# CS60038
# Advances in Operating System Design
# **Assignment 1 [Part A]**

---

Rajas Bhatt 19CS30037

---

## Removing NUMA memory allocation, scheduler and emulation

**Default**: Included

### Differences in the Config Files

Configuration files used are **`config-5.10.191`** and **`config-5.10.191non-numa`**. It can be seen that in the file with NUMA enabled (**`config-5.10.191`**), **`CONFIG_NUMA`** is true, while in the other file it is not set.

```
rajas@rajas:~/Desktop/configs$ diff -y config-5.10.191 config-5.10.191non-numa | grep '[<>|]'
CONFIG_LOCALVERSION=""                                  | CONFIG_LOCALVERSION="non-numa"
CONFIG_NUMA_BALANCING=y                                 <
CONFIG_NUMA_BALANCING_DEFAULT_ENABLED=y                 <
CONFIG_X86_NUMACHIP=y                                   <
CONFIG_X86_UV=y                                         <
CONFIG_NUMA=y                                           | # CONFIG_NUMA is not set
CONFIG_AMD_NUMA=y                                       <
CONFIG_X86_64_ACPI_NUMA=y                               <
CONFIG_NUMA_EMU=y                                       <
CONFIG_NODES_SHIFT=10                                   <
CONFIG_USE_PERCPU_NUMA_NODE_ID=y                        <
CONFIG_ACPI_NUMA=y                                      <
CONFIG_ACPI_HMAT=y                                      <
CONFIG_EFI_SOFT_RESERVE=y                               <
CONFIG_NEED_MULTIPLE_NODES=y                            <
CONFIG_NUMA_KEEP_MEMINFO=y                              <
CONFIG_HMEM_REPORTING=y                                 <
CONFIG_SGI_XP=m                                         <
CONFIG_SGI_GRU=m                                        <
# CONFIG_SGI_GRU_DEBUG is not set                       <
CONFIG_UV_MMTIMER=m                                     <
CONFIG_DEV_DAX_HMEM=m                                   <
CONFIG_DEV_DAX_HMEM_DEVICES=y                           <
```

### NUMA Detection

In kernel 5.10.191, the output of the **`numactl --hardware`** command is given as follows:

```
rajas@rajas:~/Desktop$ uname -r
5.10.191
rajas@rajas:~/Desktop$ numactl --hardware
available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 7921 MB
node 0 free: 6018 MB
node distances:
node   0
  0:  10
```

In the customized kernel with NUMA disabled, the output of the **`numactl --hardware`** command is given as follows:

```
rajas@rajas:~$ uname -r
5.10.191non-numa
rajas@rajas:~$ numactl --hardware
No NUMA available on this system
```

It can be seen that NUMA is successfully disabled in the customized kernel.

### Performance Impact due to NUMA

Under normal usage, the impact of NUMA vs UMA is difficult to measure. More importantly, having one NUMA node is not different from having no NUMA nodes at all. In systems which have more than one hardware configured NUMA node, then memory allocated to the CPU-affiliated NUMA node will provide smaller AMAT compared to the node which does not contain the CPU.

