Solving Problems in Parallel

Pragya Jain
Computer Service Center
Indian Institute of Technology, Delhi

Problem

There are 1000 scripts to be marked. Each script has 4 answers. Each answer takes 5 minutes to be marked.

Sequential Solution

Sequential Solution

Procedure for evaluation

- Step 1: Take an answer book from the pile of scripts (input)
- Step 2: for I = 1 to 4 do step 3
- Step 3: Grade answer to Q_i.
- Step 4: Add marks given for each question
- Step 5: Put the script in the pile of marked answer books (output)
- Step 6: Repeat steps 1 to 5 until no more scripts are left in the input

Sequential Solution

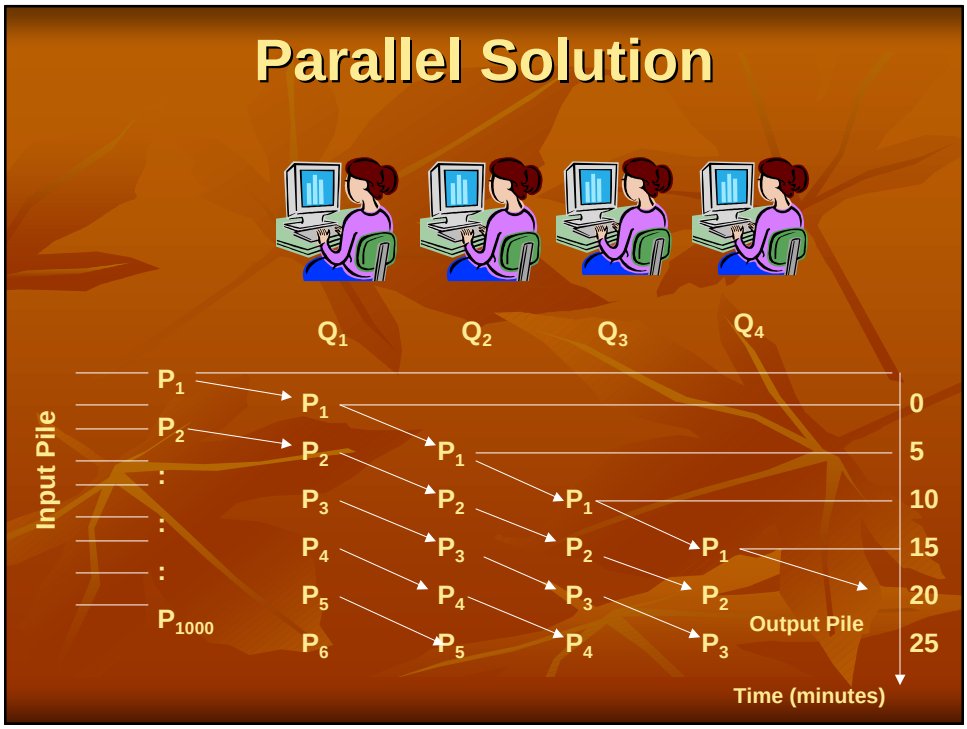
If one teacher is evaluating these 1000 answer scripts, he/ she takes 20 minutes for each script. So the total time taken is 20000 minute. This is sequential solution of the problem.

Disadvantage

Takes a lot of time.

To reduce time, we introduce parallelism

Parallel Solution



Parallel Solution

Total time taken =

20 + (999*5) min.

5015 min.

¼ (sequential time)

Parallel Solution

Temporal Parallelism

or

Assembly Line Processing

or

Pipeline Processing

Most appropriate if

- The jobs to be carried out are identical
- A job can be divided into many independent tasks
- Time taken for each task is the same
- Time taken to send the task to one worker to another is negligible as compared to the time taken to evaluate that task
- The number of tasks is much smaller as compared to the number of jobs to be carried out (1000 scripts vs 4 answers)

Temporal Parallelism

Let the number of jobs = n (1000)
Let the number of tasks = k (4)
Let the time taken to do a job = p (20)

Let each job be divided into k tasks and each task take equal time to execute. Let each task be executed by different workers.

