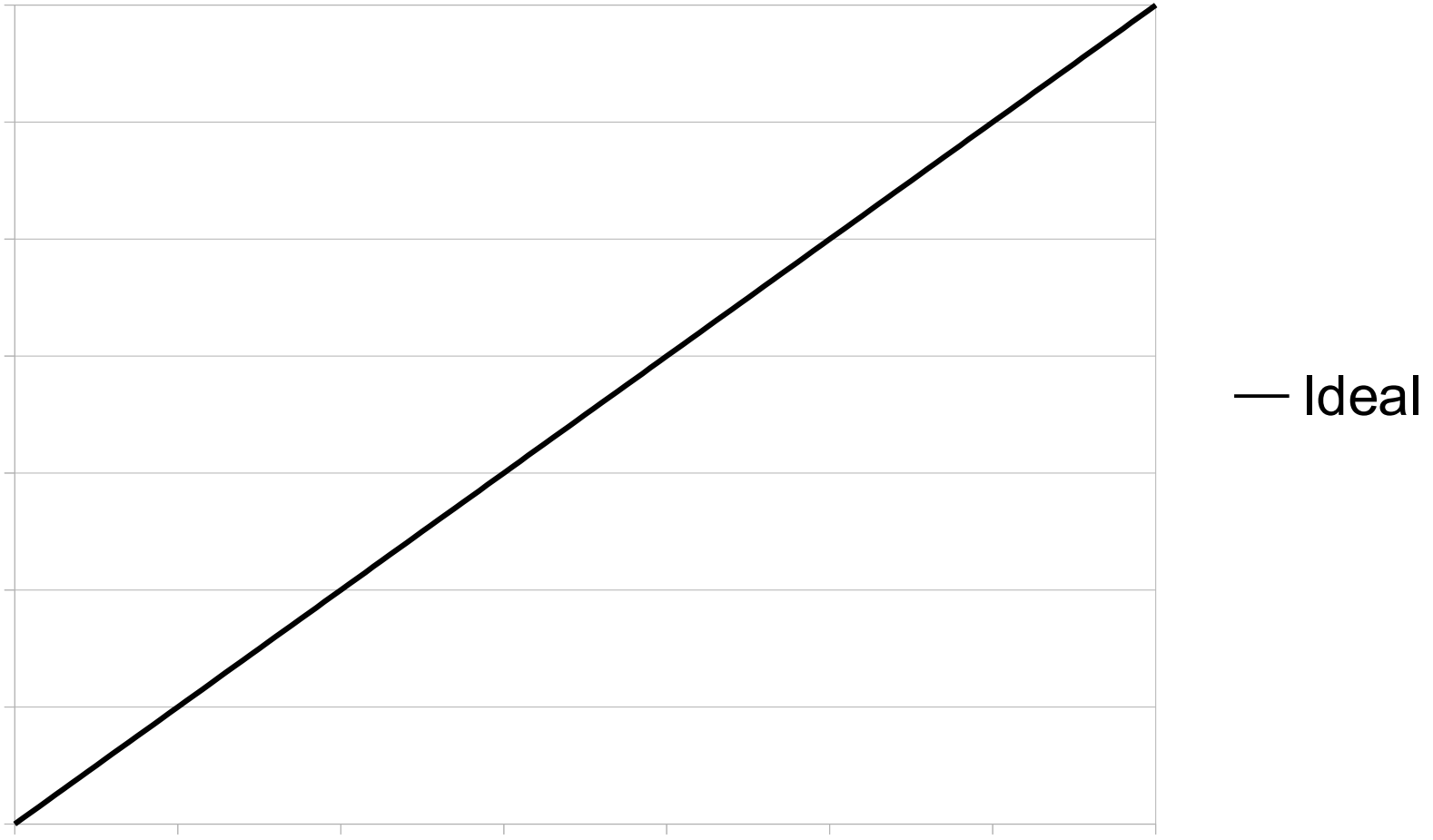
# How SMPng Works and Why It Doesn't Work The Way You Think

