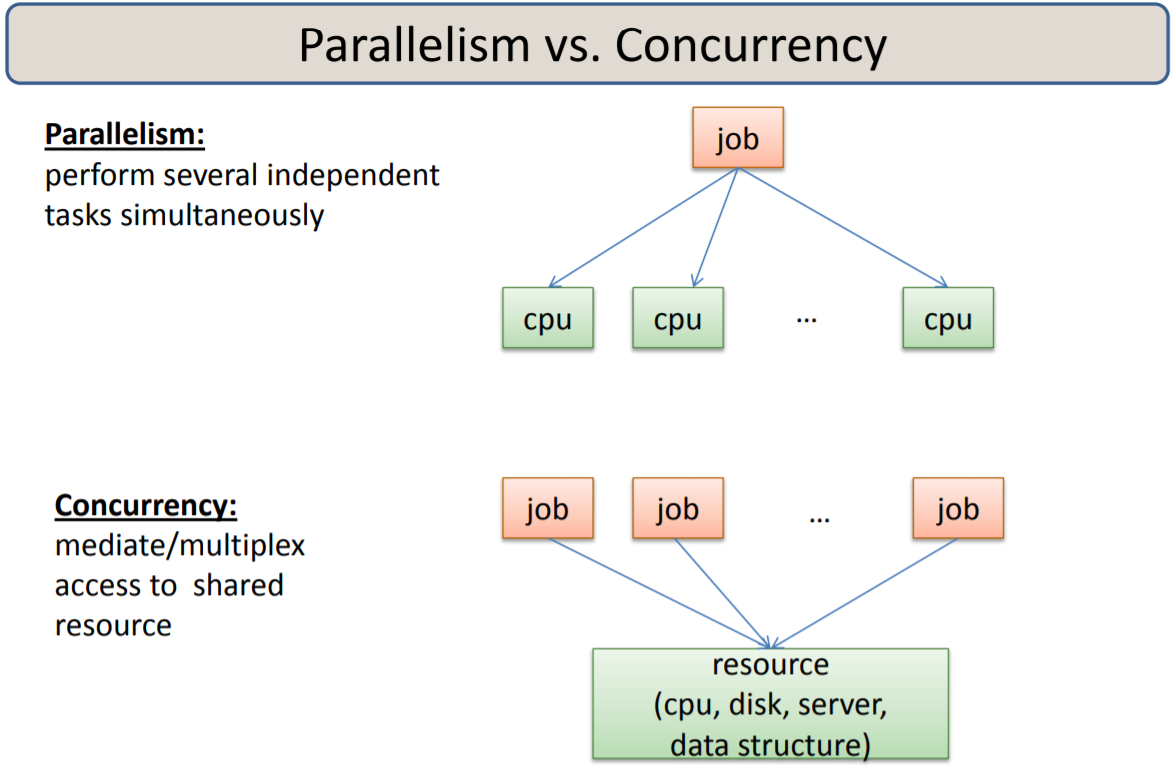
**Q1**

Parallelism:

An application splits its tasks up into smaller subtasks which can be processed in parallel, for instance on multiple CPUs at the exact same time.

1. Concurrency:

An application is making progress on more than one task at the same time (concurrently). Well, if the computer only has one CPU the application may not make progress on more than one task at exactly the same time, but more than one task is being processed at a time inside the application. It does not completely finish one task before it begins the next. Instead, the CPU switches between the different tasks until the tasks are complete.



1. CPU core:

A CPU core is a single independent execution unit that can fetch instructions and execute them one by one simultaneously like other cores.

