

Workload Scheduling

Sergei Trifonov (Core Team)

31 October 2023



||||· ClickHouse

Agenda

01

What is workload scheduling?

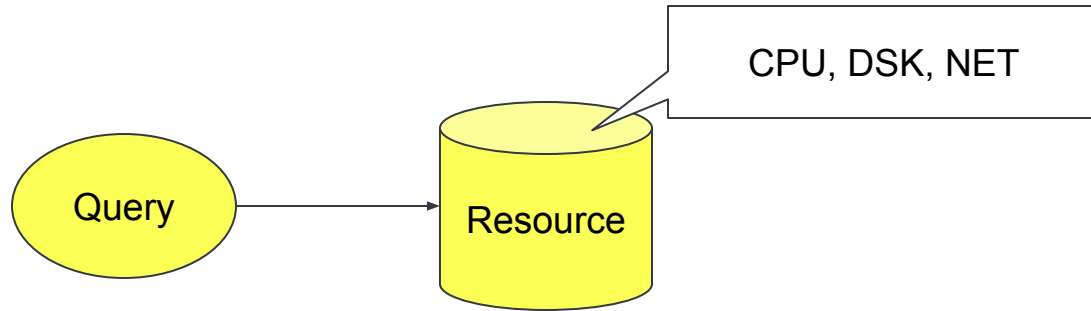
02

How to configure?

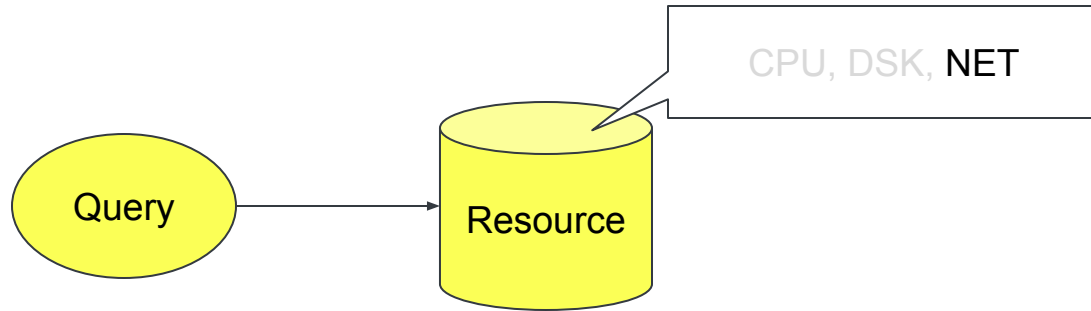
03

How to apply in practice?

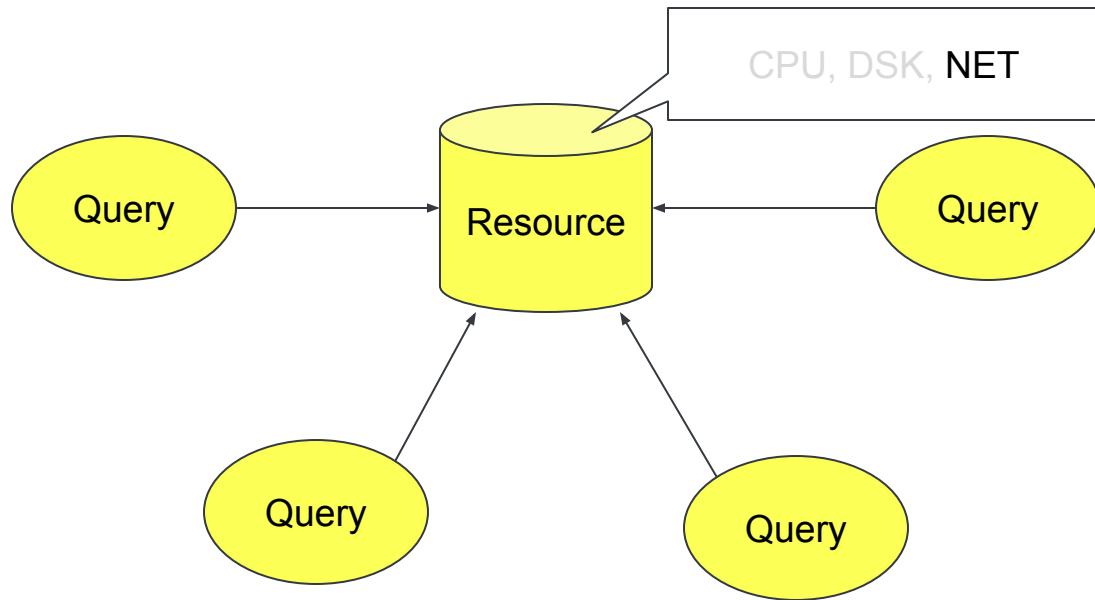
What is workload scheduling?