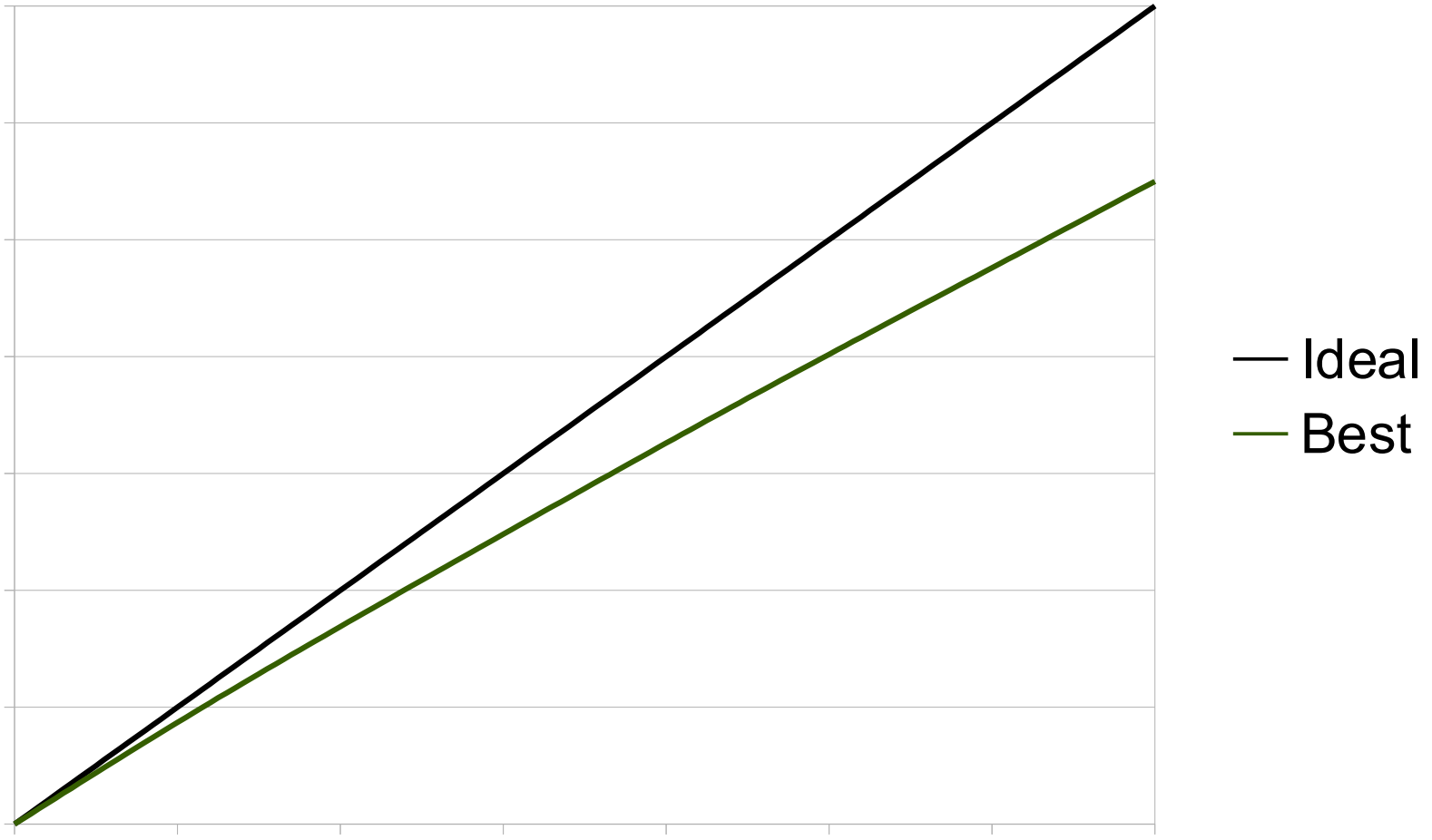
NYC*BUG
February 6, 2013

John Baldwin
jhb@FreeBSD.org

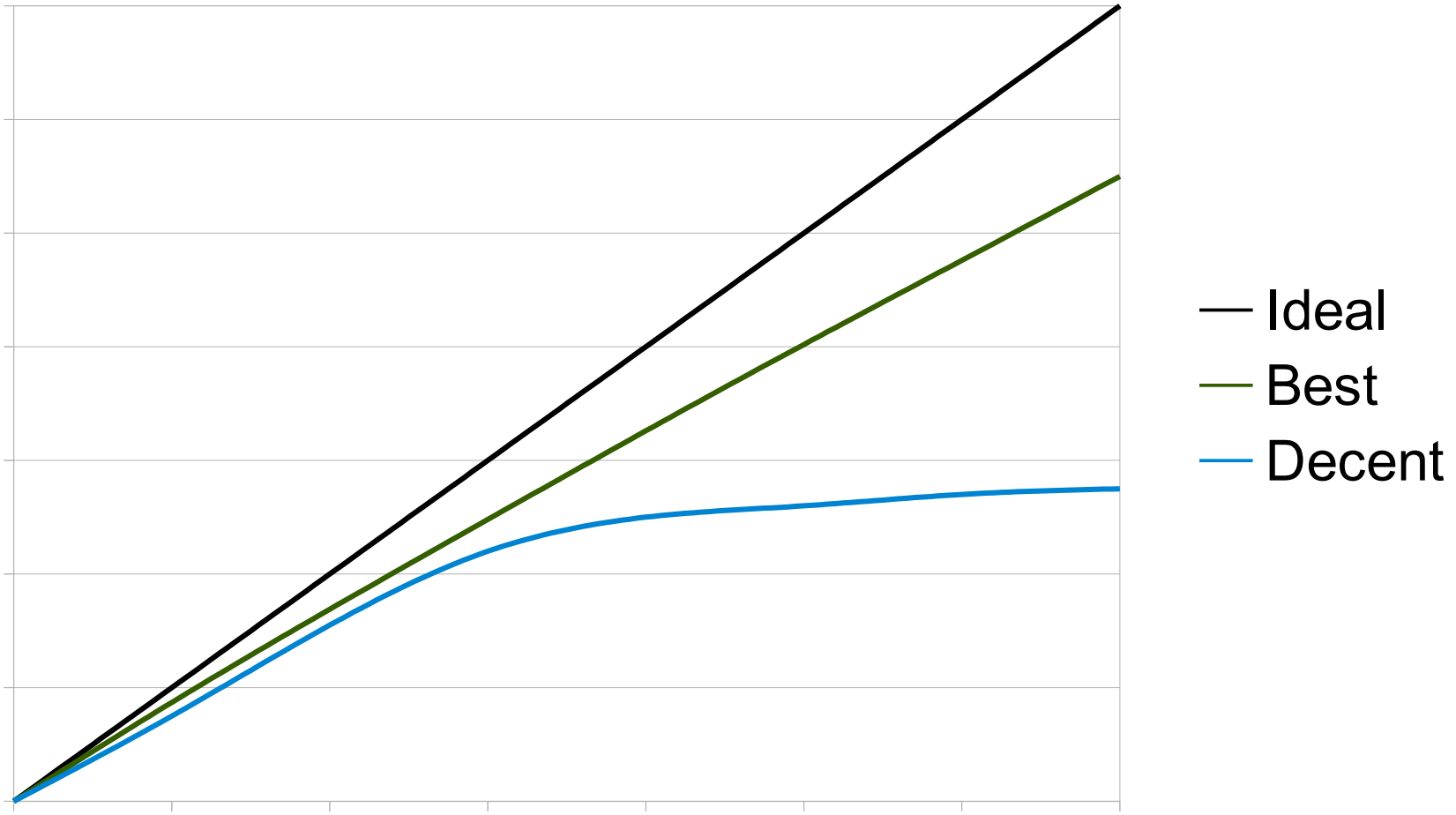
# Ideal SMP



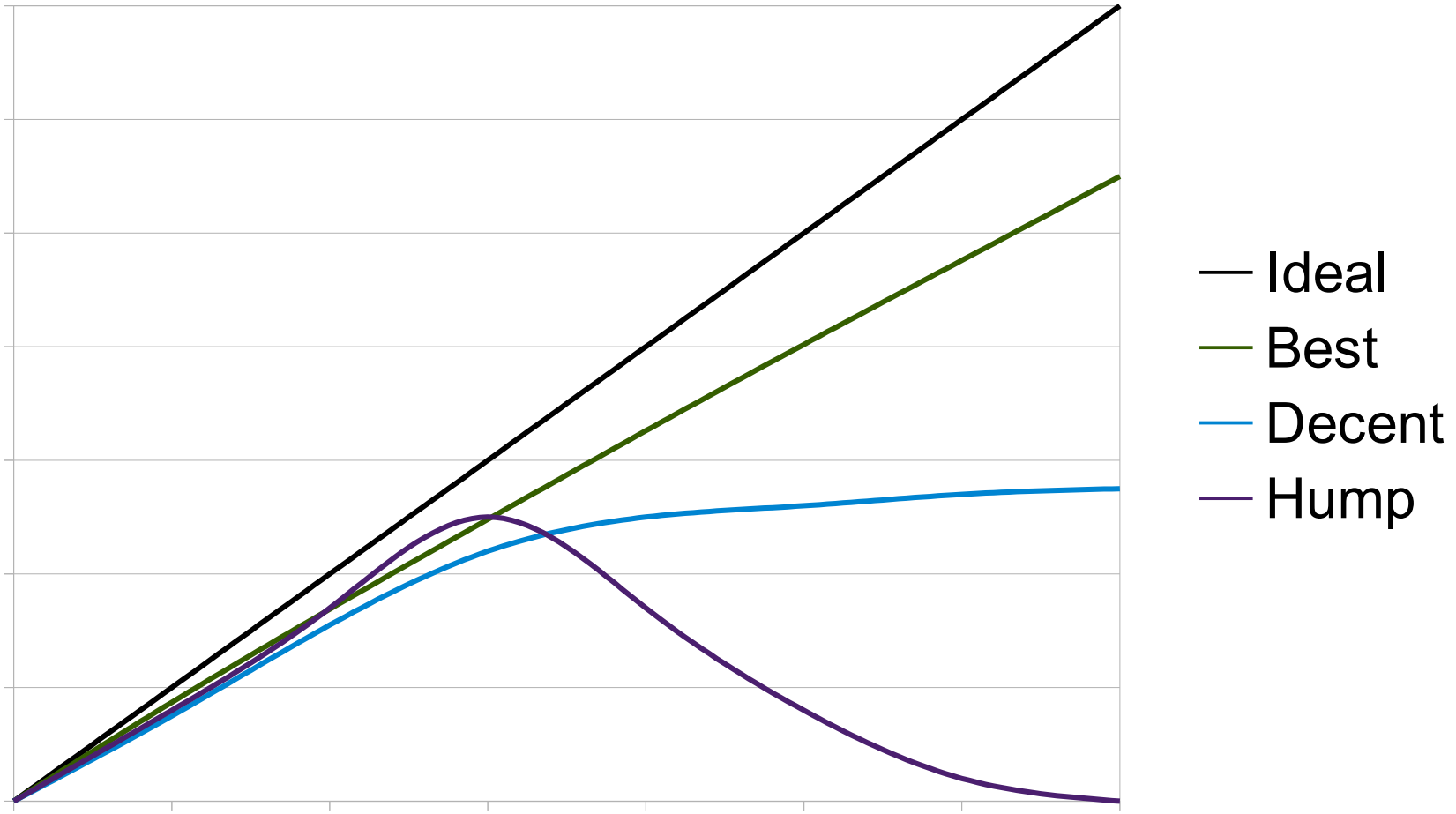

