CA READING ASSIGNMENT REPORT

R.Amshu Naik (B20CS046)

SCHEDULING AND RESOURCE MANAGEMENT

Scheduling in the operating system handles running of a process.It manages all the related activities of ready , waiting queue for the process executing. Processes in the operating system depend on the scheduling algorithm being used to execute the process in a certain order, for example in the case of priority scheduling algorithm, on the basis of process priority the process is executed .We selected the PAPER 1, PAPER 2, PAPER 3 keeping scheduling as the common topic. In paper we did analysis of different scheduling algorithms , their applications, drawbacks etc for and the performance metrics .

PAPER : 2DFQ: Two-Dimensional Fair Queueing for Multi-Tenant Cloud Services

- **Paper title :** 2DFQ: Two-Dimensional Fair Queueing for Multi-Tenant Cloud Services
- **Venue** : SIGCOMM '16: Proceedings of the 2016 ACM SIGCOMM Conference
- **Publication year** : 2016
- **Web link** : https://dl.acm.org/doi/10.1145/2934872.2934878

PROBLEM STATEMENT

Most of the datacenter services such as storage, queueing, and other most of the services are shared among multiple tenants simultaneously, due to better efficiency, and scalability.All these tenants compete in a process for resources.Fair queue schedulers like WFQ (Weighted Fair Queueing) and WF2Q do not provide fairness in sharing due to insufficient resource isolation which causes starvation , high latencies , reduced throughput whenever multiple tenants try to execute a request. It slows down working of the cpu. .

Resource allocation and sharing using WFQ lead to many challenges like :

- Resource concurrency due to sequential execution of the requests.
- High and different variance cost of different tenants as well as also because of no interruption of the executing of requests due to a high priority request .
- No details of resource cost and requirements of the system before its execution.

That's why I came up with ideas for 2DFQ due to the tendency to have a mix of predictable, cheap and expensive requests in real workloads and its deterioration

capability is not bad.2DFQ ensures to provide better quality of service than WFQ and W2FQ.

METHODOLOGICAL CONTRIBUTIONS

In this paper we came to know the features and the advantages provided by 2DFQ over WFQ and WF^2Q .We came to know the possible ways provided by 2DFQ request scheduler to overcome the challenges based on fair queue schedulers.
Here in the paper we are most concerned about the effective way of dealing with unknown request costs using the pessimistic estimation strategy and book keeping mechanisms such as retroactive charging and refresh charging.
Retroactive charging: Every time at the scheduled time ,the counter is updated.Using retroactive charging we try to minimize the difference between virtual timing and resource usage for every request to guarantee that the tenant will receive its true fair share of resources in the long run.
Refresh charging: It is a damage control mechanism which periodically measures the resource usage by the requests.It makes us notice when the tenant submits cheap requests and transitions itself to expensive requests then, 2DFQ becomes susceptible to underestimation which lowers the virtual time. We found that refresh charging for every 10ms had no significant overhead.
Other contributions in PAPER 1 are as follows:

- Production traces from Azure Storage , used to demonstrate scheduling challenges arised due to concurrency and variable, unpredictable request costs.
- Handling of unknown request cost, despite the presence of unpredictable requests using 2DFQ.
- Improving Ways to avoid bursty schedules based on biasing requests of different sizes to different threads not like WF^2Q.
- 2DFQ to extend cost-based partitioning with pessimistic cost estimation to mitigate the impact of unpredictable tenants.
- Reduces mean and tail latency, for predictable workloads when they contend against large or unpredictable requests to improve the servicing.

KEY TAKEAWAY FROM METHOD AND RESULTS

From the methodologies mentioned in the paper we got to know a lot of features of the 2DFQ scheduler and how it tries to provide solutions to the challenges produced in case of WFQ or WF^2Q.
Features provided by 2DFQ scheduling algorithm are :

- **Minimize burstiness** : It is observed that WFQ and WF^2Q provide a bursty scheduler .In case of WFQ schedulers get burst if they provide high and zero order of throughput and in case of WF^2Q working with multiple threads causes the burstiness. But 2DFQ makes a request eligible at different times for different

worker threads and prioritizes working on high index threads rather than low index threads.

- **Partition request** : 2DFQ partitions request on the basis of size so as to make requests to work on high index threads.
- **Improve performance of tenants** : 2DFQ improves 99% latency for predictable tenants and also reduces the magnitude of service lag variation with small tenants.
- **Provide stability :** Provide stability and consistently smooth service with occasional oscillations provided by 2DFQ is more stable.
- **Fairness** : 2DFQ treats unpredictable tenants as expensive , and produces fair and smooth schedules in systems that can process multiple requests in a small time interval.

DRAWBACKS

- **Delay in execution**: When systems get backlogged, and underutilized, expensive requests are serviced by threads which cause delay for small requests by 2DFQ scheduler.
- **Over saturation of CPU** : Over-saturation of the CPU and slows the running of requests are caused if we try to make work-conserving.
- **Latency** : 2DFQ does not improve 99th percentile latency for tenants with large and/or unpredictable requests as in case of small and predictable requests.
- **Variation in cost** : Many challenges faced by 2DFQ scheduler due to wide variation in cost in cloud services such as affect the execution efficiency.

FUTURE WORK

- Need to compute ways to reduce the cost variation of cloud services by 2DFQ , by splitting long requests to short requests as much possible to make the requests to be executed by low index threads.
- Another task would be to improve the way of dealing with unknown cost estimation requests.

CONTRIBUTIONS

R. Amshu Naik (B20CS046) : Contributed towards reading and analysis of paper 1 about “2DFQ: Two-Dimensional Fair Queueing for Multi-Tenant Cloud Services”, and in understanding different section for the report such as problem statement, methodological contributions, drawbacks etc of PAPER 1. I learnt a lot from the paper and am looking forward to continuing further research on the topic in the near future.