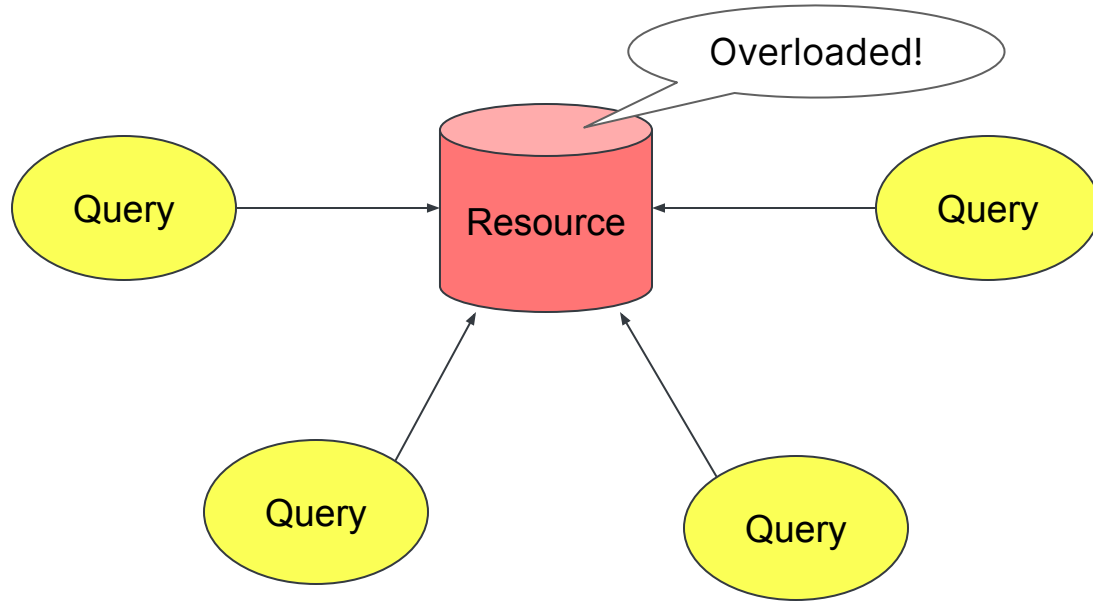
What is workload scheduling?