# Ideal SMP



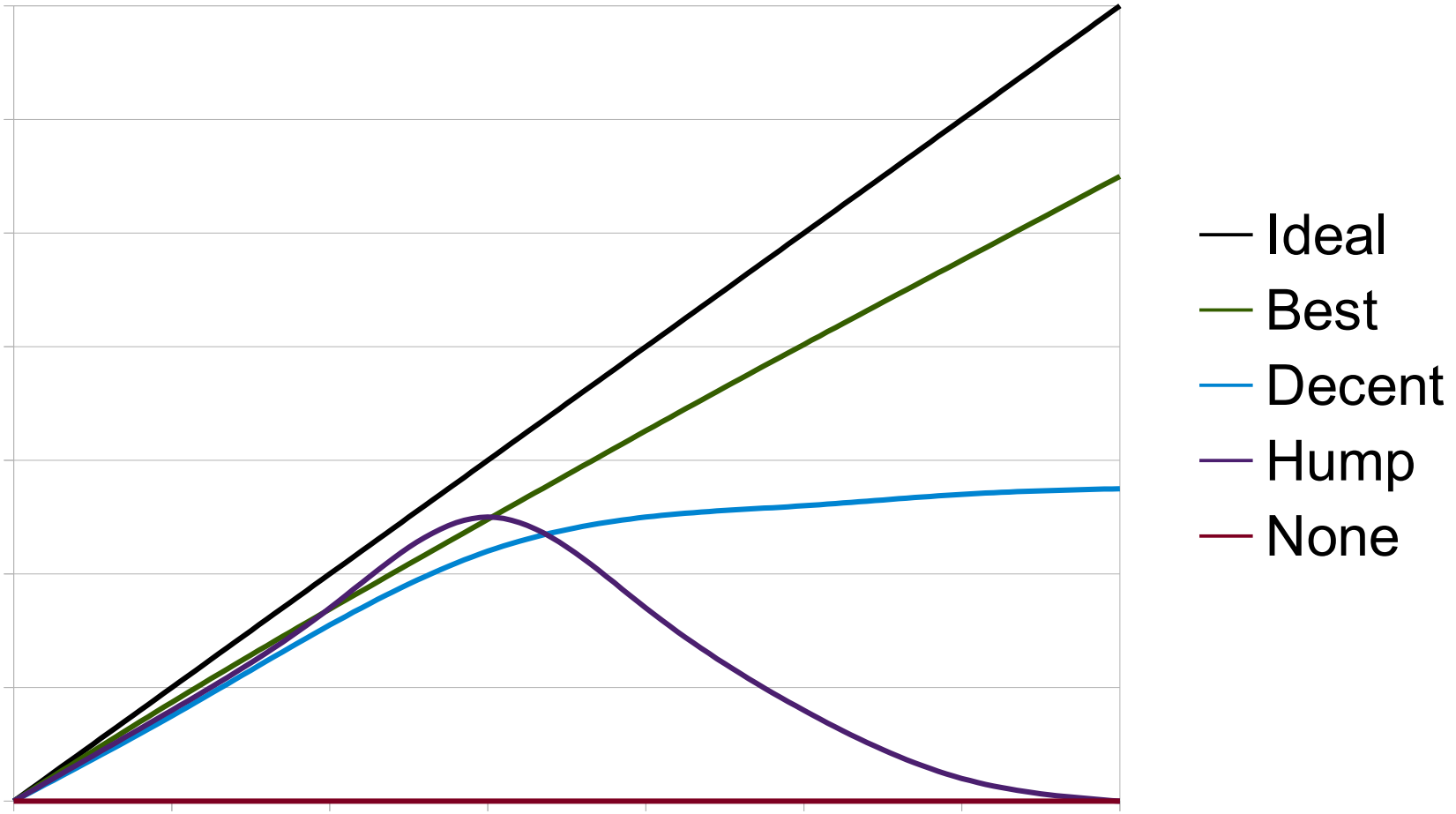

# Ideal SMP




# Ideal SMP




# Ideal SMP




# Why is Scaling Hard?

- N CPUs should do N amount of work
- Shared Resources
  - Can only be changed by one CPU at a time
  - Multiple CPUs have to take turns (“synchronize”)
- Examples
  - Devices
  - TCP Connections
  - TTYs

