

# Domain Sockets on Steroids

Bypassing the Kernel using Shared Memory

Peter Goldsborough  
Alexander van Renen  
Viktor Leis

July 28, 2016

# Motivation

# Motivation

Everyone is using domain sockets

# Motivation

Everyone is using domain sockets

PostGres      MySQL      HyPer

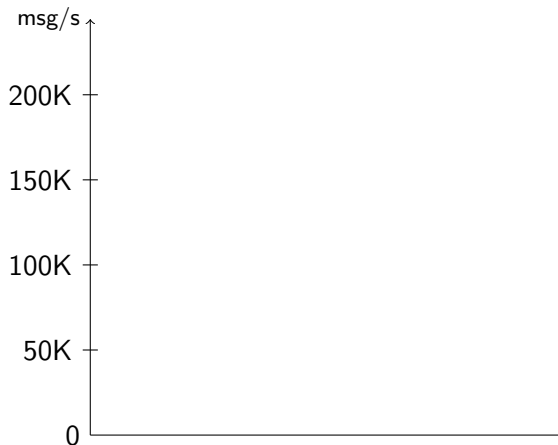
# Motivation

Everyone is using domain sockets

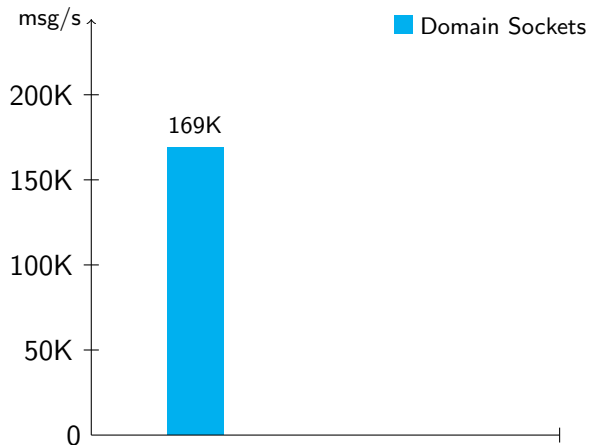
PostGres      MySQL      HyPer

So, obviously, domain sockets are the best?

