MAHARISHI UNIVERSITY OF MANAGEMENT

Higher Consciousness
and Professional Excellence

Parallel Programming
Infinity and a Point

CS471
March 2019

Bruce Lester, Ph.D.

2019
Maharishi's Eleventh Year of
Global Raam Raj

Maharishi University of Management is an Equal Opportunity Institution.

CS471

Parallel Programming: Infinity and a Point

March 2019

Course Overview

	Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**
Week 1	Overview of Multi-Core Processors	Locks and Reduction in OpenMP	Performance Improvements in OpenMP	Memory Issues	Parallel Regions in OpenMP	Lab Day
Week 2	Java Threads	Locks in Java	Thread Synchronization in Java	Lab Day	Midterm Examination	Java Executors and Lambda Expressions
Week 3	Java Executors (con't)	Callable and Future Interfaces	Lab Day	Streams and Map-Reduce	Streams and Map-Collect	Lab Day
Week 4	Java Concurrent Data Structures	Jacobi Relaxation Project	Jacobi Relaxation Project	Final Examination		

CS471
Parallel Programming

Infinity and a Point

March 2019

Dr. Bruce P. Lester

GOAL

To gain knowledge and develop practical skills of writing parallel programs that perform well.

OBJECTIVES

1. To gain knowledge of the fundamental parallel programming patterns, and how to use them.

2. To gain knowledge of the OpenMP parallel programming standard for C and C++.

3. To gain experience with using the parallel programming features of Java 8 for improving the performance of Java programs.

4. To develop practical skills of coding and debugging parallel computer programs with good performance.

5. To gain a broader comprehension of the dynamics of programs involving many computational activities occurring at the same time.

6. To gain a more profound understanding of the cosmic computing of natural law.

SYLLABUS

WEEK ONE

Monday	Lesson 1: Overview of Multi-Core Processors Logarithm of Array Merging Sorted Lists
Tuesday	Lesson 2: Locks and Reduction in OpenMP Bucket Sort (Due Thursday)
Wednesday	Lesson 3: Performance Improvements in OpenMP Exercises OMP1 - 8 Bucket Sort (con't)
Thursday	Lesson 4: Memory Issues Read: Patterns for Cache Optimizaton Matrix Multiplication
Friday	Lesson 5: Parallel Regions Exercise OMP9 Numerical Inegration (due Monday)
Saturday	Lab Day

WEEK TWO

Monday	Lesson 6: Java Threads Read Ch 1 and 2 of *Mastering* text Matrix Multiplication Exercise Java Merging Sorted Lists (due Wednesday)
Tuesday	Lesson 7: Locks in Java Exercise J2, J3, J4, J5 Java Merging Sorted Lists (con't)
Wednesday	Lesson 8: Thread Synchronization in Java Matrix Search from Ch 3 of *Concurrency Cookbook* text Jacobi Relaxation using Cylic Barrier (due Friday)

Thursday	Lab Day
Friday	Midterm Exam
Saturday	Lesson 9 (Part I): Java Executors and Lambda Expressions Read Ch 3 of *Mastering* text k-Nearest Neighbors Algorithm (due Tuesday)

WEEK THREE

Monday	Lesson 9 (Part II): Java Executors (con't) k-Nearest Neighbors example (from *Mastering* text Ch 3)
Tuesday	Lesson 10: Callable and Future Interfaces Read Ch 5 of *Mastering* text Best Matching Algorithm for Words (due Thursday)
Wednesday	Lab Day
Thursday	Lesson 11 (Part I): Streams (Map-Reduce) Numerical Summarizaton Example (from Ch 8 of *Mastering* text) Read Ch 8 of *Mastering* text Student Grades Project (due Saturday)
Friday	Lesson 11 (Part II): Streams (Map-Collect) Online Retail example (from Ch 9 of *Matering* text) Read Ch 9 of *Mastering* text
Saturday	Lab Day Histogram of an Image Project

WEEK FOUR

Monday	Lesson 12: Concurrent Data Structures Read Ch 11 of *Mastering* text Exercise J6, J7
Tuesday	Jacobi Relaxation Project
Wednesday	Jacobi Relaxation (con't)
Thursday	Final Examination

Evaluation Criteria

Midterm Examination	40%
Final Examination	40%
Programming Projects	15%
Attendance	5%

Course Text

Mastering Concurrency Programming with Java 9, Second Edition
by Javier Gonzalez
Packt Publishing, 2017.