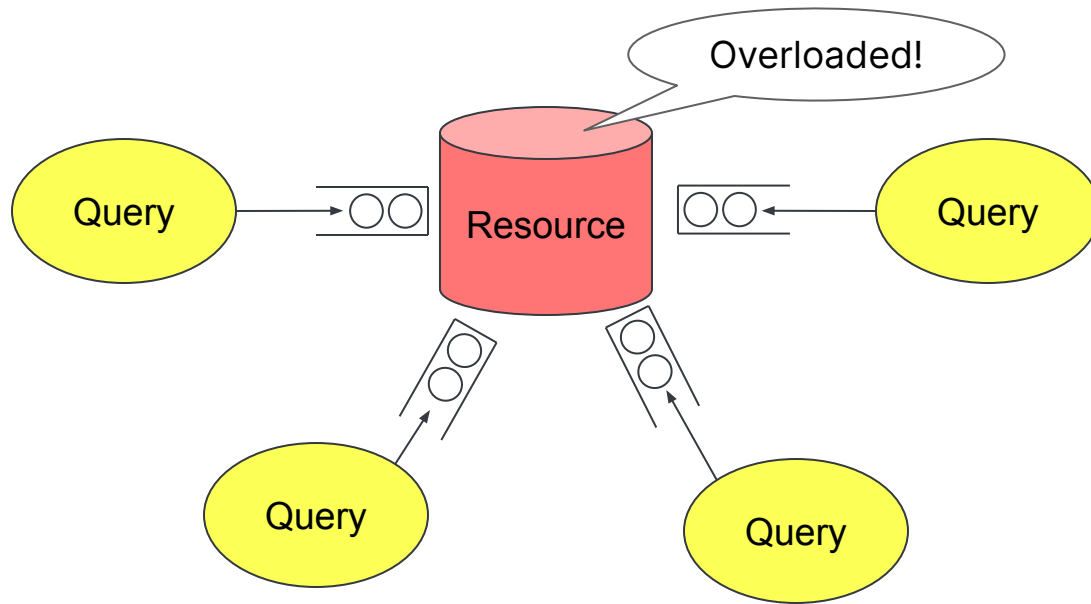
What is workload scheduling?



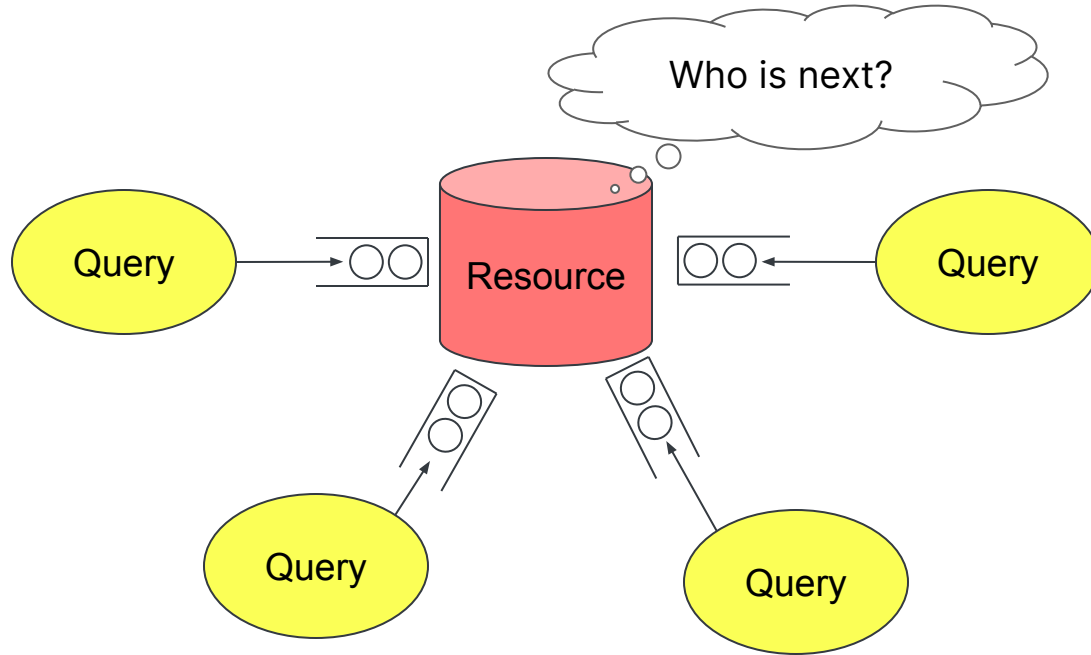
What is workload scheduling?