**public** **class** **CPUCores** {

**public** **static** **void** **main**(String[] args) {

**int** processors = Runtime.getRuntime().availableProcessors();

System.out.println("CPU cores: " + processors);

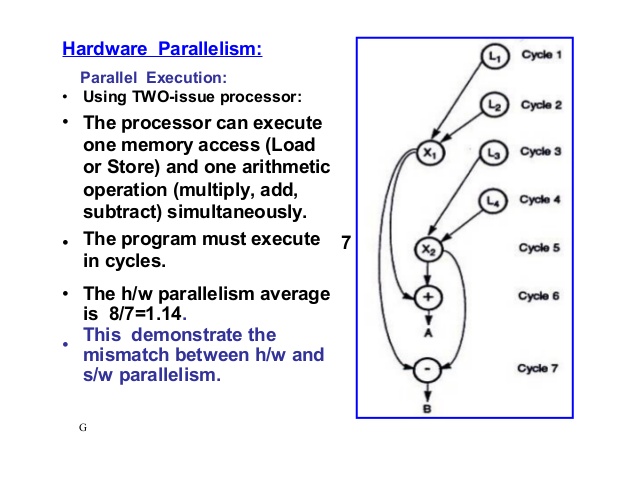
}

}

1. Hardware Parallelism:

This refers to the type of parallelism defined by the machine architecture and hardware multiplicity.

Hardware parallelism is a function of cost and performance tradeoffs. It displays the resource utilization patterns of simultaneously executable operations. It can also indicate the peak performance of the processors.



1. 32-bit versus 64-bit architecture：
2. 32 bit:
3. bit is a type of CPU architecture which is capable of transferring 32 bits of data. It is the amount of information which can be processed by your CPU whenever it performs an operation.
4. bit can access 232 memory addresses. And it’s not an ideal option for stress testing and multi-tasking. By the way, 32-bit systems limited to 3.2 GB of RAM 32 bit Windows. It addresses limitation doesn't allow you to use full 4GB of Physical memory space.

i.e 4 GB of RAM or physical memory.

1. 64 bit:

In computer architecture, 64 bit refers to the number of bits which should be processed or transmitted in parallel or the number of bits used for a single element in a data format. A 64-bit microprocessor allows computers to process data and memory address which are represented by 64 bits.

1. bit can access 2 memory addresses. But it’s can work best for performing multi-tasking and stress testing. By the way, 64-bit systems will enable you to store up to 17 Billion GB of RAM.

i.e actually 18-Quintillion GB of RAM

**Q2**

ParallelExample2:

The getData() method returns a list of String about the alphabet. The sentence “alpha.stream().forEach(System.out::println);” print the whole list orderly from letter “a” to letter “z”. The sentence “alpha2.parallelStream().forEach(System.out::println);” print the whole list out of order from letter “a” to letter “z”.

The output of the code will be:

Normal...

a

b

c

d

e

f

g

h

i

j

k

l

m

n

o

p

q

r

s

t

u

v

w

x

y

z

Parallel...

q

s

r

x

b

u

a

w

d

h

y

n

l

e

z

c

v

j

t

p

k

m

o

i

f

g

ParallelExample3a:

The sentence “IntStream range = IntStream.rangeClosed(1, 10);” create an int stream from 1 to 10. The method “range.isParallel()” is a Boolean that judges whether stream “range” is paralleled or not. The sentence “range.forEach(System.out::println);” print the whole stream one by one. The sentence “IntStream range2Parallel = range2.parallel();” makes the stream “range2” paralleled.

So the output of the code will be:

Normal...

false

1

2

3

4

5

6

7

8

9

10

Parallel...

true

7

6

8

2

10

1

9

5

4

3

ParallelExample3b:

3b is similar to 3a but add the print commend to print the name of the thread. When the stream is not paralleled, the print commend will be executed all by main thread. When the stream is paralleled, the print commend will be executed randomly by main or by the threads in the ForkJoinPool that is created automatically.

So the output will be:

Normal...

Thread : main, value: 1

Thread : main, value: 2

Thread : main, value: 3

Thread : main, value: 4

Thread : main, value: 5

Thread : main, value: 6

Thread : main, value: 7

Thread : main, value: 8

Thread : main, value: 9

Thread : main, value: 10

Parallel...