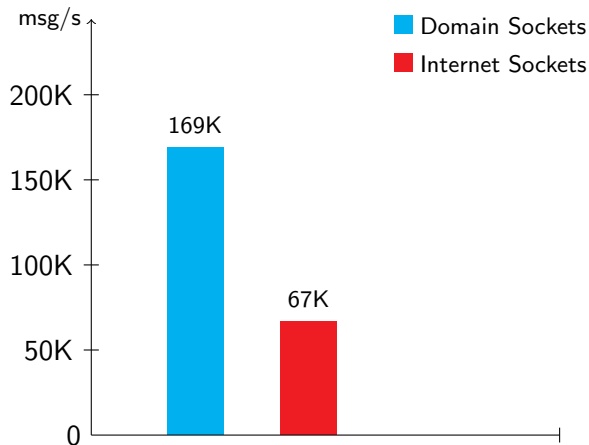
# Comparing IPC Methods



# Comparing IPC Methods

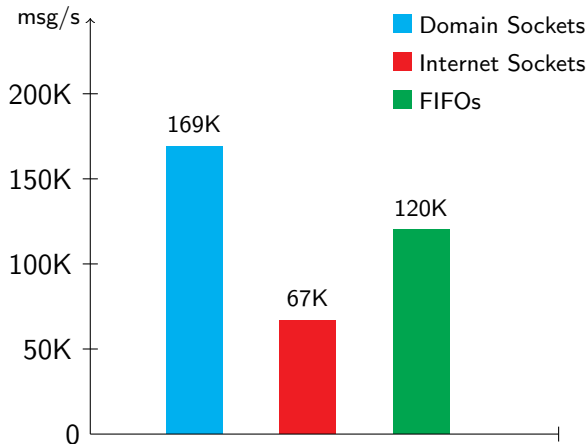


# Comparing IPC Methods

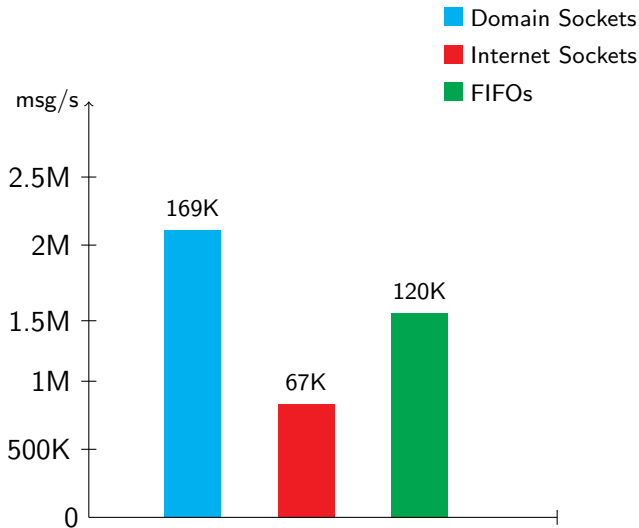




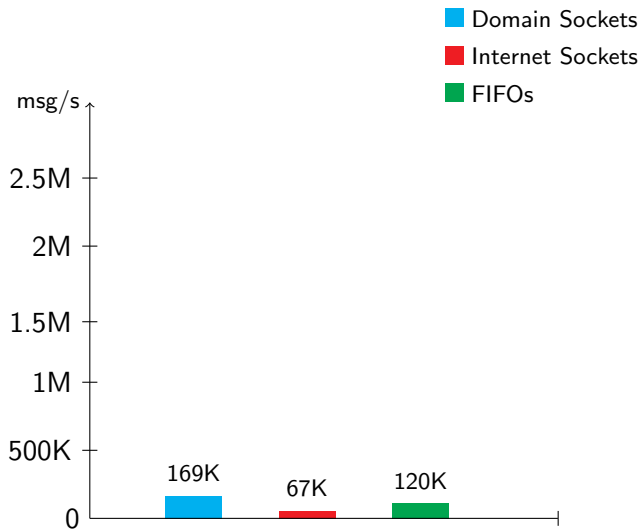
# Comparing IPC Methods



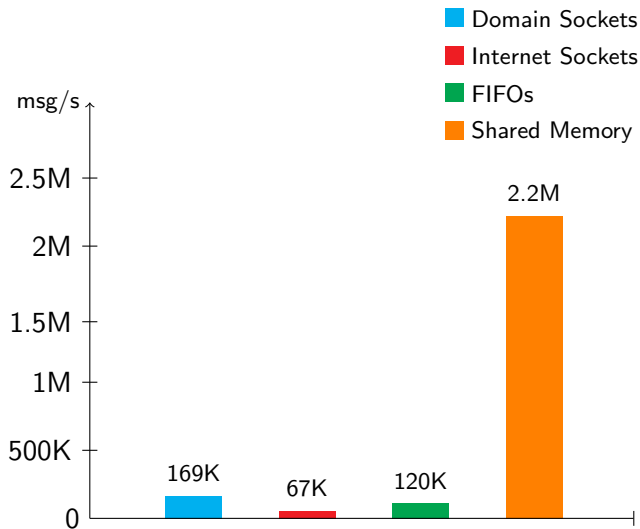
# Comparing IPC Methods



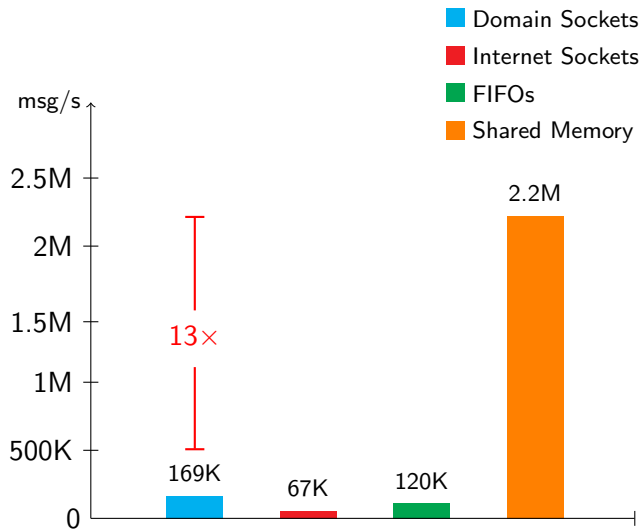
# Comparing IPC Methods



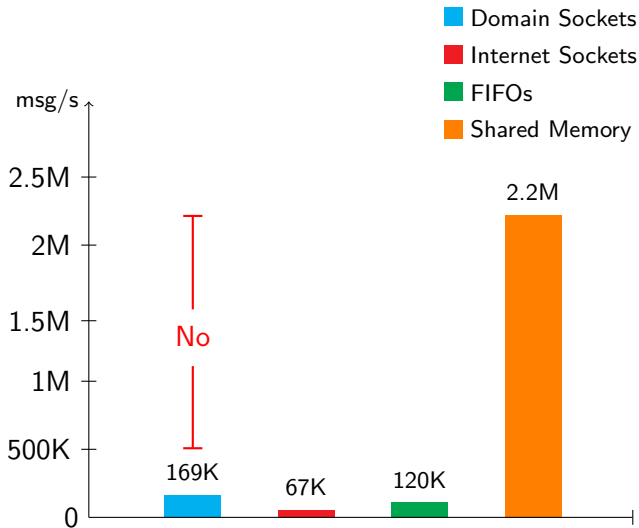
# Comparing IPC Methods



# Comparing IPC Methods



# Comparing IPC Methods



# Basic Idea

# Basic Idea

1. Put on mad scientist hat



# Basic Idea

1. Put on mad scientist hat
2. Overwrite syscalls

```
socket()  
accept()  
connect()
```

# Basic Idea

1. Put on mad scientist hat
2. Overwrite syscalls

```
socket()  
accept()  
connect()
```

3. Create shared memory channel in the background

# Basic Idea

1. Put on mad scientist hat
2. Overwrite syscalls

```
socket()  
accept()  
connect()
```

3. Create shared memory channel in the background
4. Don't recompile a single file

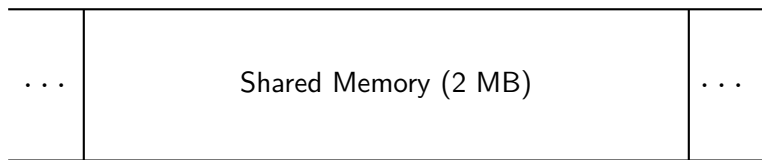
```
LD_PRELOAD=/path/to/our/library ./hyper
```