References

Java 9 Concurrency Cookbook, Second Edition
by Javier Gonzalez
Packt Publishing, 2017.

Oracle Java Documentation: The Java Tutorials
https://docs.oracle.com/javase/tutorial/index.html

OpenMP Tutorial
Lawrence Livermore National Laboratory
https://computing.llnl.gov/tutorials/openMP/

Class Attendance

Attendance at all classes is required, because all elements of class — lectures, questions and answers, discussions, laboratory work — contribute to the learning process. Absences are usually excused only if you are sick in bed or have a family emergency.

If you must miss a class, please let your instructor know ahead of time. Call, send an email, or send a note with a friend. There is no such thing as a "personal day." If you have personal business to take care of, please schedule it for after class or during the days between blocks. At the same time, it may occasionally be necessary for you to miss a class (or part of a class) for some reason other than illness or family emergency. Please speak with the instructor beforehand, who will be open to considering your needs.

The first lesson of each course is the most important. Students are expected to be present from the first lesson onward. Any student not present on the first morning (except for such compelling reasons as illness or family emergency) may be asked to withdraw from the course. Unexcused absences may result in the student receiving a grade of NC (No Credit) for the whole course.

Lesson 1
Overview of Multi-Core Processors: Infinite Parallelism in Nature

Main Points

1. A Multi-Core Processor has two or more execution cores capable of executing independent instruction streams in parallel. Every point of the universe contains the total potential of natural law.
2. Hyper-Threading allows execution of multiple software threads to be interleaved on a single processor core, thus creating the appearance of parallel execution. The basis of all order is the universe is the infinite organizing power inherent in the field of pure intelligence.
3. Multi-core processors have real parallelism: threads are executed in parallel by different cores. All processes in nature are governed with perfect parallelism from the Unified Field of natural law.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

1. Through software multi-threading, a program can utilize the processor resources more efficiently and improve responsiveness to the User.

2. On a multi-core processor, software multi-threading can significantly reduce program execution time through parallel execution.

---

3. **Transcendental Consciousness** is the field of pure intelligence, the source of all activity.

4. **Impulses within the Transcendental Field** are the basis of the computing of nature.

5. **Wholeness moving within itself**: The cosmic computing of the unified field operates with infinite parallelism to compute the activity of everything in the universe.

Lesson 2
Locks and Reduction:
Perfection of Cosmic Computing

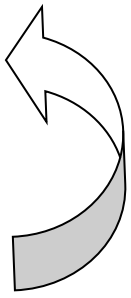
Main Points

1. When shared data values are modified by many threads, timing-dependent errors may result. Natural law coordinates the activity of all processes in nature without an error.
2. Using *locks*, access to shared data by parallel processes can be controlled and coordinated. Present in every grain of the universe, the laws of nature control and coordinates all activity.
3. A *reduction* operation can remove the performance degradation associated with locked regions. By reducing individual awareness to cosmic awareness, all problems and limitations of the individual are removed.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge

1. Locks can be used to prevent data races, but may result in significant performance degradation.

2. Fine grain locking can prevent data races without as much performance loss as a single lock.



---

3. **Transcendental Consciousness** is the field of pure intelligence, the source of all solutions.

4. **Impulses within the Transcendental Field**: The cosmic software of nature operates with perfect efficiency using the principle of least action.

5. **Wholeness moving within itself**: In unity consciousness, one experiences everything as an aspect of oneself: complete and perfect sharing.

Lesson 3
Performance Improvements: Pure Knowledge Has Infinite Organizing Power

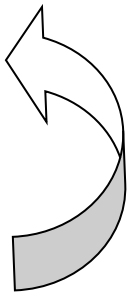
Main Points

1. With Static Scheduling, chunks of equal size are assigned in a round-robin manner to the threads at the start of the loop. Pure intelligence organizes the activity of everything in the universe according to the principle of least action.
2. With Dynamic Scheduling, a new chunk of iterations is assigned to each thread after it finishes with the previous chunk. The field of pure intelligence has a "thread" at every point of the universe and can operate everywhere at the same time.
3. Static Scheduling has a lower overhead, but Dynamic Scheduling may reduce the load imbalance when loop iteration execution times vary. Knowledge has organizing power.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

1. The default scheduling method divides the loop iterations equally among the available threads using equal-size blocks.

2. By using static scheduling with a specific chunk size, or dynamic scheduling, the performance can be improved when there is some variation in the execution time of the loop iterations.