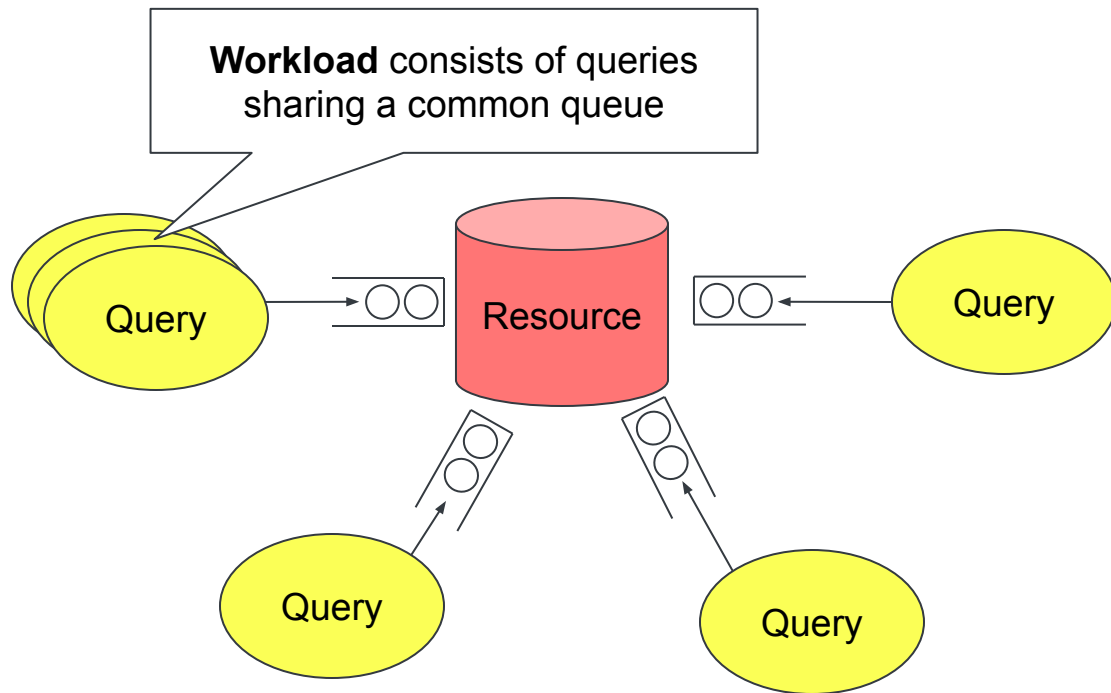
What is workload scheduling?



What is workload scheduling?



What is workload scheduling?



Features

Control over resource distribution between concurrently executing queries:

- Workload markup
- Never fail queries
- Bandwidth limits
- Max-min fair bandwidth sharing
- Priorities
- Hierarchical scheduling
- Introspection through `system.scheduler` table

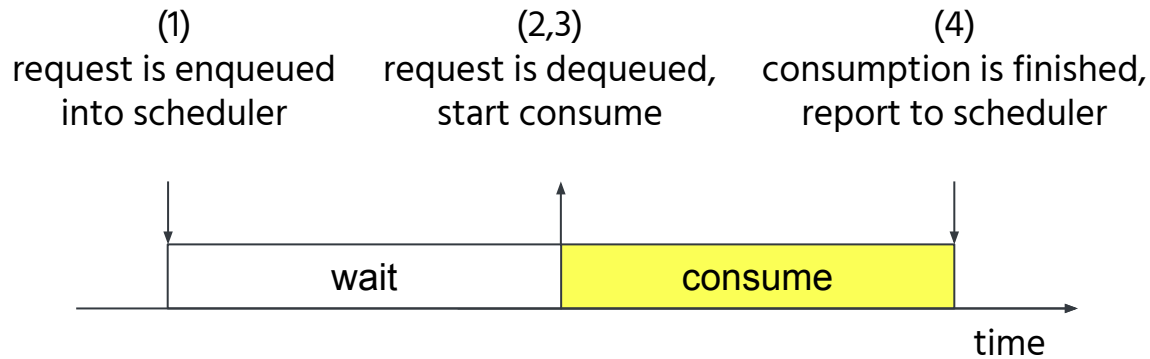
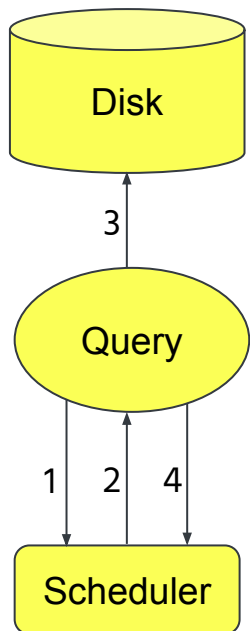


Caveats

- Experimental
- No SQL support for configuring
- Currently implemented only for **remote IO** (s3, hdfs, azure)
- Will be implemented for local disks and CPUs



Resource Requests



How to configure?