# Implementation

(1) `connect()`  $\longleftrightarrow$  `accept()`

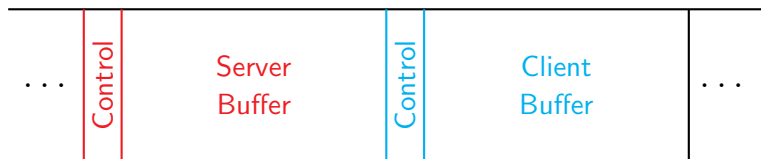
# Implementation

(1) `connect()`  $\longleftrightarrow$  `accept()`



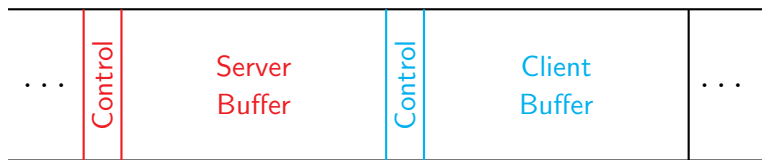
# Implementation

(1) `connect()`  $\longleftrightarrow$  `accept()`



# Implementation

(2) Client write(500 KB)



# Implementation

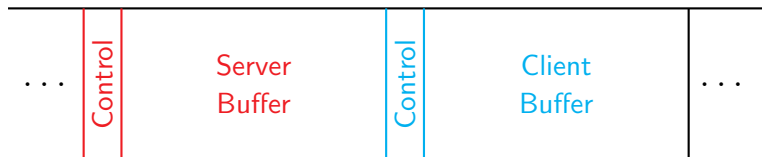
## (2) Client write(500 KB)

```
int write(int fd, void* buffer, int length) {  
    if (using_our_library(fd)) {  
        connection = lookup(fd);  
        return buffer_write(connection, buffer, length);  
    } else {  
        return real_write(fd, buffer, length);  
    }  
}
```



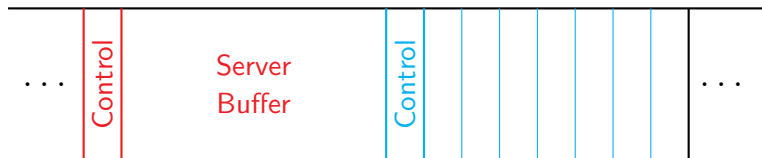
# Implementation

(2) Client write(500 KB)



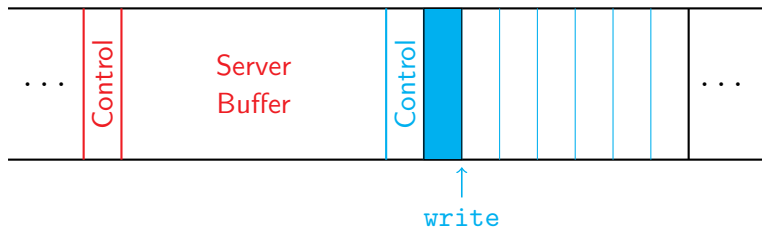
# Implementation

(2) Client write(500 KB)



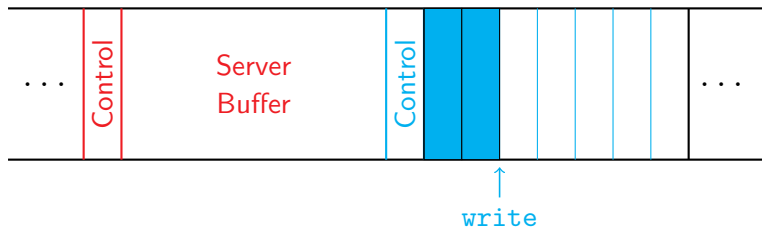
# Implementation

(2) Client write(500 KB)



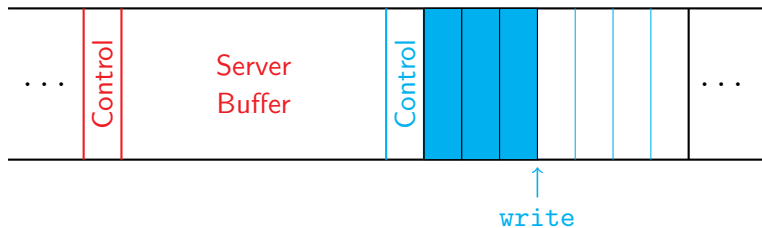
# Implementation

(2) Client write(500 KB)



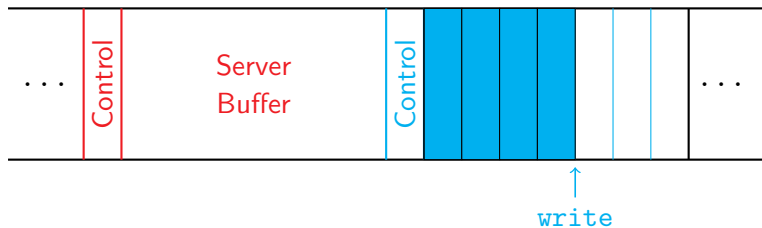
# Implementation

(2) Client write(500 KB)