Time taken to do one task = p/k
Time taken to complete n jobs with no pipeline processing = np
Time taken to do n jobs with k workers in pipeline = p + (n-1)*p/k
Speedup due to pipeline =

$$\frac{np}{p\,(k + n - 1)\,/\,k} = \frac{k}{1 + \frac{k - 1}{n}}$$

Temporal Parallelism

If n » k then (k-1)/n « 1 and the speed up is nearly equal to k.

Thus the speedup is directly proportional to the number of workers working in the pipeline provided the number of jobs is very large as compared to the number of tasks per job.

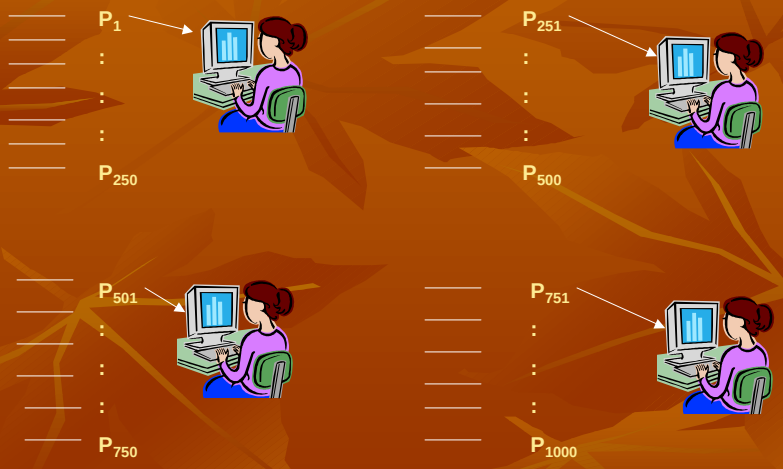
Problems

- **Synchronisation:** equal time should be taken in performing each task.
- **Bubbles in pipeline:** If some tasks are absent in the pipeline e.g. some questions are unanswered, bubbles are formed i.e. the workers of those jobs will remain idle for some time.
- **Fault tolerance:** The system does not tolerate faultsi.e. if one teachers gets up and goes for coffee, the whole pipeline is affected.
- **Intertask communication:** The time taken to pass answer books between the teachers should be negligible as compared to the time taken to mark one question.

Advantage

The main advantage of this method is that each worker in the pipeline can be fine tuned to do one kind of job. All the workers need not be experts in all the jobs.

Data Parallelism



Data Parallelism

Total time taken =

20 * 250

= 5000 min

= ¼ (sequential time)

Advantages

- There is no communication required as each teacher works independently.
- There is no synchronization is required between the teachers.
- Bubbles do not cause a problem because the teacher getting bubbles will finish his/ her job faster without affecting the others.
- It is more fault tolerant. If one teacher gets up and goes for coffee, the others are not affected.

Disadvantages

- The assignment of jobs to each teacher is pre-decided. This is called static assignment. If a teacher is fast, he/ she will finish the work fast and then sit idle. The teacher who is slow will slow down the whole process of result formulation.
- The set of jobs must be partitionable into equal number of mutually independent jobs. Each subset should take the same time to complete.
- Each teacher must be capable of grading all the answers.
- The time taken to divide the jobs into equal subsets of jobs must be very small.

