Concurrency

Friday, 27 September 2024 11:12 AM

www.agiledeveloper.com

Parallel and Async
Nature of problem
When to use Parallel --> Divide and concur does not work for parallel str
When to use asynchronous

Parallel Stream --
Give a example of list of even integer square sum

Collection Pipe line pattern
From imperative to functional
Benefits of pipeline pattern
Parallel as Master switch
Sequential execution
Observing the thread -- display the Thread name in fucntion
Order of execution
Controlling the order
Parallel and filter
Parallel and map
Parallel and Reduce
Using parallel streams
ON IO Problem
ON Computational problem
Formula to decide the number of threads
Configuring programmatically
Parallel is not always fast
When we need to go for parallel (collection is big

eam

You can give example for concurrency when debugging the code for perf
concurency

Elegance and efficient

The Past

Structure of concurrent code is very different from the structure of sequ

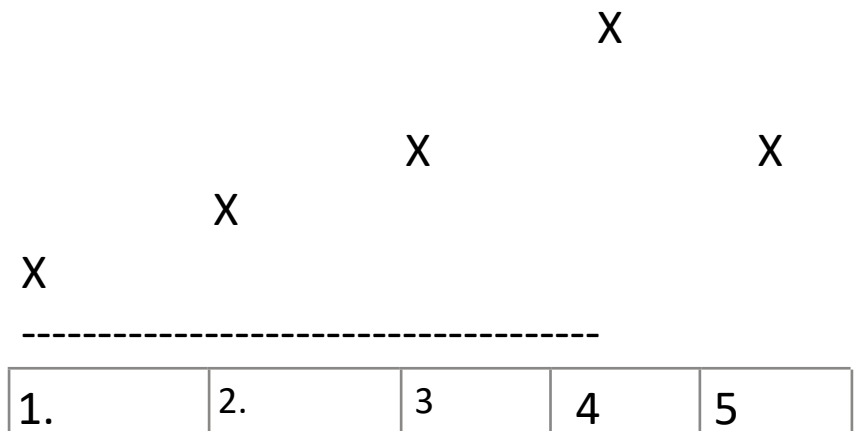
Structure of concurrent code is same from the structure of sequential co

If you are author of the stream then you can use parallel stream or you u
If the last call is sequential then entire program will run sequential and if
entire function will run parallel

Stream has ability to create the process as parallel

How many threads we can create
How many thread should I create the best question
In Computation intensive operation # threads should be less than or equ
4 monks -
T1 T2

Time



formance and searching the

ential code

ode

use parallel() method .
f last call is parallel then

ual to number of cores

Threads (# cores is 4 on your system).
Give example I made parallel of program and it took more time than bef

IO Intensive operation

$$0 < \text{blocking factor} < 1$$

$$\#\ \text{Thread} <= \frac{\text{number of Cores}}{1 - \text{Blocking factor}}$$

If blocking factor is 0.5 that means half of the time CPU is idle then 1 -0.5 == 0.5 , Here assur
cores are 10 10/0.5 ==> 20 threads .

When we try to get he common pool information it will be always one parallelism less is bec
running the main thread will also involve in the task provided if the main thread is not occup
steelng

Example below
java.util.concurrent.ForkJoinPool@5b6f7412[Running, parallelism = 10, size = 9, active = 0, r
submissions = 0]
Djava.util.concurrent.ForkJoinPool.common.parallelism = 100

Where we run the terminal operation is matter rather than the when created in stream out

Completable Feature

- Asynchronous execution
- Drawbacks of Future

Callbacks

- Lacks consistency
- Hard to compose

Promise has Data and error track
------ Data------ f1-f2-f3

ore

mption is made like number of

ause of when forkJoin pool is
ied. It called main thread work

unning = 0, steals = 6, tasks = 0,

of collection

-------e1,e2------

Func

.then(f1)
.cathc(e1)
.then(f2)
.catch(e2)

runAysnc
SupplyAsync

Available

29 --
2 months

Feb -10
TCS offer did not join
PF offer

Runnable ()
Void run. It does not take any input does not give any thing
Future , Call get and get result
Java script is Async program
Callbakcls
Lacks consistency
Hard to compose
Hard to deal with error

They introduced Promise

-----Data--f1-f2
-----Error---e1-e2

```
Func ()
    .then()
    .catch()
```

If java promises , it is called CompletableFeature

Run in pool
ThenacceptAsync will accept the pool it should run