How to configure? [1/4]

Mark up queries with special **workload** setting:

```
SELECT count() FROM my_table WHERE value = 42 SETTINGS workload = 'production'  
SELECT count() FROM my_table WHERE value = 13 SETTINGS workload = 'development'
```

<https://clickhouse.com/docs/en/operations/workload-scheduling>



How to configure? [2/4]

Specify **resources** to be used by remote disks:

```
<clickhouse>
  <storage_configuration>
    <disks>
      <s3>
        <type>s3</type>
        <endpoint>https://end.point/bucket/path</endpoint>
        <read_resource>network_read</read_resource>
        <write_resource>network_write</write_resource>
      </s3>
    </disks>
  </storage_configuration>
</clickhouse>
```



How to configure? [3/4]

Describe **scheduling nodes** for every resource:

```
<clickhouse>
  <resources>
    <network_read>
      <node path="/">
        <type>inflight_limit</type><max_requests>100</max_requests></node>
      <node path="/fair">
        <type>fair</type></node>
      <node path="/fair/prod">
        <type>fifo</type><weight>3</weight></node>
      <node path="/fair/dev">
        <type>fifo</type></node>
    </network_read>
  </resources>
</clickhouse>
```


How to configure? [4/4]

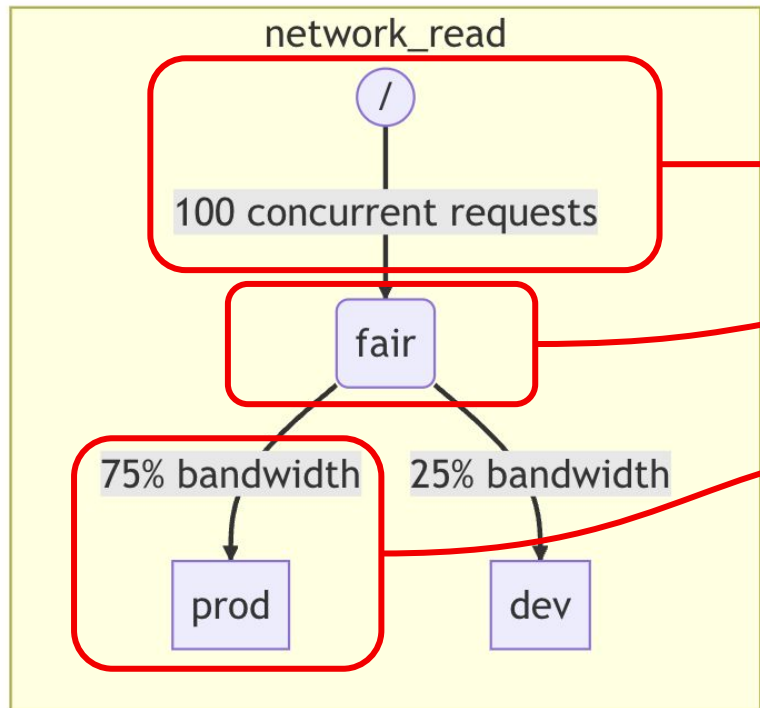
Classify workloads into queues for every resource:

```
<clickhouse>
  <workload_classifiers>
    <production>
      <network_read>/fair/prod</network_read>
      <network_write>/fair/prod</network_write>
    </production>
    <development>
      <network_read>/fair/dev</network_read>
      <network_write>/fair/dev</network_write>
    </development>
  </workload_classifiers>
</clickhouse>
```