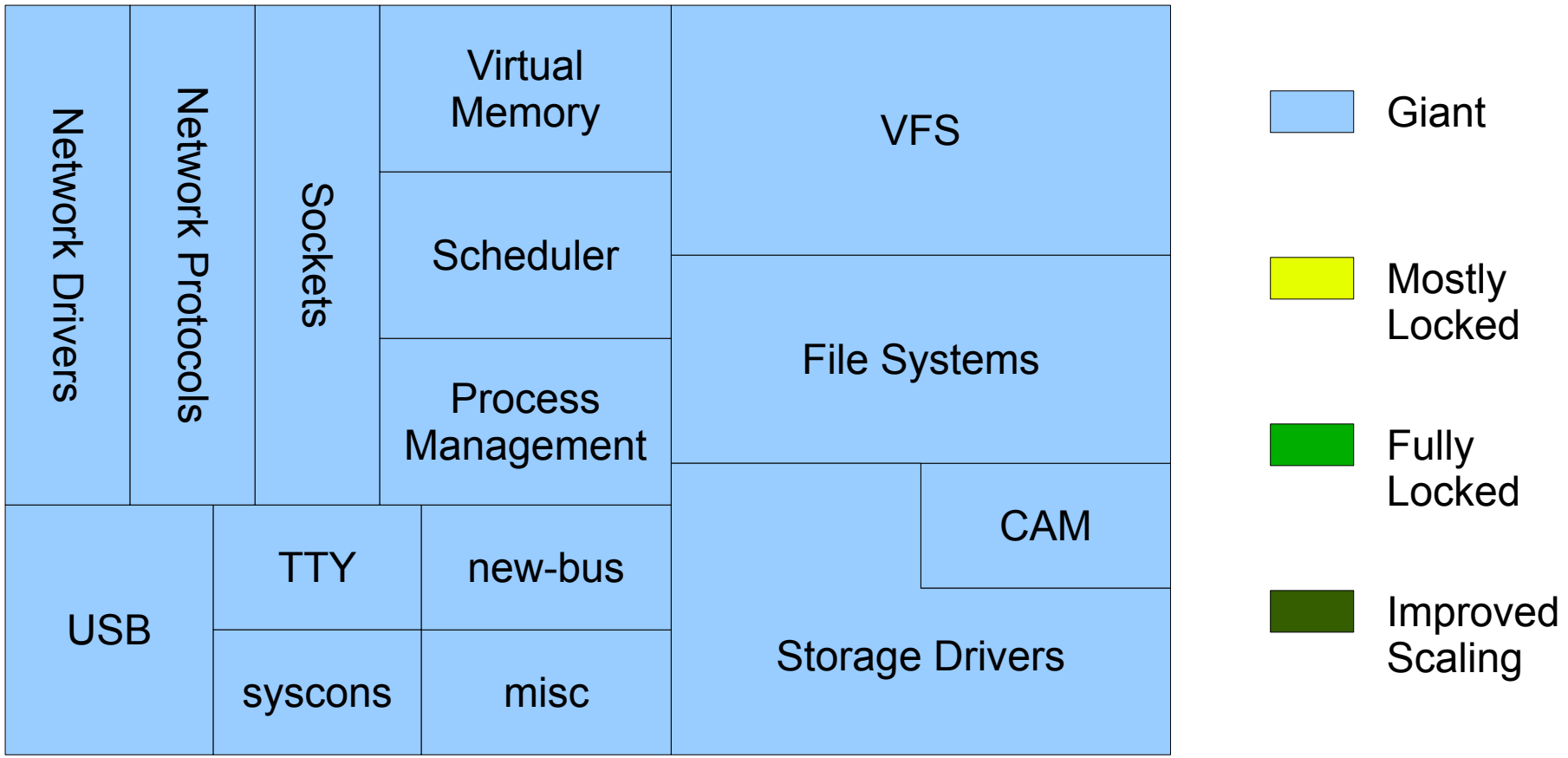
# Old Problem

- Signal handlers in userland
  - Limited ability (sigaction(2) list)
  - Threads can mask handlers
- Interrupts in traditional UP kernels
  - Limited ability (not able to block)
  - “Top-half” masked interrupts
  - spl(9) API in BSD and FreeBSD < 5