---

3. **Transcendental Consciousness** is the field of perfect order.

4. **Impulses within the Transcendental Field** are the source of the organizing power of natural law.

5. **Wholeness moving within itself**: The cosmic computing of the unified field has infinite organizing power.

Lesson 4
Memory Issues:
Origin of Memory in the Unified Field

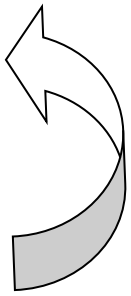
Main Points

1. If many cores are using the same memory location, the copy in the cache will be invalidated quickly, and most accesses will have to use the main memory. An enlightened individual does not lose memory of the Self even when engaged in dynamic activity.
2. If two cores are writing the same cache line frequently, a cache ping-pong effect occurs in which the cores are continually invalidating each other's cache line. The unified field has memory of all its expressions.
3. Interchanging loops in Matrix Multiplication may reduce cache misses and result in a significant performance improvement. Nature operates with the principle of least action.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge

1. Memory access issues may seriously degrade the performance of a parallel program.

2. Through careful design of parallel programs, cache misses can be significantly reduced to improve program performance.



---

3. **Transcendental Consciousness** is the self-referral state of consciousness, individual intelligence has memory of its infinite state.

4. **Impulses within the Transcendental Field**: Pure intelligence has memory of all possible expressions.

5. **Wholeness moving within itself**: The individual in unity consciousness has regained complete memory of the Self.

Lesson 5
Parallel Regions:
Diversity arises from Unity

Main Points

1. A parallel region creates a team of threads that all execute the same code in parallel and then synchronize at the end of the region, returning control to the master thread. The master thread of the universe is the field of pure intelligence.
2. A for-statement in a parallel region will divide the loop iterations among the threads to be executed in parallel. Loop iterations are an example of self-referral through a sequence of steps. The unified field has instanteous self-referral, resulting in unlimited computing power.
3. Sections in a parallel region define distinct blocks of code that will be executed in parallel by different threads. All diversity in the universe results from the collapse of infinity to its own point.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge

1. Parallel-for pragmas in OpenMP all loop iterations to be executed in parallel.

2. Parallel Regions in OpenMP allow more flexibility in allocation computing tasks to parallel threads.

---

3. **Transcendental Consciousness** is the self-referral state of consciousness, in which pure intelligence knows itself.

4. **Impulses within the Transcendental Field** compute the activity of everything in the universe without a problem.

5. **Wholeness moving within itself**: All activity in nature is the flow of pure intelligence withing itself.

Lesson 6
Java Threads:
Infinity at a Point

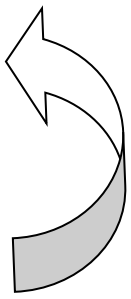
Main Points

1. Any class that implements the Runnable interface may be turned into a Thread object that is executed as a parallel activity. Infinite parallelism is displayed by the cosmic computing of natural law.
2. Lock objects in the java.util.concurrent library perform the same function as OpenMP Locks. The unified field is freely available to everyone —it is never locked.
3. In the Matrix Multiplication program, the parallelism is limited by the overhead associated with creating parallel threads. The unified field manages the parallel activity of everything in the universe without any loss of time.
4. To increase concurrency, the Read-Write lock in the java.util.concurrent library allows multiple concurrent readers, but only one writer.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

1. OpenMP has a few simple primitives for creating and controlling parallel computing: parallel-for, parallel regions, locks, reductions, and loop scheduling.

2. Java has primitives for creating parallel threads, along with a wide range of additional high-level primitives to make parallel programming easier.



3. **Transcendental Consciousness** is the self-referral state of consciousness, in which pure intelligence knows itself.

4. **Impulses within the Transcendental Field**: The cosmic software of nature exists in its totality in every grain of the universe.

5. **Wholeness moving within itself**: All parallel computing in nature arises from the collapse of infinity to its own point.

Lesson 7
Locks in Java:
Perfection of Cosmic Computing

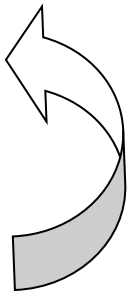
Main Points

1. Lock objects in the java.util.concurrent library perform the same function as OpenMP Locks. The unified field is freely available to everyone —it is never locked
2. Putting code inside a synchronized block makes the compiler append instructions to acquire the lock on the specified object before executing the code and release it afterwards. Present in every grain of the universe, the unified field controls and coordinates all activity.
3. To increase concurrency, the Read-Write lock in the java.util.concurrent library allows multiple concurrent readers, but only one writer. Natural law coordinates the activity of all processes without an error.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

1. Use of shared data by parallel processes can result in timing-dependent errors or performance degradation from contention.