Combined Data and Temporal Parallelism

Combined Dat and Temporal Parallelism

Total time taken =

20 + 499*5

= 2515 min

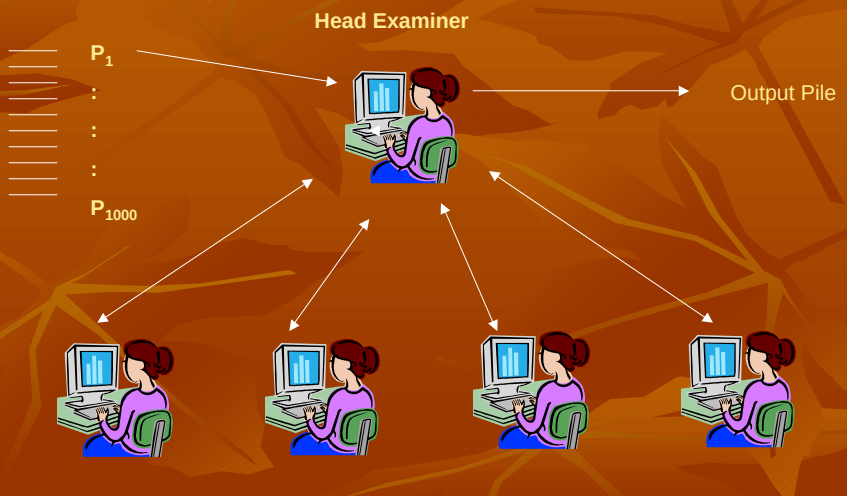
(sequential time)

Combined Data and Temporal Parallelism

- It reduces time to complete the total job.
- It has advantages and disadvantages of both data and temporal parallelism.
- Method is effective only if the number of jobs given to each pipeline is much larger than the number of stages in the pipeline.

Multiple Pipeline

Data Parallelism with Dynamic Assignment



Scalability of the method

Let the total number of answer books = n

Let time taken to grade i^{th} script = p_i

Sequential time required = $\sum_{i=1}^{n} p_i$

Let k teachers be working together

Let waiting time for a teacher to get script from head examiner = q_i

Time taken by each teacher = $p_i + q_i$

Total time taken = $\frac{1}{k}\sum_{i=1}^{n}(q_i + p_i)$

Speed up due to parallelism = $\frac{\sum_{i=1}^{n} p_i}{\left(\sum_{i=1}^{n}(q_i + p_i)\right)/k}$

Scalability of the method

Let $\bar{q} = \left(\sum_{i=1}^{n} q_i\right)/n$

$\bar{p} = \left(\sum_{i=1}^{n} p_i\right)/n$

As long as $\bar{q} \ll \bar{p}$ the speed up approaches k, the ideal value. If this condition is not satisfied, speed up is low.

Advantages

- Balancing of work assigned to each teacher is done dynamically.
- A fast teacher does not sit idle because of another slow teacher.
- Problem of bubbles is not there. It speeds up the process.
- Overall time taken to grade answer books is minimised.

Disadvantages

- If many teachers finish at the same time then there is a long queue for the examiner and some teachers have to remain idle.
- The head examiner can become a bottleneck. If he gets up and goes for coffee, all the teachers will be idle. However if one teacher goes for coffee, it does not affect the process.
- The head examiner is himself idle between handing over the scripts.
- It is difficult to increase the number of teachers because that increases the chances of many teachers finishing the job at the same time.

Data Parallelism with Quasi-dynamic Scheduling

The problem in the previous method can be reduced by handing over unequal set of answer scripts to different teachers e.g. give 7, 9, 11, 13 scripts to teacher no. 1, 2, 3, 4 respectively. This will randomize the time of their completion of the job and thus queue formation. When they complete the job, they may be given another bunch of scripts.

Quasi-dynamic Scheduling

Data parallelism worked on the principal of static scheduling and previous method was dynamic scheduling. This method is called quasi-dynamic scheduling which is a mid-way through purely static and purely dynamic assignment. In this method, the time taken by the head examiner to hand over one copy each time is reduced by handing over a set of copies. Also, if one teacher finishes the job fast, he/ she can get another set of scripts immediately without being idle.

Comparison between Temporal and Data Parallelism

Temporal Parallelism

- Job is divided into a set of independent tasks and tasks are assigned for processing.
- Tasks should take equal time. Pipeline stages should thus be synchronized.
- Bubbles in jobs lead to idling of processors.
- Processors specialized to do specific tasks efficiently.
- Task assignment static.
- Not tolerant to faults.
- Efficient with fine grained tasks.
- Scales well as long as no. of jobs to be processed is much larger than the no. of processors in the pipeline and communication time is minimal.