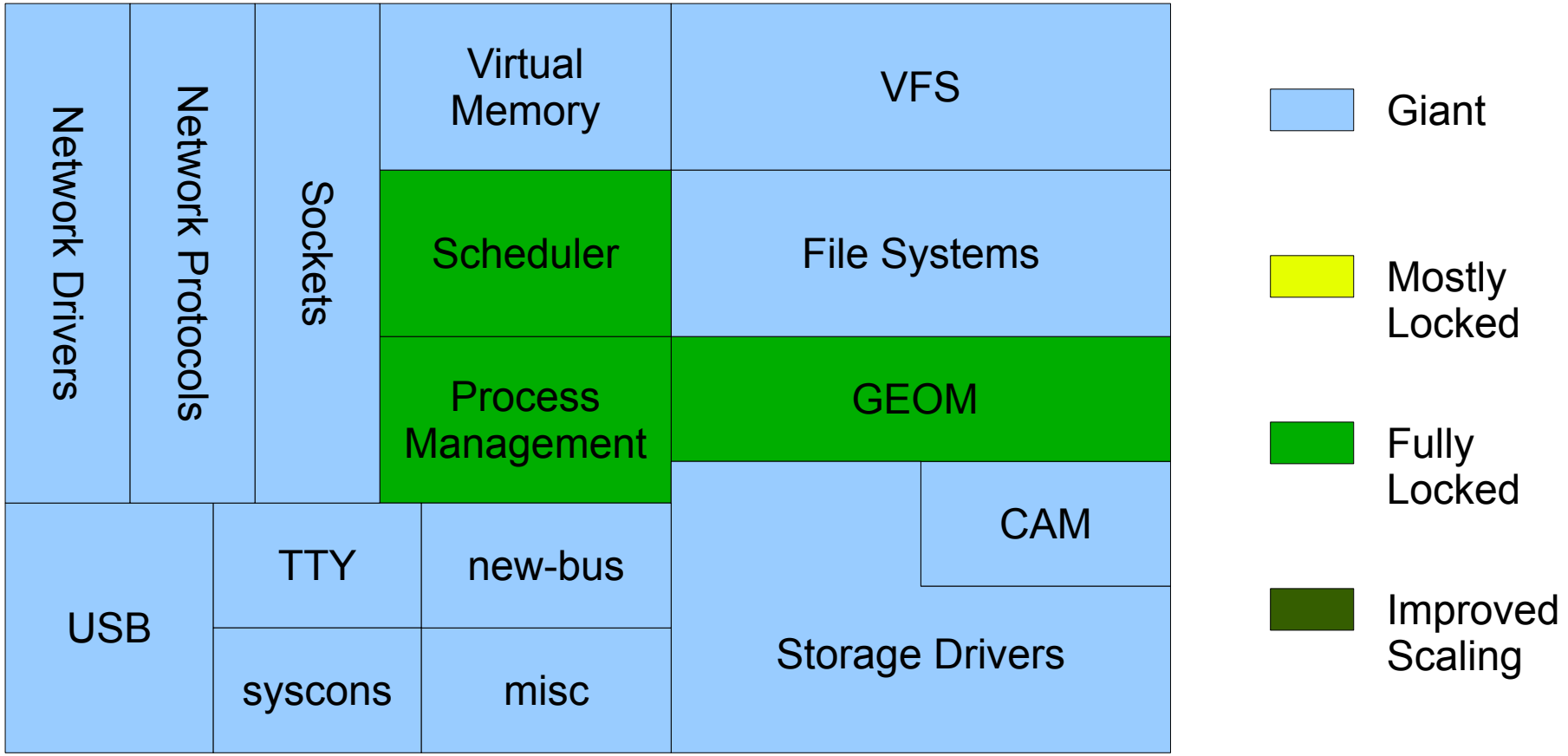
# SMP: New Wrinkle

- Adding threads to a userland application requires additional synchronization (locking)
- SMP kernels require locking as well
- Improving scalability is an iterative process of identifying bottlenecks and improving them
- SMPng is a project to replace a “Giant” lock around the entire kernel with “smaller” locks