2. The intelligent use of locks can create atomic operations for access to shared data, while minimizing contention.



3. **Transcendental Consciousness** is the self-referral state of consciousness, in which pure intelligence knows itself.

4. **Impulses within the Transcendental Field**: The cosmic software of nature exists in its totality in every grain of the universe.

5. **Wholeness moving within itself**: In unity consciousness, one experiences everything as an aspect of oneself: complete and perfect sharing.

Lesson 8
Thread Synchronization in Java: No Barriers in Pure Intelligence

Main Points

1. Java Barriers are useful for synchronizing a group of parallel threads. From the level of pure intelligence, there are no barriers to the fulfillment of any desire.
2. Jacobi Relaxation can be used to solve Laplace's Equation using a two-dimensional grid of points. Every point of the universe contains the total potential of natural law.
3. Cyclic Barrier action can be adapted to provide a convergence test in a parallel Jacobi Relaxation program. The cycle of Rishi (knower), Devata (process of knowing), and Chhandas (known), creates boundaries from the unbounded field of pure intelligence.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

1. In parallel iterative algorithms, threads work on different portions of the data array, and gradually move in the direction of a numerical solution.

2. With an efficient parallel convergence test, the iterations will continue until the desired accuracy is achieved.

---

3. **Transcendental Consciousness** is the self-referral state of consciousness, in which pure intelligence knows itself.

4. **Impulses within the Transcendental Field**: Each sound of the Veda collapses into the unmanifest gap, from where the next sound emerges in sequence.

5. **Wholeness moving within itself**: All parallel computing in nature arises from the collapse of infinity to its own point.

Lesson 9 – Part I
Java Executors:
Pure Intelligence is the Source of All Computing

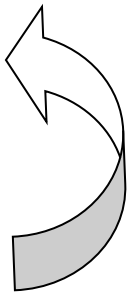
Main Points

1. A Thread Pool contains a work queue which holds tasks waiting to get executed (*Runnable* objects). A desire on the level of the pure intelligence will be fulfilled immediately.
2. The threads are constantly running and are checking the work query for new work. The unified field computes the activity of natural law at every point simultaneously.
3. The Executor framework contains interfaces and classes to support thread pools. The practice of the TM technique provides the individual an interface to the unified field.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

1. Java Threads can perform computing tasks in parallel.

2. Using Java Executors and Lambdas simplifies the code and makes parallel programming easier.



---

3. **Transcendental Consciousness** is the self-referral state of consciousness, in which pure intelligence knows itself.

4. **Impulses within the Transcendental Field**: The cosmic software of nature exists in its totality in every grain of the universe.

5. **Wholeness moving within itself**: All parallel computing in nature arises from the unified field of pure intelligence.

Lesson 9 – Part II
Lambda Expressions:
Pure Intelligence Knows Itself

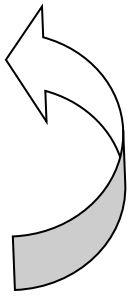
Main Points

1. A *lambda expression* is the Java counterpart of a free-standing function definition, and can be passed around to be executed later, or multiple times. Infinite parallelism is displayed by the cosmic computing of natural law.
2. Whenever an object of a functional interface is expected, a lambda expression can be supplied instead. All forms and activity in the universe are an excitation of the unified field.
3. A Java lambda expression can be passed around as if it was an object and executed on demand. All object and activity arise from pure intelligence knowing itself.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge

1. Prior to Java-8, all functions had to be created using methods, which was sometimes cumbersome.

2. Lamdas provide free-standing Java functions to simplify the code.



---

3. **Transcendental Consciousness** is the self-referral state of consciousness, in which pure intelligence knows itself.

4. **Impulses within the Transcendental Field**: The cosmic software of nature exists in its totality in every grain of the universe.

5. **Wholeness moving within itself**: All parallel computing in nature arises from the unified field of pure intelligence.