Data Parallelism

- Full jobs are assigned for processing.
- Jobs may take different times. No need to synchronize beginning of jobs.
- Bubbles do not cause idling of processors.
- Processors should be general purpose and may not do every job efficiently.
- Job assignment may be static, dynamic or quasi-dynamic.
- Tolerates faults.
- Efficient with coarse grained tasks and quasi-dynamic scheduling.
- Scales well as long as no. of jobs are much greater than the no. of processors and processing time is much higher than the communication time.

Data Processing with Specialized Processors

- Fine grain Data Parallelism – Specialist Data Parallelism
- Coarse grain Data Parallelism – Coarse grain Specialist Temporal Parallelism

Specialist Data Parallelism

- Give one answer book to each teacher.
- When a correctedscript is returned, check if answers to all questions are graded. If yes, add marks and put the script in output pile.
- If no, check which questions are not graded.
- Give script to teacher T_i for ungraded question Q_i
- Repeat steps 2,3,4 until script remains in input pile and all teachers are idle.

Disadvantage

- Load is not balanced in this method. If one question takes more time to be graded then the others will be idle for that time.
- The same problem occurs if one question is not answered by many students.
- The head examiner wastes a lot of time checking the script for unanswered questions and the teachers are sitting idle at that time.

Coarse grain specialist Temporal Parallelism

To reduce time of each teacher waiting for the other to finish marking we adopt another method. Divide the whole set of scripts into 4 equal part. The teachers sit in a circle. Every teacher has an in tray and an out tray. Every teacher marks his/ her question from the in tray, puts it in the out tray. When the in tray is empty, waits for the in tray of the next teacher to be empty, empties his/ her out tray in the next in tray. All scripts will be graded when each teacher's out-tray is filled 4 times.

Coarse Grain Specialist Parallelism

- Step 1: Take a script from in-tray.
- Step 2: Grade question Q_i and put in out-tray.
- Step 3: Repeat steps 1&2 till no answer books are left in in-tray.
- Step 4: Check if next teacher's in-tray is empty.
- Step 5: As soon as it is empty, empty your out-tray in it and wait till your in-tray is filled.

Each teacher will repeat these steps 4 times.

Advantages

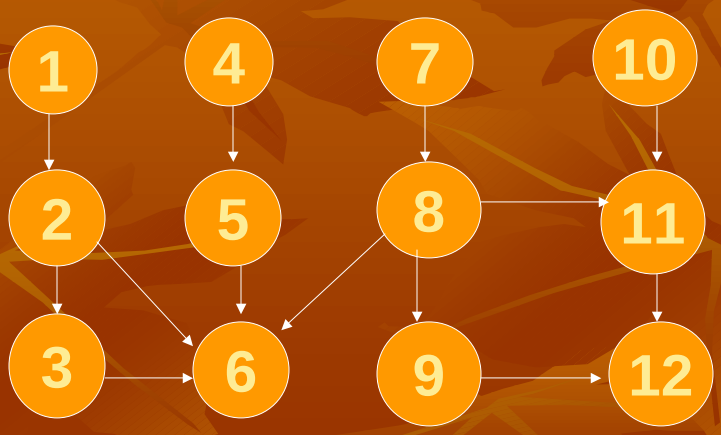
- Processing of special tasks are done by specialized processors.
- The method uses the concept of pipelined processing in a circular pipeline.
- There is buffering (in-tray & out-tray) between pipeline stages.
- Each stage has a chunk of work to do.
- Does not need strict synchronization.
- Tolerates bubbles.

Inter-task Dependency

We have seen examples of jobs where tasks are independent of each other. But in real life, the tasks are inter-dependent.