Thread : main, value: 7

Thread : main, value: 6

Thread : ForkJoinPool.commonPool-worker-23, value: 8

Thread : ForkJoinPool.commonPool-worker-5, value: 9

Thread : ForkJoinPool.commonPool-worker-23, value: 10

Thread : ForkJoinPool.commonPool-worker-19, value: 3

Thread : ForkJoinPool.commonPool-worker-27, value: 4

Thread : ForkJoinPool.commonPool-worker-13, value: 1

Thread : ForkJoinPool.commonPool-worker-5, value: 5

Thread : ForkJoinPool.commonPool-worker-9, value: 2

**Q4:**

a) When JVM requests for a class, it invokes loadClass function of the ClassLoader by passing the fully classified name of the Class.

The loadClass function calls for findLoadedClass() method to check that the class has been already loaded or not. It’s required to avoid loading the class multiple times.

If the Class is not already loaded then it will delegate the request to parent ClassLoader to load the class.

If the parent ClassLoader is not finding the Class then it will invoke findClass() method to look for the classes in the file system.

b) The output will be:

Loading Class 'YuchenZhao.NO4.Foo.Foo'

Foo Constructor >>> 1212 1313

Bar Constructor >>> 1212 1313

Bar ClassLoader: jdk.internal.loader.ClassLoaders$AppClassLoader@15db9742

Foo ClassLoader: [jdk.internal.loader.ClassLoaders$AppClassLoader@15db9742](mailto:jdk.internal.loader.ClassLoaders$AppClassLoader@15db9742)

c) The path of the GraderDriver class is added to the argument of the CCRun class. The CCRun class will compile and run the GraderDriver class. So the output will be:

Loading Class 'YuchenZhao.NO4.Student.GradesDriver'

Thread-1 completed!

Thread-2 completed!

Thread-3 completed!

Thread-4 completed!

Thread-5 completed!

Name: A ID: 1 Midterm: 63 Project: 61 Final: 50 Grade: F

Name: B ID: 2 Midterm: 59 Project: 74 Final: 78 Grade: C

Name: C ID: 3 Midterm: 52 Project: 94 Final: 77 Grade: C

Name: D ID: 4 Midterm: 57 Project: 59 Final: 52 Grade: F

Name: E ID: 5 Midterm: 53 Project: 78 Final: 55 Grade: D

Thread-6 completed!

Thread-7 completed!

Thread-8 completed!

Thread-9 completed!

Thread-10 completed!

Name: A ID: 1 Midterm: 63 Project: 61 Final: 50 Grade: F

Name: B ID: 2 Midterm: 59 Project: 74 Final: 78 Grade: C

Name: C ID: 3 Midterm: 52 Project: 94 Final: 77 Grade: C

Name: D ID: 4 Midterm: 57 Project: 59 Final: 52 Grade: F

Name: E ID: 5 Midterm: 53 Project: 78 Final: 55 Grade: D

Name: F ID: 6 Midterm: 77 Project: 82 Final: 55 Grade: D

Name: G ID: 7 Midterm: 68 Project: 87 Final: 74 Grade: C

Name: H ID: 8 Midterm: 79 Project: 56 Final: 80 Grade: C

Name: I ID: 9 Midterm: 61 Project: 88 Final: 99 Grade: B

Name: J ID: 10 Midterm: 84 Project: 71 Final: 95 Grade: B

Thread-11 completed!

Thread-12 completed!

Thread-13 completed!

Thread-14 completed!

Thread-15 completed!

Name: A ID: 1 Midterm: 63 Project: 61 Final: 50 Grade: F

Name: B ID: 2 Midterm: 59 Project: 74 Final: 78 Grade: C

Name: C ID: 3 Midterm: 52 Project: 94 Final: 77 Grade: C

Name: D ID: 4 Midterm: 57 Project: 59 Final: 52 Grade: F

Name: E ID: 5 Midterm: 53 Project: 78 Final: 55 Grade: D

Name: F ID: 6 Midterm: 77 Project: 82 Final: 55 Grade: D

Name: G ID: 7 Midterm: 68 Project: 87 Final: 74 Grade: C

Name: H ID: 8 Midterm: 79 Project: 56 Final: 80 Grade: C

Name: I ID: 9 Midterm: 61 Project: 88 Final: 99 Grade: B

Name: J ID: 10 Midterm: 84 Project: 71 Final: 95 Grade: B

Name: K ID: 11 Midterm: 88 Project: 93 Final: 62 Grade: C

Name: L ID: 12 Midterm: 99 Project: 50 Final: 62 Grade: D

Name: M ID: 13 Midterm: 50 Project: 86 Final: 62 Grade: D

Name: N ID: 14 Midterm: 61 Project: 88 Final: 55 Grade: D

Name: O ID: 15 Midterm: 80 Project: 93 Final: 78 Grade: B

Thread-16 completed!