## Removing Kyber I/O Scheduler

**Default**: Included as Kernel Module

### Differences in the Config Files

Configuration files used are **`config-5.10.191`** and **`config-5.10.191non-kyber`**. It can be seen that in the file with NUMA enabled (**`config-5.10.191`**), **`CONFIG_MQ_IOSCHED_KYBER`** is set to **`m`** (included as kernel module), while in the other file it is not set.

```
rajas@rajas:~/Desktop/configs$ diff -y config-5.10.191 config-5.10.191non-kyber | grep '[<>|]'
CONFIG_LOCALVERSION=""                                  | CONFIG_LOCALVERSION="non-kyber"
CONFIG_NUMA_BALANCING=y                                 | # CONFIG_NUMA_BALANCING is not set
CONFIG_NUMA_BALANCING_DEFAULT_ENABLED=y                 <
CONFIG_X86_NUMACHIP=y                                   | # CONFIG_X86_NUMACHIP is not set
CONFIG_X86_UV=y                                         | # CONFIG_X86_UV is not set
CONFIG_ACPI_HMAT=y                                      | # CONFIG_ACPI_HMAT is not set
CONFIG_EFI_SOFT_RESERVE=y                               <
CONFIG_MQ_IOSCHED_KYBER=m                               | # CONFIG_MQ_IOSCHED_KYBER is not set
CONFIG_HMEM_REPORTING=y                                 <
CONFIG_SGI_XP=m                                         <
CONFIG_SGI_GRU=m                                        <
# CONFIG_SGI_GRU_DEBUG is not set                       <
CONFIG_UV_MMTIMER=m                                     <
CONFIG_DEV_DAX_HMEM=m                                   <
CONFIG_DEV_DAX_HMEM_DEVICES=y                           <
```

### Detection of Kyber I/O Scheduler

In kernel 5.10.191, the Kyber I/O Scheduler is present, **`kyber-iosched.ko`** can be seen.

```
rajas@rajas:~$ uname -r
5.10.191
rajas@rajas:~$ cd /lib/modules/5.10.191/kernel/block/
rajas@rajas:/lib/modules/5.10.191/kernel/block$ ls
bfq.ko  kyber-iosched.ko
```

In the customized kernel, with Kyber I/O Scheduler disabled (**`5.10.191non-kyber`**), **`kyber-iosched.ko`** cannot be seen in **`/lib/modules/5.10.191non-kyber/kernel/block/`**. Instead, only BFQ (Budget Fair Queueing) I/O Scheduler is visible.

```
rajas@rajas:~$ uname -r
5.10.191non-kyber
rajas@rajas:~$ cd /lib/modules/5.10.191non-kyber/kernel/block/
rajas@rajas:/lib/modules/5.10.191non-kyber/kernel/block$ ls
bfq.ko
```

## Usage of the Kyber I/O Scheduler

### Enabling Kyber I/O Scheduler

To check the default scheduler, we do **`$ cat /sys/block/sda/queue/scheduler.`** It can be seen that **mq-deadline** is the default scheduler.

```
rajas@rajas:~$ cat /sys/block/sda/queue/scheduler
[mq-deadline] none
```

It was noticed that the default scheduler is set to mq-deadline on both the kernel versions (even in the one with the **Kyber** scheduler). We change the default scheduler to **Kyber**. This does not require a reboot since **Kyber** exists as a Kernel Module.

```
root@rajas:/home/rajas# echo kyber > /sys/block/sda/queue/scheduler
root@rajas:/home/rajas# cat /sys/block/sda/queue/scheduler
mq-deadline [kyber] none
```

### Benchmarking Kyber I/O Scheduler

**Kyber** is designed to be a low latency scheduler, therefore its results compared to **mq-deadline** were better as far as latency was concerned. However, **bfq** seemed to outperform **Kyber** in terms of latency. Shown below are the latency measurements on the **`fio`** tool using the command:

**`sudo fio --name=fiotest --ioengine=libaio --size 1Gb --rw=randwrite --bs=1M --direct=1 --numjobs=4 --iodepth=8 --runtime=60 --group_reporting`**

| Read Write Test (1GB x 4) Latency | | | |
|---|---|---|---|
| | Kyber (ms) | mq-deadline (ms) | bfq(ms) |
| 1 | 64.68 | 69.26 | 43.33 |
| 2 | 65.9 | 62.29 | 39.03 |
| 3 | 65.45 | 80.41 | 41.81 |
| 4 | 57.4 | 68.2 | 43.87 |
| 5 | 66.01 | 68.03 | 49.28 |
| 6 | 62.65 | 65.5 | 41.37 |
| 7 | 66.95 | 67.72 | 44.76 |
| 8 | 61.85 | 65.24 | 54.41 |
| 9 | 63.85 | 71 | 49.12 |
| 10 | 62 | 67.14 | 46.63 |
| | **63.674** | **68.479** | **45.361** |

In terms of read and write speeds, **bfq** showed the highest speeds, whereas those of **Kyber** and **mq-deadline** were similar.
Under normal usage of the computer, no major differences in responsiveness were observed.