Lesson 10
Callable and Future Interfaces:
Spontaneous Computing of Natural Law

Main Points

1. *Callables* are functional interfaces just like runnables, but instead of being *void*, they return a value. All desires on the level of pure intelligence are immediately fulfilled.
2. When a *Callable* task is sent to an Executor, it will return a *Future* object. The Future will be better if one continues to practice the TM technique.
3. The *get*( ) method of the Future object gets the value returned by the Callable task. In the state of enlightenment, the individual can get anything he/she wants.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge

1. *Runnable* Java objects can be used to represent computing tasks.

2. *Callable* tasks extend Runnables by adding a return value represented by a *Future*.

---

3. **Transcendental Consciousness** is the field of self-referral awareness.

4. **Impulses within the Transcendental Field**: The activity of natural law is computed spontaneously within the unified field.

5. **Wholeness moving within itself**: All parallel computing in nature arises from the unified field of pure intelligence knowing itself.

Lesson 11 – Part I
Streams and Map-Reduce:
Evolution from Pure Intelligence

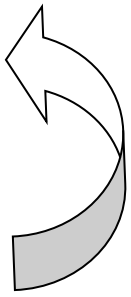
Main Points

1. A *stream* is not a data structure that stores elements; instead, it conveys elements from a source such as a data structure, an array, a generator function, or an I/O channel, through a pipeline of computational operations. Do less and accomplish more.
2. An operation on a stream produces a result but does not modify its source. Pure intelligence guides the evolution of everything but itself remains unchanged.
3. Stream operations are divided into *intermediate* (Stream-producing) operations and *terminal* (value- or side-effect-producing) operations. All activity in nature arises from the unified field of pure intelligence.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge

1. The addition of the *Stream* is one of the major new functionality in Java 8.

2. *Streams* are wrappers around a data source, allowing us to operate with that data source and making bulk processing convenient and fast.



3. **Transcendental Consciousness** is the self-referral state of consciousness, in which pure intelligence knows itself.

4. **Impulses within the Transcendental Field** compute the activity of everything in the universe without a problem.

5. **Wholeness moving within itself**: The self-referral flow of pure intelligence expresses itself into the evolution of the universe.

Lesson 11 – Part II
Streams and Map-Collect:
Collapse of Infinity to a Point

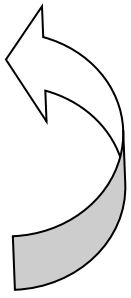
Main Points

1. *collect*( ) is an extremely useful terminal operation to transform the elements of the stream into a different kind of result, e.g. a List, Set or Map. The unified field is the source of all transformatons.
2. Stream API allows reducing a sequence of elements to some value according to a specified function with the help of the *reduce*( ) method of the type Stream. The reduction of Infinity to a Point is the first step in manifestation of the unified field.
3. To perform a sequence of operations over the elements of the data source and aggregate their results, three parts are needed – the source, intermediate operation(s) and a terminal operation. The three basic parts of the unified field are Rishi (knower), Devata (knowing), and Chhandas (known).

Connecting the Parts of Knowledge
With the Wholeness of Knowledge

1. Java 7 has a variety of library classes to support parallelism, including thread creation, thread pools, locks, and blocking queues.

2. Java 8 has some new primitives to make parallel programming easier, including parallel streams.



---

3. **Transcendental Consciousness** is the self-referral state of consciousness, in which pure intelligence knows itself.

4. **Impulses within the Transcendental Field**: The cosmic software of nature exists in its totality in every grain of the universe.

5. **Wholeness moving within itself**: All parallel computing in nature arises from the collapse of infinity to its own point.

Lesson 12
Java Concurrent Data Structures: Perfection of Cosmic Computing

Main Points

1. Java contains a variety of thread-safe data structures including Blocking Queue, Concurrent HashMap, Concurrent Linked Queue, and Copy-on-Write Arrays. Natural Law manages the parallel activity of everything in the universe without a problem.
2. A Data Race in a parallel program can lead to nondeterministic and incorrect results. The cosmic computing of nature does make any errors.
3. Data Races can be eliminated using Atomic Variables and Atomic Adders. The unified field of pure intelligence is shared by all without any race conditions.

Connecting the Parts of Knowledge With the Wholeness of Knowledge

1. Parallel operations on shared data can lead to data races that are difficult to detect.

2. Java has a variety of features to avoid data races including locking and thread-safe data structures.

3. **Transcendental Consciousness** is the simplest state of awareness, completely unified.

4. **Impulses within the Transcendental Field** create diversity from within the field of unity.

5. **Wholeness moving within itself**: In unity consciousness, unity and diversity coexist in one grand wholeness of awareness.