Thread-17 completed!

Thread-18 completed!

Thread-19 completed!

Thread-20 completed!

Name: A ID: 1 Midterm: 63 Project: 61 Final: 50 Grade: F

Name: B ID: 2 Midterm: 59 Project: 74 Final: 78 Grade: C

Name: C ID: 3 Midterm: 52 Project: 94 Final: 77 Grade: C

Name: D ID: 4 Midterm: 57 Project: 59 Final: 52 Grade: F

Name: E ID: 5 Midterm: 53 Project: 78 Final: 55 Grade: D

Name: F ID: 6 Midterm: 77 Project: 82 Final: 55 Grade: D

Name: G ID: 7 Midterm: 68 Project: 87 Final: 74 Grade: C

Name: H ID: 8 Midterm: 79 Project: 56 Final: 80 Grade: C

Name: I ID: 9 Midterm: 61 Project: 88 Final: 99 Grade: B

Name: J ID: 10 Midterm: 84 Project: 71 Final: 95 Grade: B

Name: K ID: 11 Midterm: 88 Project: 93 Final: 62 Grade: C

Name: L ID: 12 Midterm: 99 Project: 50 Final: 62 Grade: D

Name: M ID: 13 Midterm: 50 Project: 86 Final: 62 Grade: D

Name: N ID: 14 Midterm: 61 Project: 88 Final: 55 Grade: D

Name: O ID: 15 Midterm: 80 Project: 93 Final: 78 Grade: B

Name: P ID: 16 Midterm: 78 Project: 75 Final: 71 Grade: C

Name: Q ID: 17 Midterm: 80 Project: 62 Final: 56 Grade: D

Name: R ID: 18 Midterm: 66 Project: 88 Final: 82 Grade: C

Name: S ID: 19 Midterm: 92 Project: 54 Final: 55 Grade: D

Name: T ID: 20 Midterm: 78 Project: 84 Final: 96 Grade: B

Thread-21 completed!

Thread-22 completed!

Thread-23 completed!

Thread-24 completed!

Thread-25 completed!