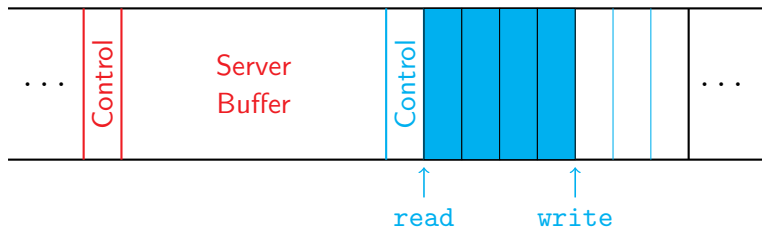
# Implementation

(2) Client write(500 KB)



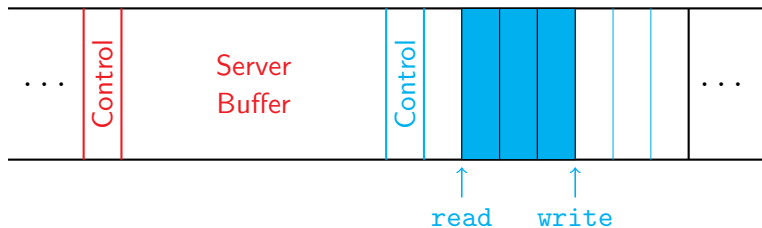
# Implementation

(2) Server read(250 KB)



# Implementation

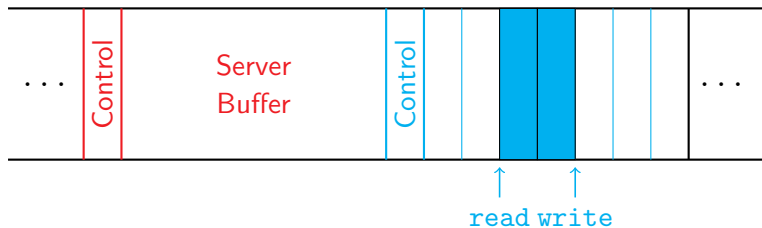
(2) Server read(250 KB)





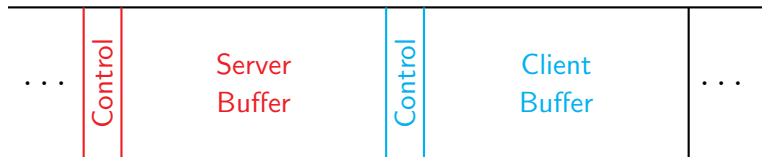
# Implementation

(2) Server read(250 KB)



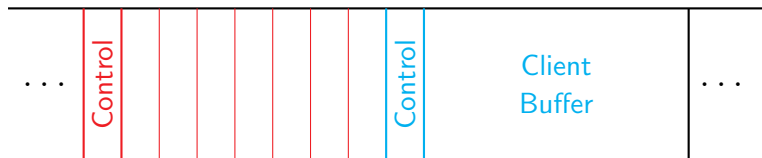
# Implementation

(4) Server write(1.2 MB)



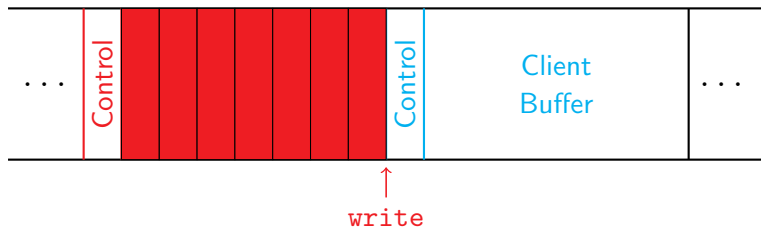
# Implementation

(4) Server write(1.2 MB)



# Implementation

(4) Server write(1.2 MB)



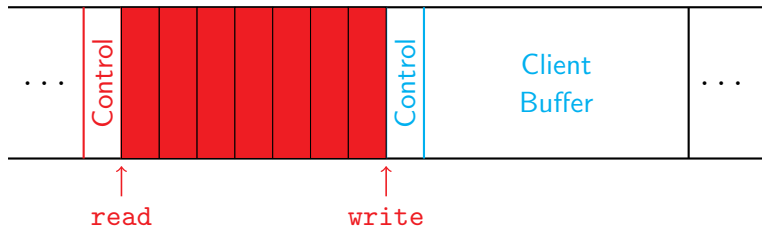
# Implementation

## (4) Server write(1.2 MB)

```
start = now(); // asm("rdtsc")
while(not enough space) {
    switch(now() - start) {
        case elapsed_time < LEVEL_ONE: asm("pause");
        case elapsed_time < LEVEL_TWO: sched_yield();
        case elapsed_time < TIMEOUT: usleep(1);
        default: return TIMEOUT_ERROR;
    }
}
```

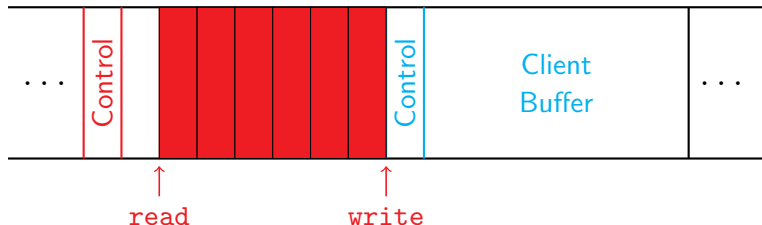
# Implementation

(5) Client read(200 KB)



