

# QoS-Aware Virtualization

Christopher Feener  
Advisor: Xuechen Zhang



## Motivations

- Want faster system with limited swapping.

## Challenges

- Speed of SSD is fast, but **not durable**.
- Disk is durable, but **slow**.
- Last attempt of “Hybrid Swapping” was done on Linux kernel 2.6, and many parts of the kernel have since changed.
- Lack of documentation.

## Our Approach

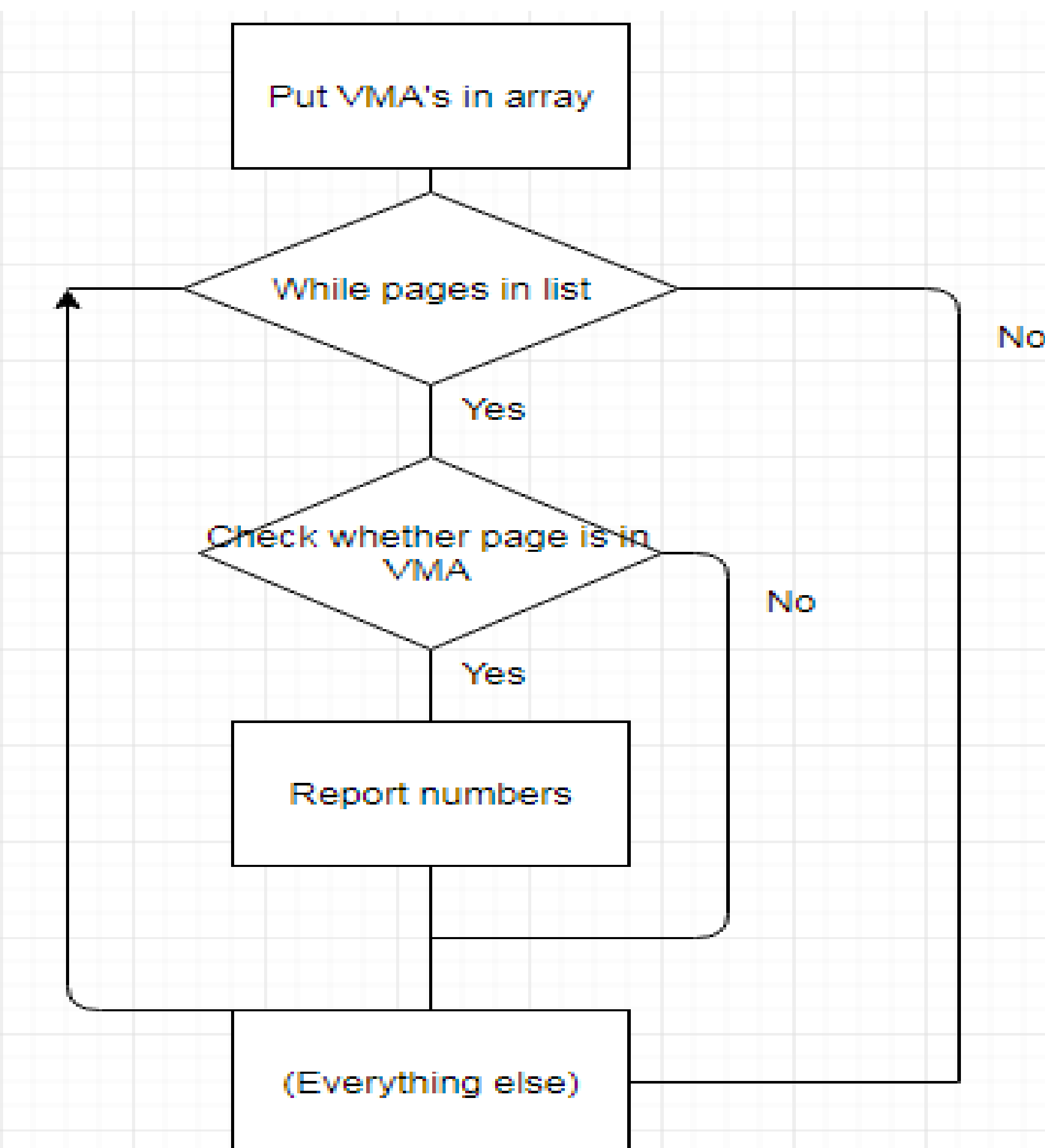
- “Hybrid Swapping” could use the advantages of both speed and limited swapping. It did this by swapping only page faults to disk.
- Only minor additions were made to linux kernel 4.11, specifically in the function *shrink\_page\_list()*.
- Debugging: No major program/method was used other than dmesg and printk statements.