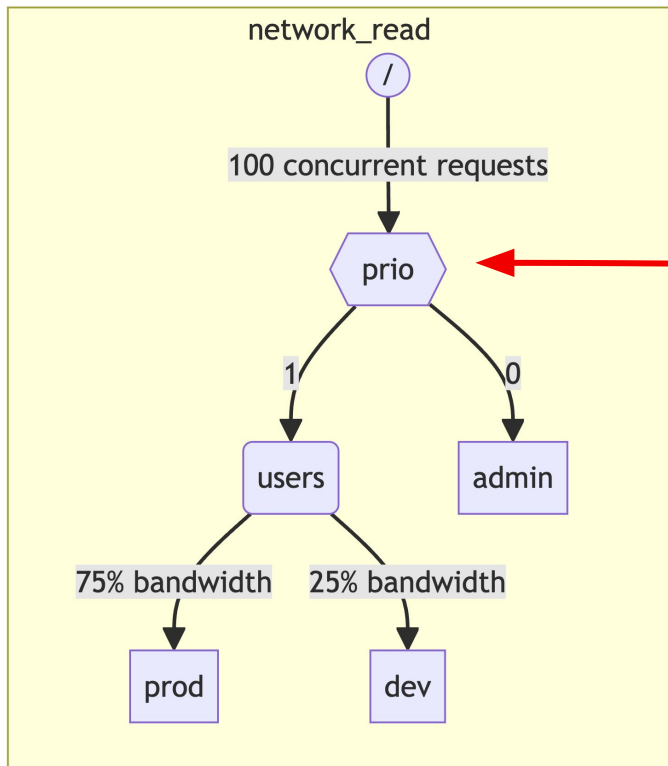
How to apply in practice?

Fair bandwidth sharing



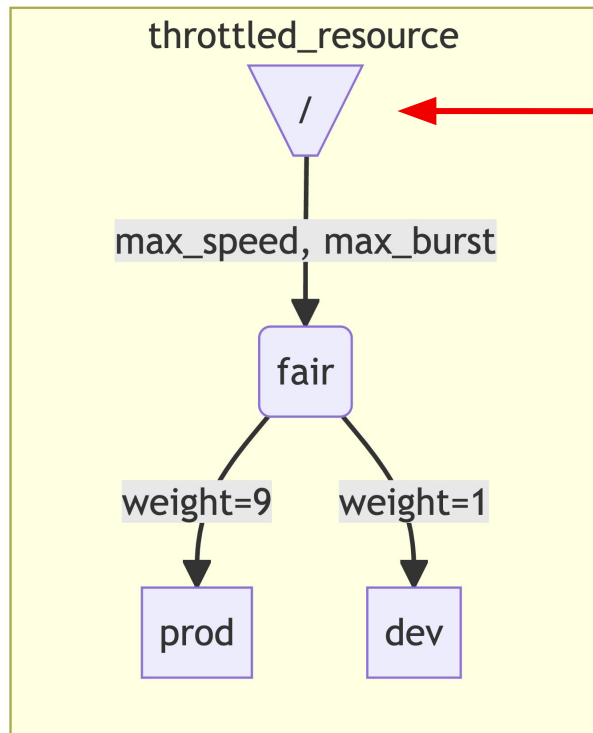
```
<network_read>  
  <node path="/">  
    <type>inflight_limit</type>  
    <max_requests>100</max_requests>  
  </node>  
  <node path="/fair">  
    <type>fair</type>  
  </node>  
  <node path="/fair/prod">  
    <type>fifo</type>  
    <weight>3</weight>  
  </node>  
  <node path="/fair/dev">  
    <type>fifo</type>  
  </node>  
</network_read>
```