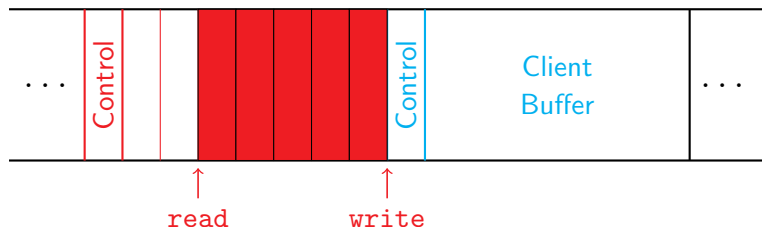
# Implementation

(5) Client read(200 KB)



# Implementation

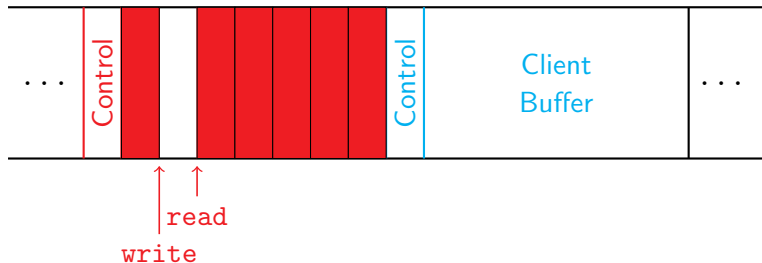
(5) Client read(200 KB)





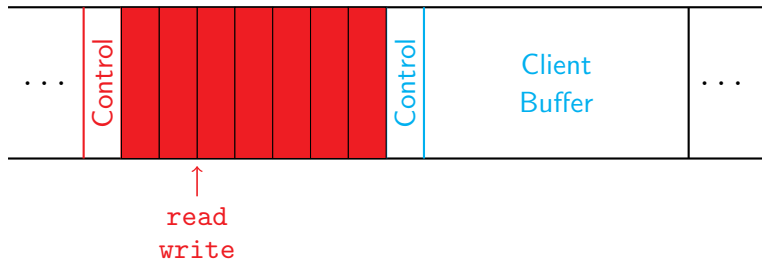
# Implementation

(5) Client read(200 KB)



# Implementation

(5) Client read(200 KB)



