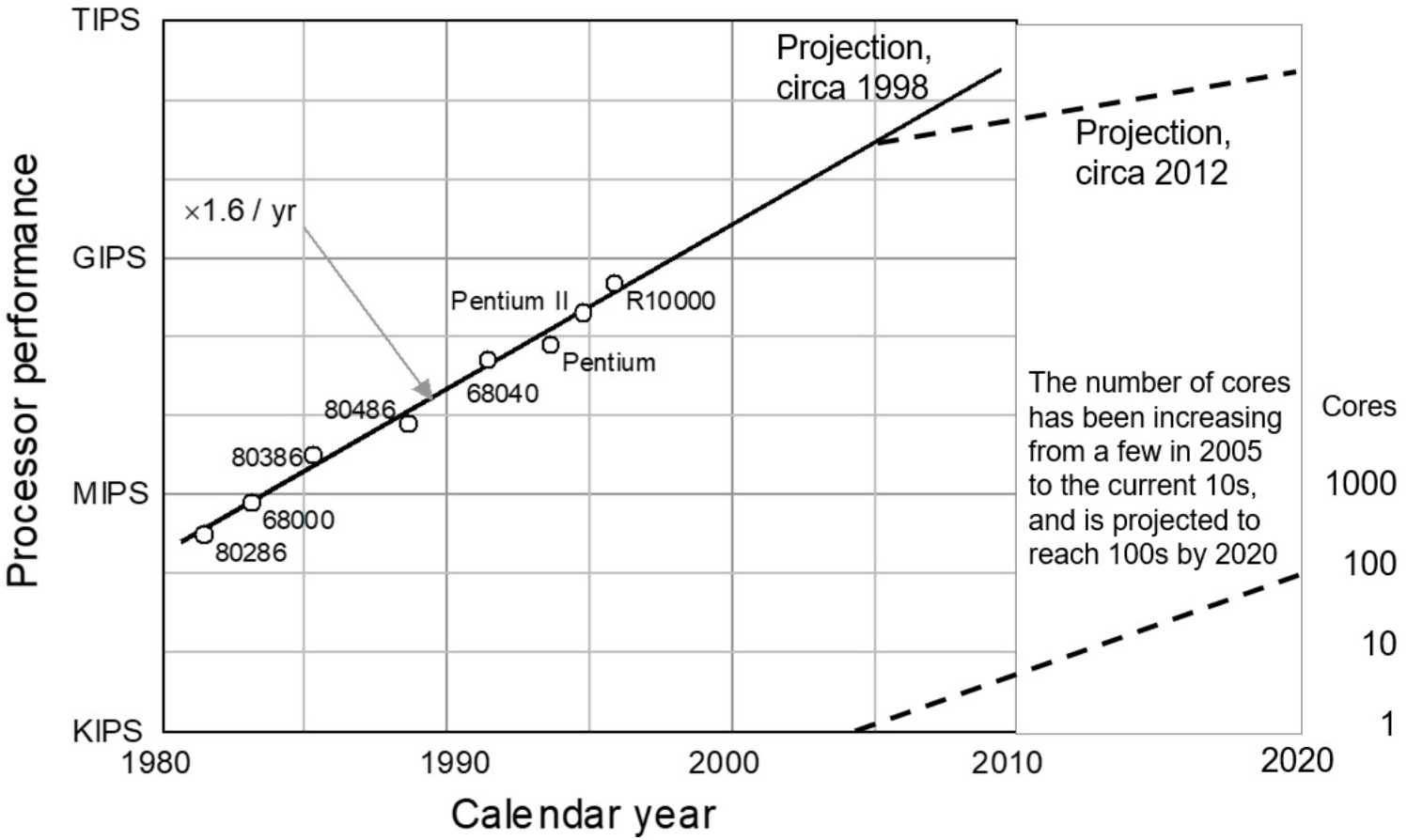
- The GPU is a processor that is made up of many smaller and more specialized cores. By working together, the cores deliver massive performance when a processing task can be divided up and processed across many cores.

What is the difference between a CPU and GPU?

Difference between a CPU and GPU

- CPUs and GPUs have a lot in common. Both are critical computing engines. Both are silicon-based microprocessors. And both handle data. But CPUs and GPUs have different architectures and are built for different purposes.
- The CPU is suited to a wide variety of workloads, especially those for which latency or per-core performance are important.
- GPUs began as specialized ASICs (Application Specific Integrated Circuits) developed to accelerate specific 3D rendering tasks. Over time, these fixed-function engines became more programmable and more flexible.

Why Parallel Processing?



(The exponential growth of microprocessor performance, known as Moore's Law.)

What is Parallel Computing?

Parallel Computing

- Traditionally, software has been written for serial computation:
- To be run on a single computer having a single Central Processing Unit.
- A problem is broken into a discrete series of instructions.
- Instructions are executed one after another.
- Only one instruction may execture at any moment in time.