## Including multipath TCP

**Default**: Included
Since multipath TCP is included by default in the kernel, we compile, build and install another kernel **`5.10.191non-multipath`** where multipath TCP has been disabled.

## Differences in the Config Files

Configuration files used are **`config-5.10.191`** and **`config-5.10.191non-multipath`**. It can be seen that in the file with NUMA enabled (**`config-5.10.191`**), **`CONFIG_MPTCP`** is set, while in the other file it is not set. Notice that disabling MPTCP also removes a few other flags such as **`CONFIG_INET_MPTCP_DIAG`** and **`CONFIG_MPTCP_IPV6`**.

```
rajas@rajas:~/Desktop/configs$ diff -y config-5.10.191 config-5.10.191non-multipath | grep '[<>|]'
CONFIG_LOCALVERSION=""                                  | CONFIG_LOCALVERSION="non-multipath"
CONFIG_NUMA_BALANCING=y                                 | # CONFIG_NUMA_BALANCING is not set
CONFIG_NUMA_BALANCING_DEFAULT_ENABLED=y                 <
CONFIG_X86_NUMACHIP=y                                   | # CONFIG_X86_NUMACHIP is not set
CONFIG_X86_UV=y                                         | # CONFIG_X86_UV is not set
CONFIG_ACPI_HMAT=y                                      | # CONFIG_ACPI_HMAT is not set
CONFIG_EFI_SOFT_RESERVE=y                               <
CONFIG_MPTCP=y                                          | # CONFIG_MPTCP is not set
CONFIG_INET_MPTCP_DIAG=m                                <
CONFIG_MPTCP_IPV6=y                                     <
CONFIG_HMEM_REPORTING=y                                 <
CONFIG_SGI_XP=m                                         <
CONFIG_SGI_GRU=m                                        <
# CONFIG_SGI_GRU_DEBUG is not set                       <
CONFIG_UV_MMTIMER=m                                     <
CONFIG_DEV_DAX_HMEM=m                                   <
CONFIG_DEV_DAX_HMEM_DEVICES=y                           <
```

## Detection of multipath TCP version

In Kernel **`5.10.191`**, multipath TCP is present since there are hash table entries for the MPTCP token.

```
rajas@rajas:~$ uname -r
5.10.191
rajas@rajas:~$ dmesg | grep MPTCP
[    0.726617] MPTCP token hash table entries: 8192 (order: 5, 196608 bytes, linear)
rajas@rajas:~$ sudo sysctl -a | grep mptcp.enabled
[sudo] password for rajas:
net.mptcp.enabled = 1
```

In Kernel **`5.10.191non-multipath`**, no hash table entries exist for the MPTCP token, hence multipath-TCP is disabled.

```
rajas@rajas:~$ uname -r
5.10.191non-multipath
rajas@rajas:~$ sudo sysctl -a | grep mptcp.enabled
[sudo] password for rajas:
rajas@rajas:~$ dmesg | grep MPTCP
rajas@rajas:~$
```

## Usage of multipath TCP

### Is MPTCP actually enabled?

The design of MPTCP has been heavily influenced by the middleboxes that have been deployed in a wide range of networks, notably in cellular and enterprise networks. On the IITKGP network, the command **`curl http://www.multipath-tcp.org`** gives no output.
Ideally, there must be an output confirming whether MPTCP is available or not. This may be due to internet restrictions.

```
rajas@rajas:~$ curl http://www.multipath-tcp.org --output -
rajas@rajas:~$
```