## Results and Data

```
QEMU [ 48.071678] Page fffffea0000e7d580 is in VMA 15 (4 currently)
[ 48.071828] Page fffffea0000e7d540 is in VMA 15 (5 currently)
[ 48.071981] Page fffffea0000e7d500 is in VMA 15 (6 currently)
[ 48.072446] Page fffffea0000e7d4c0 is in VMA 15 (1 currently)
[ 48.072614] Page fffffea0000e7d480 is in VMA 15 (2 currently)
[ 48.072765] Page fffffea0000e7d440 is in VMA 15 (3 currently)
[ 48.072913] Page fffffea0000e7df80 is in VMA 14 (1 currently)
[ 48.073063] Page fffffea0000e7df40 is in VMA 14 (2 currently)
[ 48.074135] Page fffffea0000e7df00 is in VMA 14 (3 currently)
[ 48.074286] Page fffffea0000e8cc0 is in VMA 14 (4 currently)
[ 48.074470] Page fffffea0000ec9280 is in VMA 14 (5 currently)
[ 48.074624] Page fffffea0000e8b40 is in VMA 14 (6 currently)
[ 48.074802] Page fffffea0000eed840 is in VMA 14 (7 currently)
[ 48.075012] Page fffffea0000eed780 is in VMA 14 (8 currently)
[ 48.075204] Page fffffea0000f03a00 is in VMA 14 (9 currently)
[ 48.076699] Page fffffea0000ef51c0 is in VMA 14 (10 currently)
[ 48.077081] Page fffffea0000e7a000 is in VMA 19 (1 currently)
[ 48.078413] Page fffffea0000e7bc80 is in VMA 19 (2 currently)
[ 48.078717] Page fffffea0000e780c0 is in VMA 19 (3 currently)
[ 48.079004] Page fffffea0000f10580 is in VMA 19 (4 currently)
[ 48.079290] Page fffffea0000e7a100 is in VMA 19 (5 currently)
[ 48.080893] Page fffffea0000e7aa80 is in VMA 19 (6 currently)
[ 48.081384] Page fffffea0000e7bcc0 is in VMA 19 (7 currently)
[ 48.081686] Page fffffea0000e7c280 is in VMA 19 (8 currently)
[ 48.081958] Page fffffea0000e79000 is in VMA 19 (9 currently)
[ 48.082232] Page fffffea0000e7bf40 is in VMA 19 (10 currently)
[ 48.083760] Page fffffea0000f91200 is in VMA 20 (1 currently)
[ 48.084074] Page fffffea0000f91200 is in VMA 21 (1 currently)
[ 48.085511] Page fffffea0000e850c0 is in VMA 20 (2 currently)
[ 48.085810] Page fffffea0000e850c0 is in VMA 21 (2 currently)
[ 48.086087] Page fffffea0000e9f440 is in VMA 20 (3 currently)
[ 48.086373] Page fffffea0000e9f440 is in VMA 21 (3 currently)
[ 48.086718] Page fffffea0000e31240 is in VMA 20 (4 currently)
[ 48.087013] Page fffffea0000e31240 is in VMA 21 (4 currently)
[ 48.087289] Page fffffea0000e851c0 is in VMA 20 (5 currently)
[ 48.087591] Page fffffea0000e851c0 is in VMA 21 (5 currently)
```

## Immediate Goal: Spatial Locality

- The “Hybrid Swap” method was meant to use both spatial and temporal locality.
- Spatial locality deals with how close addresses from a sequence are.
- By iterating through processes from the current task, virtual memory areas (VMA's) were put in an array.
- As the function iterated through pages, each page was checked to all the VMA's, using *page\_address\_in\_vma()*.
- Each “match” had its information printed.
- These matches could be saved and recorded in another array for future use, such as analyzing the average distance between page addresses.



## Setup

- Started with Qemu program, booting from kernel source code inside.
- Tested with ImageMagick on a large image of a galaxy. This forced the computer to have page faults and use swap.
- Coded inside function *shrink\_page\_list()*.

## Long-Term:

## Conclusion

- First
- Second