Static priorities

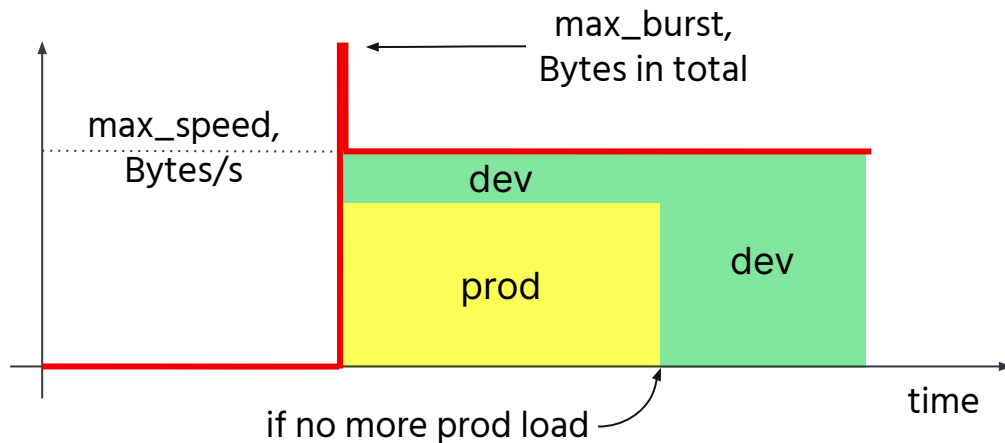


```
<node path="/prio">  
  <type>priority</type>  
</node>  
<node path="/prio/users">  
  <type>fair</type>  
  <priority>1</priority>  
</node>  
<node path="/prio/admin">  
  <type>fifo</type>  
  <priority>0</priority>  
</node>
```

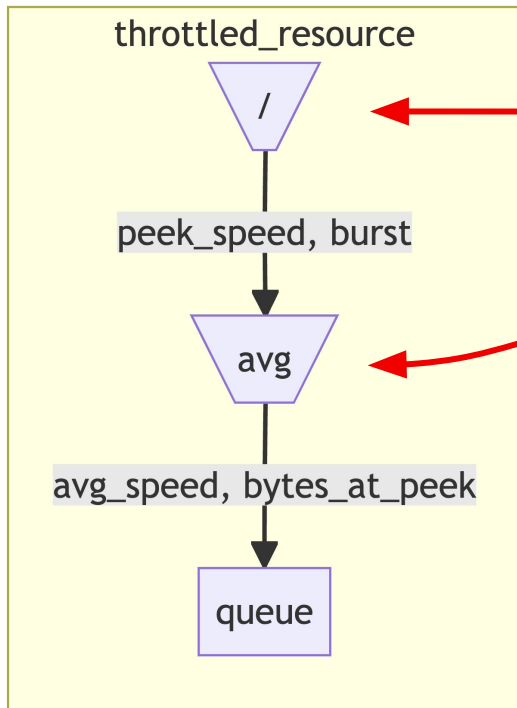
IO Throttling: Works with Fairness



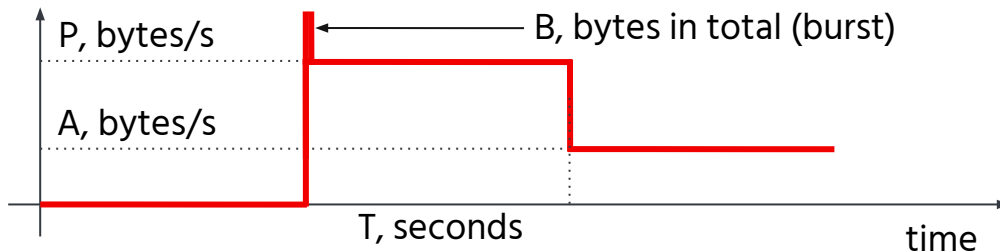
```
<type>bandwidth_limit</type>
<max_speed>100500</max_speed>  <!-- Bytes/s -->
<max_burst>100500</max_burst>  <!-- Bytes -->
```



IO Throttling: Peek Limit and Avg Limit



```
<node path="/">
  <type>bandwidth_limit</type>
  <max_speed>P</max_speed>      <!-- Bytes/s -->
  <max_burst>B</max_burst>      <!-- Bytes -->
</node>
<node path="/avg">
  <type>bandwidth_limit</type>
  <max_speed>A</max_speed>      <!-- Bytes/s -->
  <max_burst>P * T</max_burst>  <!-- Bytes -->
</node>
```



Thank you!
Questions?