## File Descriptors or Keys?

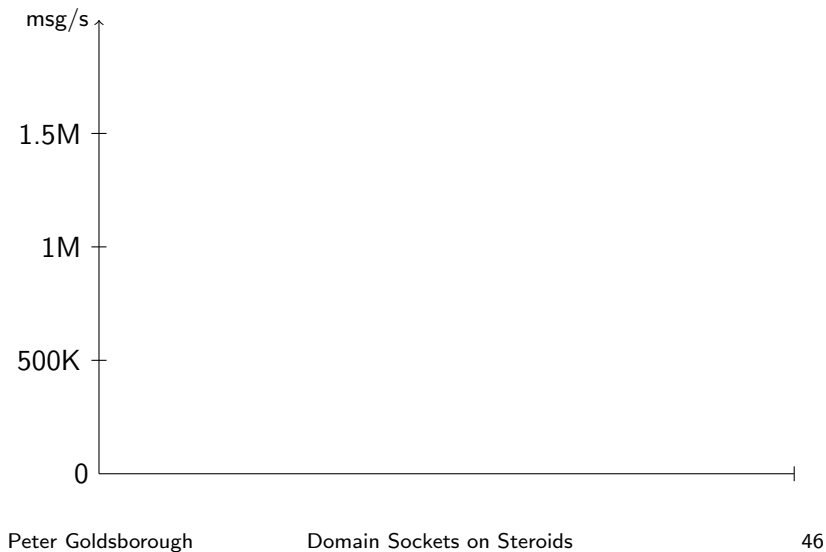
# Problems

`select()` and `poll()`

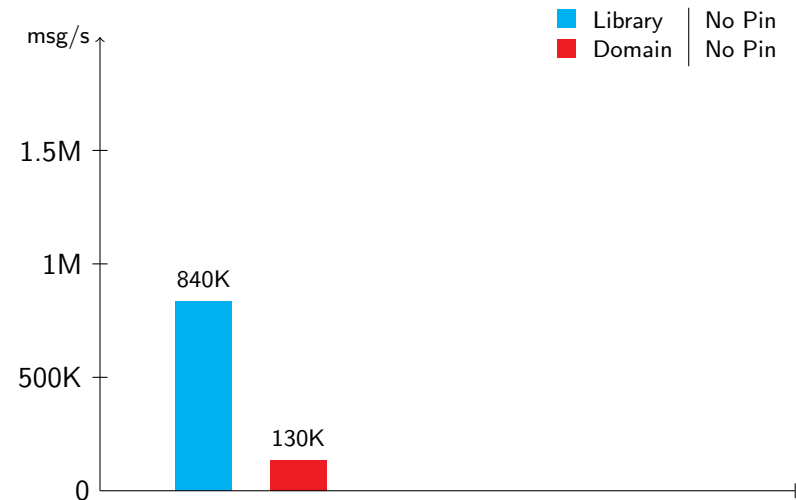
# Problems

`fork()` and `kill()`

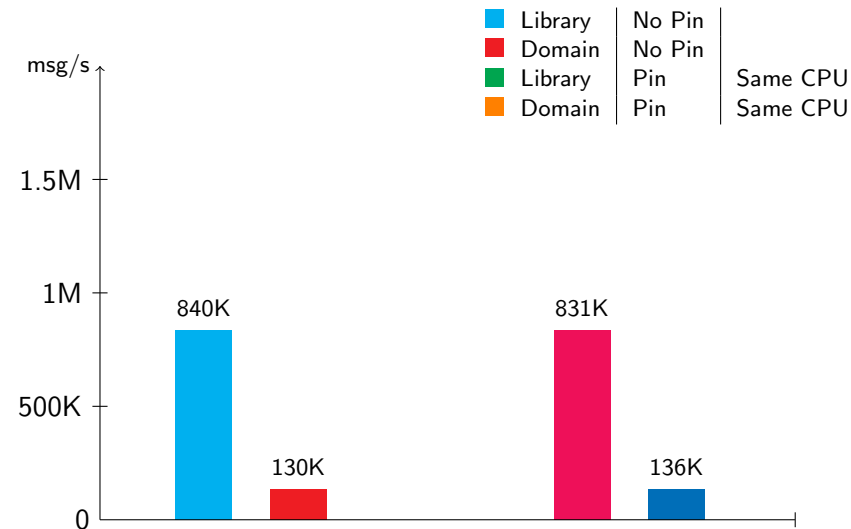