For example

Inter-task Dependency



Inter-task Dependency

One method of assigning tasks is:

Assign Q_1, Q_2 and Q_3 to Teacher 1. assign Q_4, Q_5 and Q_7 to Teacher 2.assign Q_8, Q_6 and Q_9 to Teacher 3. finally Q_{10}, Q_{11} and Q_{12} to Teacher 4.

Inter-task Dependency

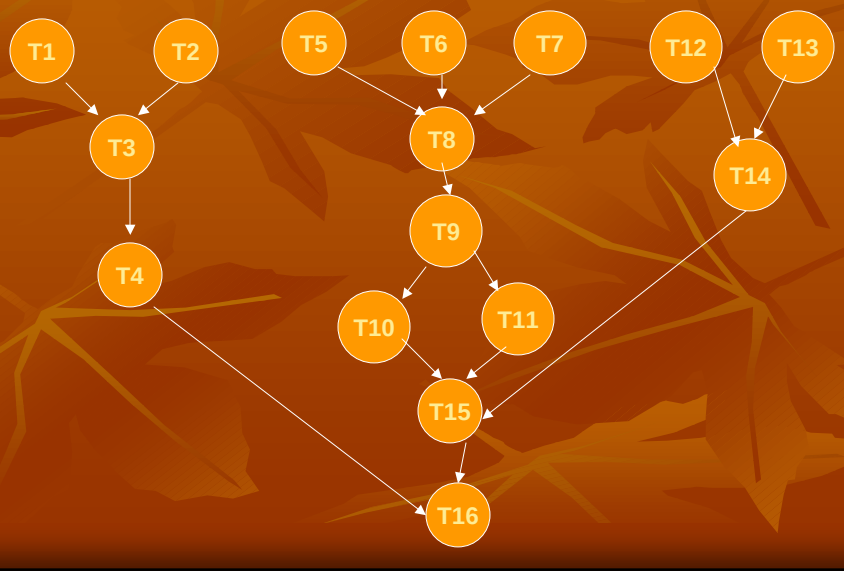
In this example, all the tasks are identical and take almost the same time to complete.

In real life, situation may be different.

Recipe for Chinese Fried Rice

- T1: Clean and wash rice (5)
- T2: Boil water in a vessel with I tsp. salt. (10)
- T3: Put rice in boiling water. Add some oil and cook till soft. (15)
- T4: Drain rice and cool. (10)
- T5: Wash and scrape carrots. (5)
- T6: Wash and string french beans. (10)
- T7: boil water with ½ tsp. salt in two vessels. (8)
- T8: Drop carrots and french beans separately in boiling water and keep for 1 minute. (1)
- T9: Drain and cool carrots and french beans. (10)
- T10: Dice carrots. (10)
- T11: Dice french beans. (10)
- T12: Peel onions and dice into small pieces. Wash and chop spring onions. (15)
- T13: Clean cauliflower. Cut into small pieces. (12)
- T14: Heat oil in a pan and fry diced onion and cauliflower for 1 minute in heated oil. (4)
- T15: Add diced carrots, french beans to above and fry for 2 minutes. (2)
- T16: Add cooled cooked rice, chopped spring onions and soya sauce to the above and stir and fry for 5 minutes. (5)

Chinese Fried Rice



Chinese Fried Rice

If there were no sequencing constraints and 4 cooks work simultaneously, then

$$\text{Time taken} = \frac{T1+T2+\ldots\ldots+T16}{4}$$

= 132/4

= 33 minutes

Chinese Fried Rice

Step1: Find tasks which can be carried out in parallel at level 1. sum their total time. In this case, the sum for tasks 1,2,5,6,7,12,13 = 65 minutes.

As there are 4 cooks, time = 65/4 = 16 minutes approx.

Step2: Find tasks which can be carried out at level 2. they are T3, T8 and T14.

Step3: At next level, tasks that be carried out in parallel are T4 and T9.

Chinese Fried Rice

Step4: Tasks that can be allocated next are T10 and T11.

Step5: In the next level, only T15 can be allocated.

Step6: Only T16 is now left. It cannot start till T4 and T15 are complete.