Serial Processing

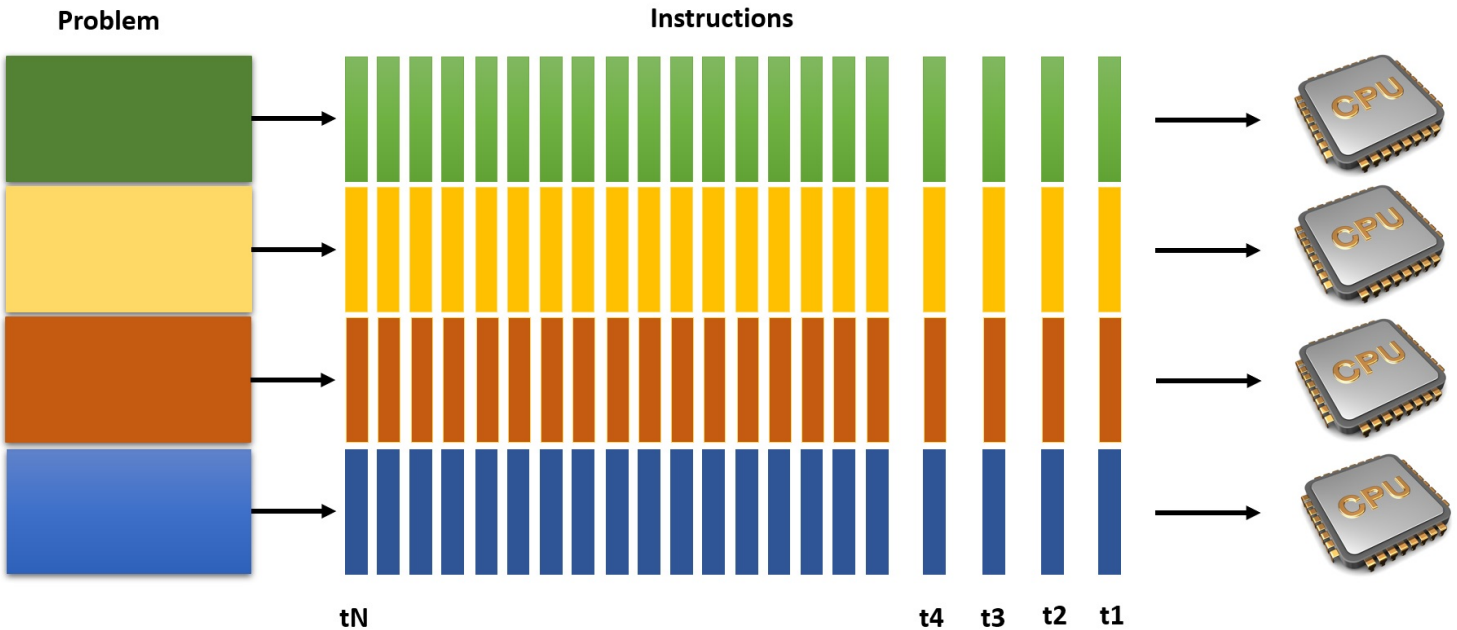
(How a large problem can be divided into small chunks.)

What is Parallel Computing?

Parallel Computing

- In the simplest sense, parallel computing is the simultaneous use of multiple compute resources to solve a computational problem.
- To be run using multiple CPUs.
- A problem is broken into discrete parts that can be solved concurrently.
- Each part is further broken down to a series of instructions.
- Instructions from each part execture simultaneously on different CPUs.

Parallel Processing



(Large problem can be divided into small chunks and processed via multiple CPU's.)

Parallel Computing: Resources

- The compute resources can include:
- A single computer with multiple processors.
- A single computer with (multiple processor(s) and some specialized computer resources (GPU, FPGA, etc).
- An arbitrary number of computers connected by a network.
- A combination of both.

Parallel Computing: The computational problem

- The computational problem usually demostrates characteristics such as the ability to be:
- Broken apart into discrete pieces of work that can be solved simultaneously.
- Execute multiple program instructions at any moment in time.
- Solved in less time with multiple compte resources than with a single compute resource.

Parallel Computing: What for ?

- Example applications include:
- Parallel databases, data mining.
- Web search engines, web based business services.
- Computer-aided diagnosis in medicine.
- Advanced graphics and virtual reality, particularly in the entertainment industry.
- Networked video and multi-media technologies.
- Ultimately, parallel computing is an attempt to maximize the infinite but scarce commodity called time.

Limitations of Serial Computing

- Limits to serial computing - both physical and practical reasons pose significant constraints to simply building ever faster serial computer.
- Transmission speeds - the speed of a serial computer is directly dependent upon how fast data can move through hardware. Absolute limits are the speed of light (30 cm/nanosecond) and the transmission limit of copper wire (9 cm/nanosecond). Increasing speeds necessitate increasing proximity of processing elements.

Limitations of Serial Computing

- Economic limitations - it is increasingly expensive to make a single processor faster. Using a larger number of moderately fast commodity processors to achieve the same (or better) performance is less expensive.

Reading

1. Reading Assignment: Read Chapter 1: Introduction to Parallel Computing 12 to page 20, from Introduction to Parallel Computing 2nd Edition by Ananth Grama.
2. List three major problems requiring the use of supercomputing in the following domains: Structural Mechanics. Computational Biology. Commercial Applications.
3. Collect statistics on the number of components in state of the art integrated circuits over the years. Plot the number of components as a function of time and compare the growth rate to that dictated by Moore's law.

End of Lecture

(نماز فجر کے بعد کی دعا)

حضرت ام سلمہ رضی اللہ عنہ سے روایت ہے کہ رسول اللہ صلی اللہ علیہ وسلم
صبح کی نماز سے سلام پھیرتے تو یہ دعا مانگتے

اَللّٰهُمَّ اِنِّیْ اَسْئَلُكَ عِلْمًا نَّافِعًا وَّرِزْقًا طَيِّبًا وَّعَمَلًا مُّتَقَبَّلًا

(ترجمہ) اے اللہ میں تجھ سے نفع دینے والے علم اور پاکیزہ رزق
اور قبول کیے گئے عمل کا سوال کرتا ہوں۔

(ابن ماجہ، صحیح ابن ماجہ)