Name: A ID: 1 Midterm: 63 Project: 61 Final: 50 Grade: F

Name: B ID: 2 Midterm: 59 Project: 74 Final: 78 Grade: C

Name: C ID: 3 Midterm: 52 Project: 94 Final: 77 Grade: C

Name: D ID: 4 Midterm: 57 Project: 59 Final: 52 Grade: F

Name: E ID: 5 Midterm: 53 Project: 78 Final: 55 Grade: D

Name: F ID: 6 Midterm: 77 Project: 82 Final: 55 Grade: D

Name: G ID: 7 Midterm: 68 Project: 87 Final: 74 Grade: C

Name: H ID: 8 Midterm: 79 Project: 56 Final: 80 Grade: C

Name: I ID: 9 Midterm: 61 Project: 88 Final: 99 Grade: B

Name: J ID: 10 Midterm: 84 Project: 71 Final: 95 Grade: B

Name: K ID: 11 Midterm: 88 Project: 93 Final: 62 Grade: C

Name: L ID: 12 Midterm: 99 Project: 50 Final: 62 Grade: D

Name: M ID: 13 Midterm: 50 Project: 86 Final: 62 Grade: D

Name: N ID: 14 Midterm: 61 Project: 88 Final: 55 Grade: D

Name: O ID: 15 Midterm: 80 Project: 93 Final: 78 Grade: B

Name: P ID: 16 Midterm: 78 Project: 75 Final: 71 Grade: C

Name: Q ID: 17 Midterm: 80 Project: 62 Final: 56 Grade: D

Name: R ID: 18 Midterm: 66 Project: 88 Final: 82 Grade: C

Name: S ID: 19 Midterm: 92 Project: 54 Final: 55 Grade: D

Name: T ID: 20 Midterm: 78 Project: 84 Final: 96 Grade: B

Name: U ID: 21 Midterm: 83 Project: 56 Final: 98 Grade: B

Name: V ID: 22 Midterm: 77 Project: 77 Final: 85 Grade: B

Name: W ID: 23 Midterm: 90 Project: 99 Final: 53 Grade: C

Name: X ID: 24 Midterm: 89 Project: 72 Final: 67 Grade: C

Name: Y ID: 25 Midterm: 91 Project: 68 Final: 54 Grade: D

Thread-26 completed!

Thread-27 completed!

Thread-28 completed!

Thread-29 completed!

Thread-30 completed!

Name: A ID: 1 Midterm: 63 Project: 61 Final: 50 Grade: F

Name: B ID: 2 Midterm: 59 Project: 74 Final: 78 Grade: C

Name: C ID: 3 Midterm: 52 Project: 94 Final: 77 Grade: C

Name: D ID: 4 Midterm: 57 Project: 59 Final: 52 Grade: F

Name: E ID: 5 Midterm: 53 Project: 78 Final: 55 Grade: D

Name: F ID: 6 Midterm: 77 Project: 82 Final: 55 Grade: D

Name: G ID: 7 Midterm: 68 Project: 87 Final: 74 Grade: C

Name: H ID: 8 Midterm: 79 Project: 56 Final: 80 Grade: C

Name: I ID: 9 Midterm: 61 Project: 88 Final: 99 Grade: B

Name: J ID: 10 Midterm: 84 Project: 71 Final: 95 Grade: B

Name: K ID: 11 Midterm: 88 Project: 93 Final: 62 Grade: C

Name: L ID: 12 Midterm: 99 Project: 50 Final: 62 Grade: D

Name: M ID: 13 Midterm: 50 Project: 86 Final: 62 Grade: D

Name: N ID: 14 Midterm: 61 Project: 88 Final: 55 Grade: D

Name: O ID: 15 Midterm: 80 Project: 93 Final: 78 Grade: B

Name: P ID: 16 Midterm: 78 Project: 75 Final: 71 Grade: C

Name: Q ID: 17 Midterm: 80 Project: 62 Final: 56 Grade: D

Name: R ID: 18 Midterm: 66 Project: 88 Final: 82 Grade: C

Name: S ID: 19 Midterm: 92 Project: 54 Final: 55 Grade: D

Name: T ID: 20 Midterm: 78 Project: 84 Final: 96 Grade: B

Name: U ID: 21 Midterm: 83 Project: 56 Final: 98 Grade: B

Name: V ID: 22 Midterm: 77 Project: 77 Final: 85 Grade: B

Name: W ID: 23 Midterm: 90 Project: 99 Final: 53 Grade: C

Name: X ID: 24 Midterm: 89 Project: 72 Final: 67 Grade: C

Name: Y ID: 25 Midterm: 91 Project: 68 Final: 54 Grade: D

Name: Z ID: 26 Midterm: 86 Project: 69 Final: 59 Grade: C

Name: AA ID: 27 Midterm: 50 Project: 70 Final: 76 Grade: D

Name: AB ID: 28 Midterm: 68 Project: 84 Final: 97 Grade: B

Name: AC ID: 29 Midterm: 80 Project: 75 Final: 80 Grade: C

Name: AD ID: 30 Midterm: 61 Project: 64 Final: 64 Grade: D

Thread-31 completed!

Thread-32 completed!

--End—

**Q5:**

I create a thread pool of 51 threads. One of them will be asked to execute the GraderThread class and the rest of them will be asked to execute the 50 student classes. In the end, the shutdown() method is called to ensure that there will not be any more threads created. The awaitTermination(Long.MAX\_VALUE, TimeUnit.MINUTES) method is called to ensure all of the threads in the thread pool finished there execution.