# Results



# Results

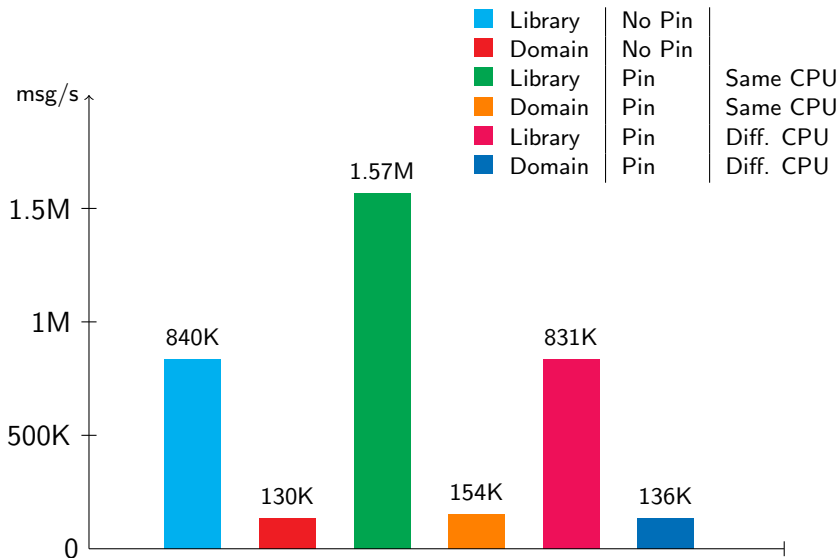


# Results

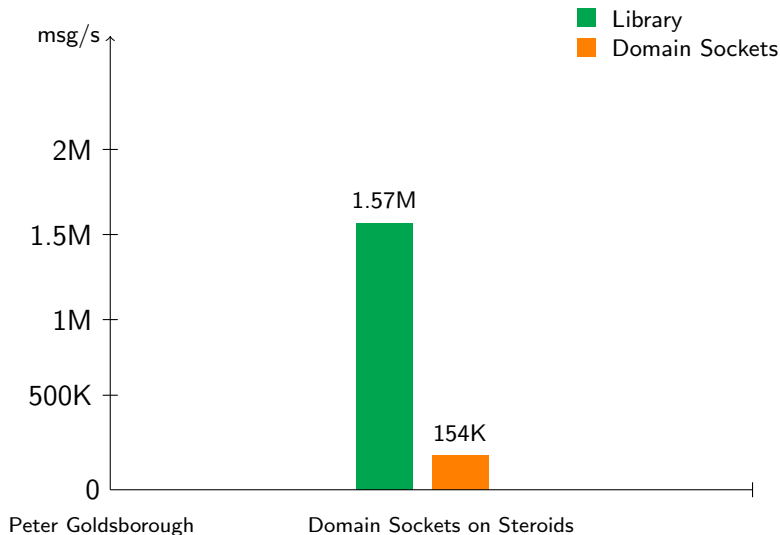




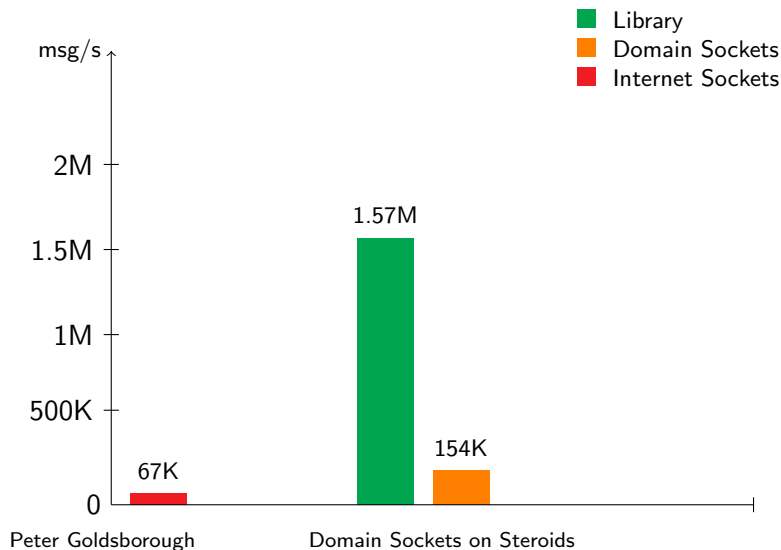
# Results



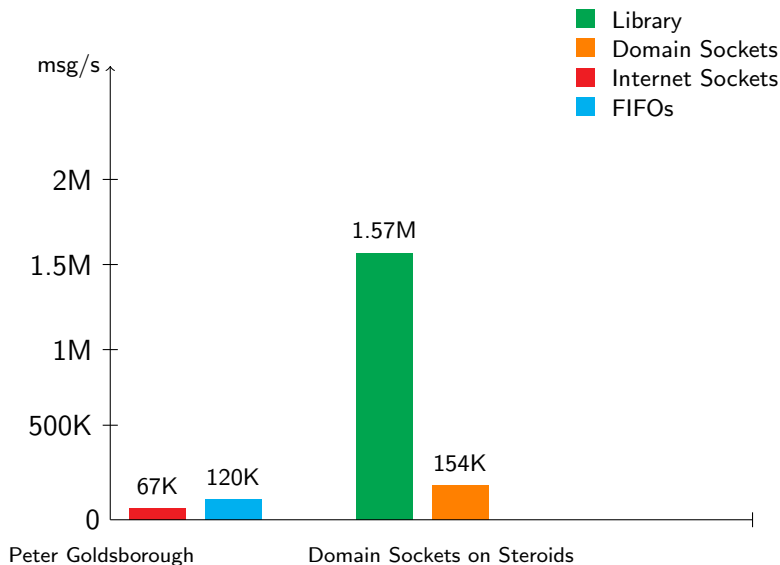
# Where does this leave us?