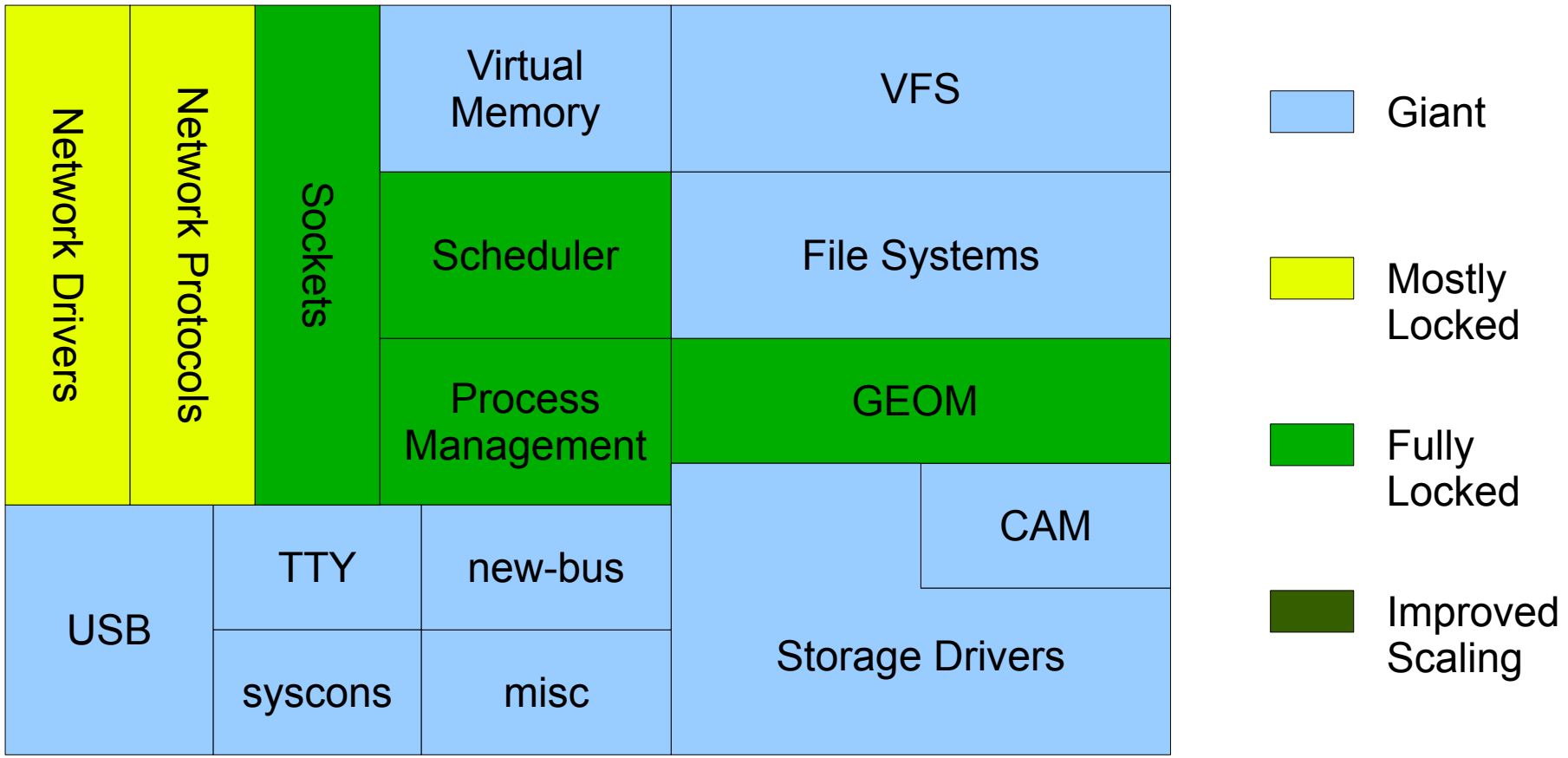
# FreeBSD 3.0


Network Drivers
Network Protocols
Sockets
Virtual Memory
Scheduler
Process Management
VFS
File Systems
CAM
Storage Drivers
USB
TTY
new-bus
syscons
misc
Giant
Mostly Locked
Fully Locked
Improved Scaling