# Where does this leave us?

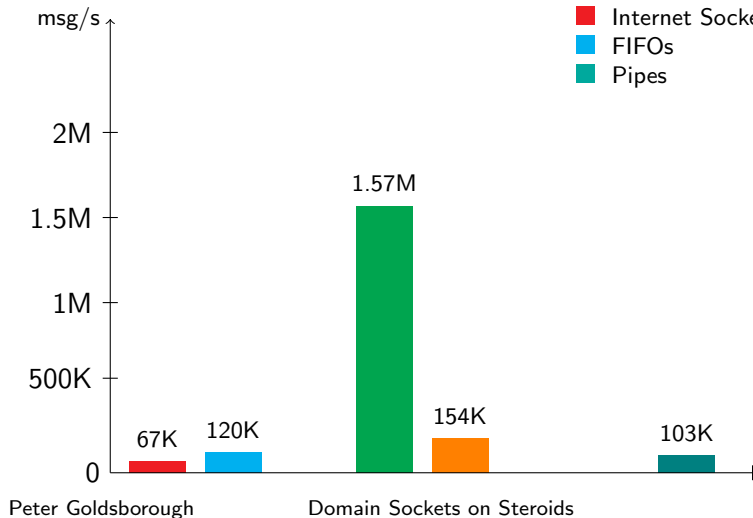


# Where does this leave us?

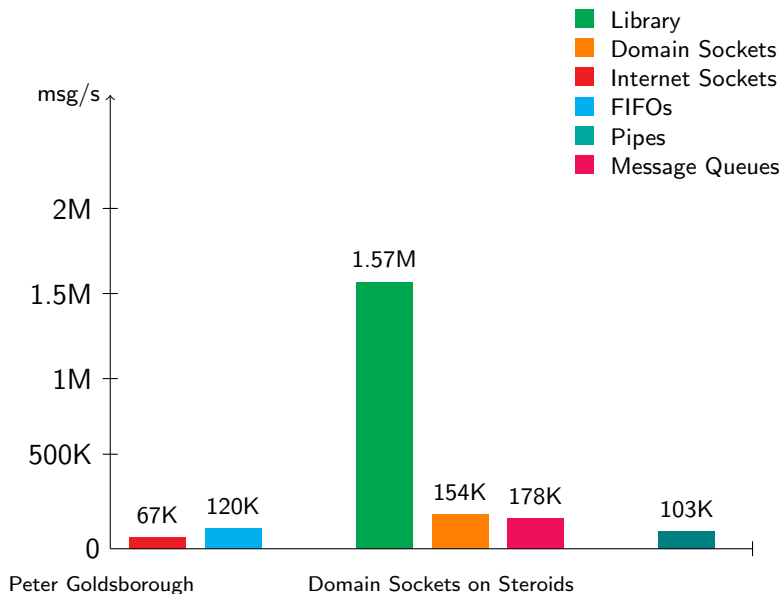


# Where does this leave us?

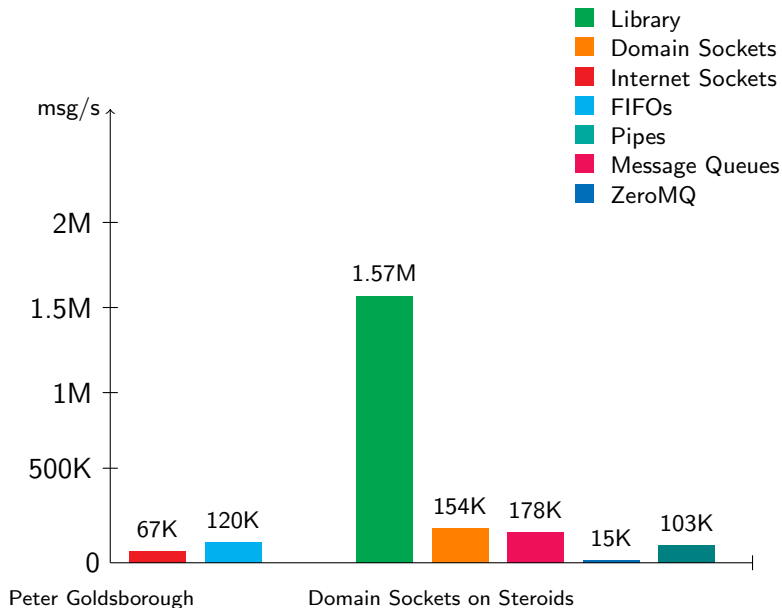
- Library
- Domain Sockets
- Internet Sockets
- FIFOs
- Pipes



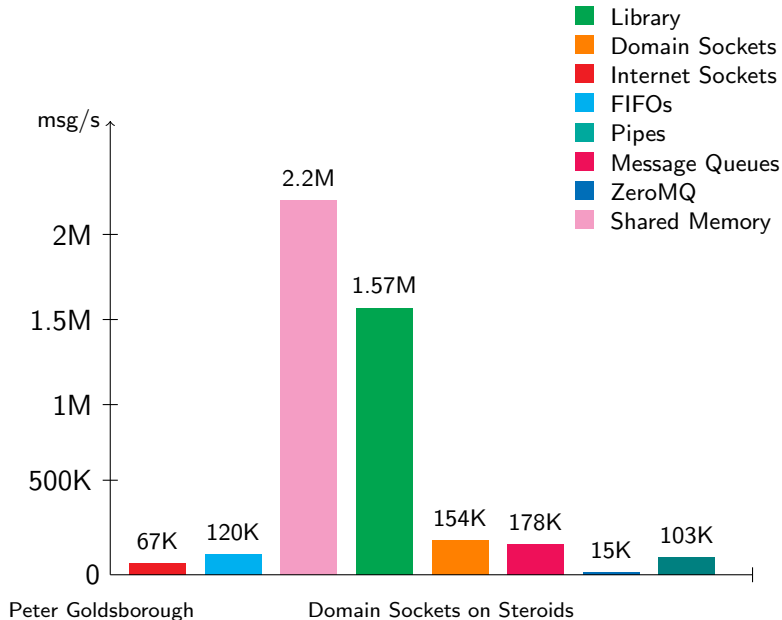
# Where does this leave us?



# Where does this leave us?



# Where does this leave us?



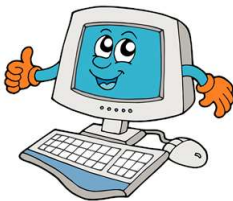


# PostGres

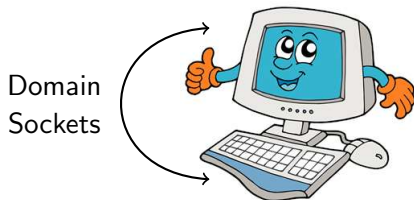


# Outlook

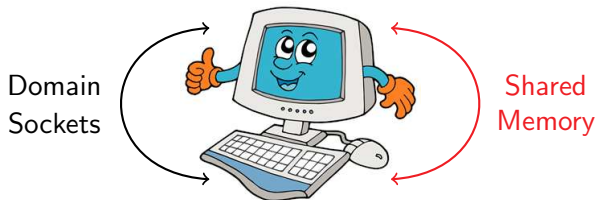
# Outlook



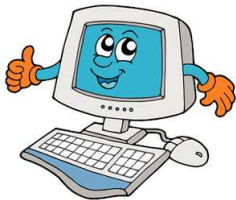
# Outlook



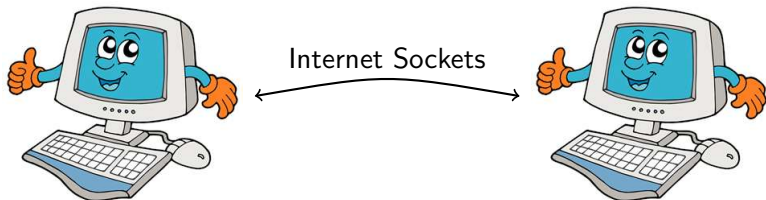
# Outlook



# Outlook

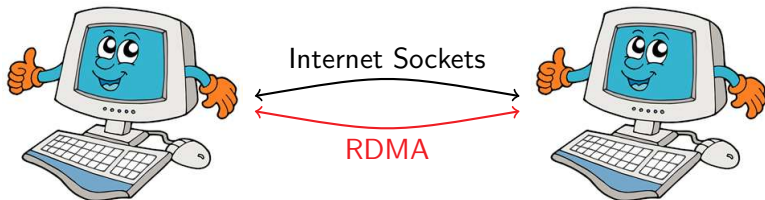


# Outlook





# Outlook



# Q & A