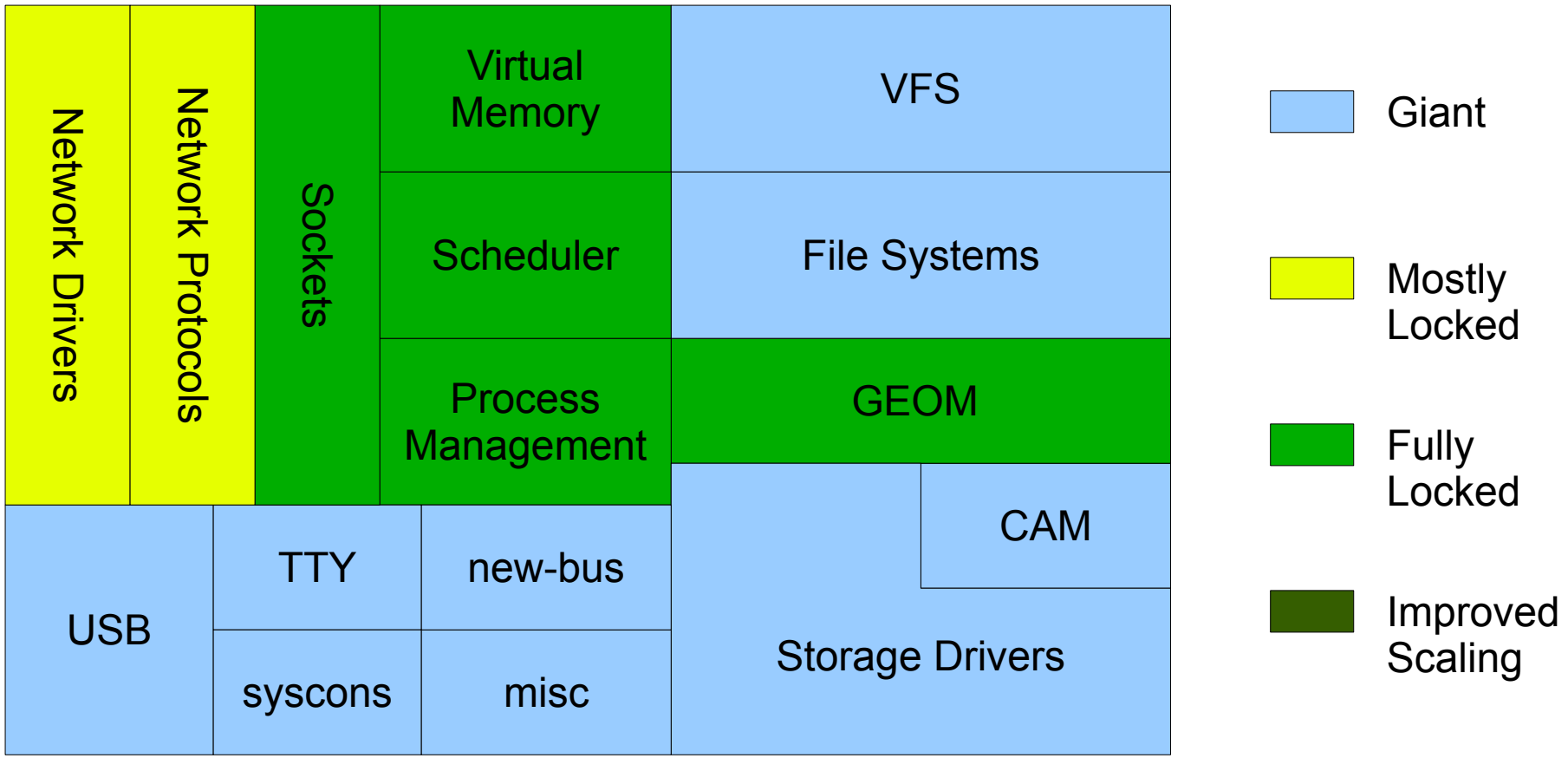
# FreeBSD 5.0


Network Drivers
Network Protocols
Sockets
Virtual Memory
VFS
Scheduler
File Systems
Process Management
GEOM
CAM
TTY
new-bus
USB
syscons
misc
Storage Drivers
Giant
Mostly Locked
Fully Locked
Improved Scaling

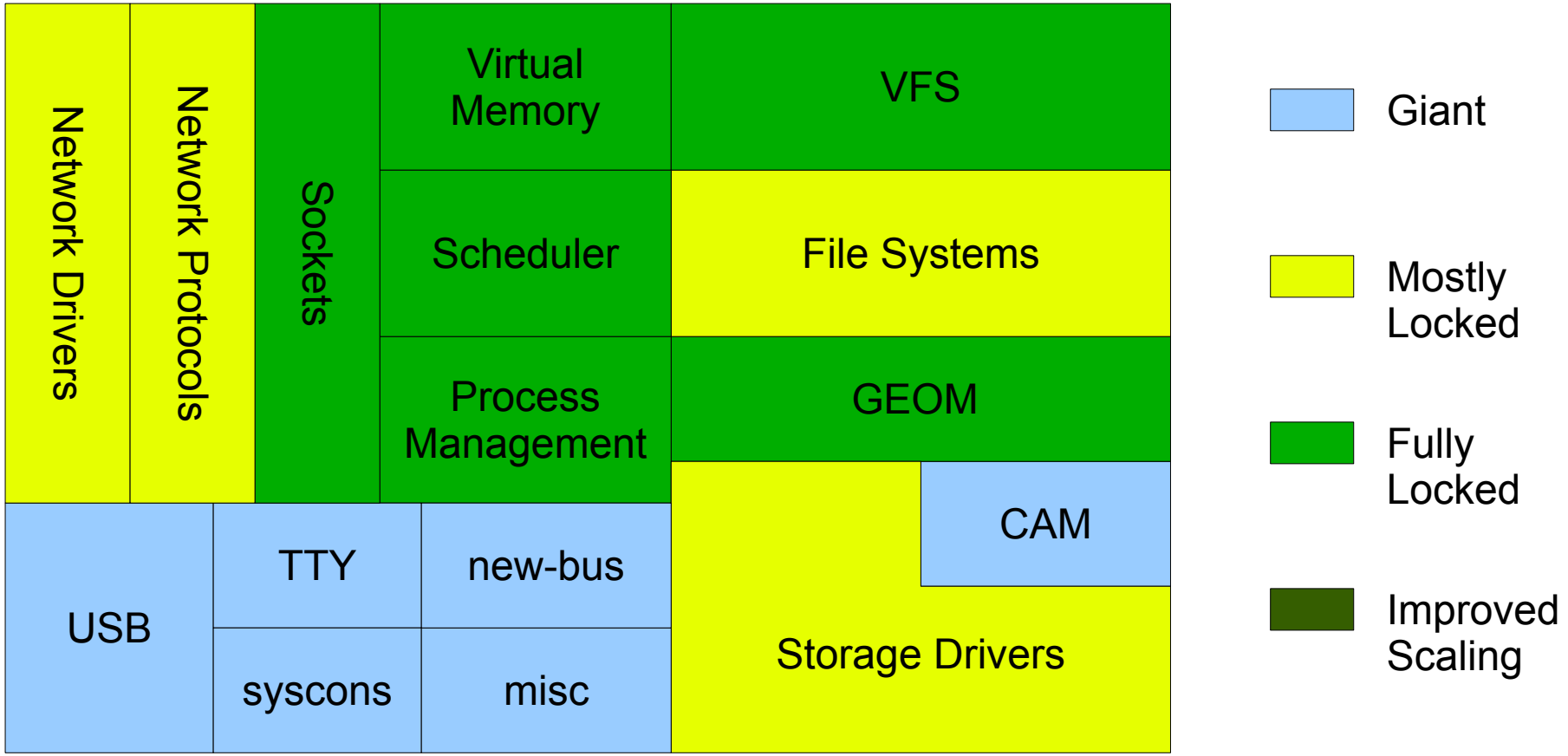
# FreeBSD 5.3


Network Drivers
Network Protocols
Sockets
Virtual Memory
VFS
Scheduler
File Systems
Process Management
GEOM
CAM
TTY
new-bus
USB
syscons
misc
Storage Drivers
Giant
Mostly Locked
Fully Locked
Improved Scaling

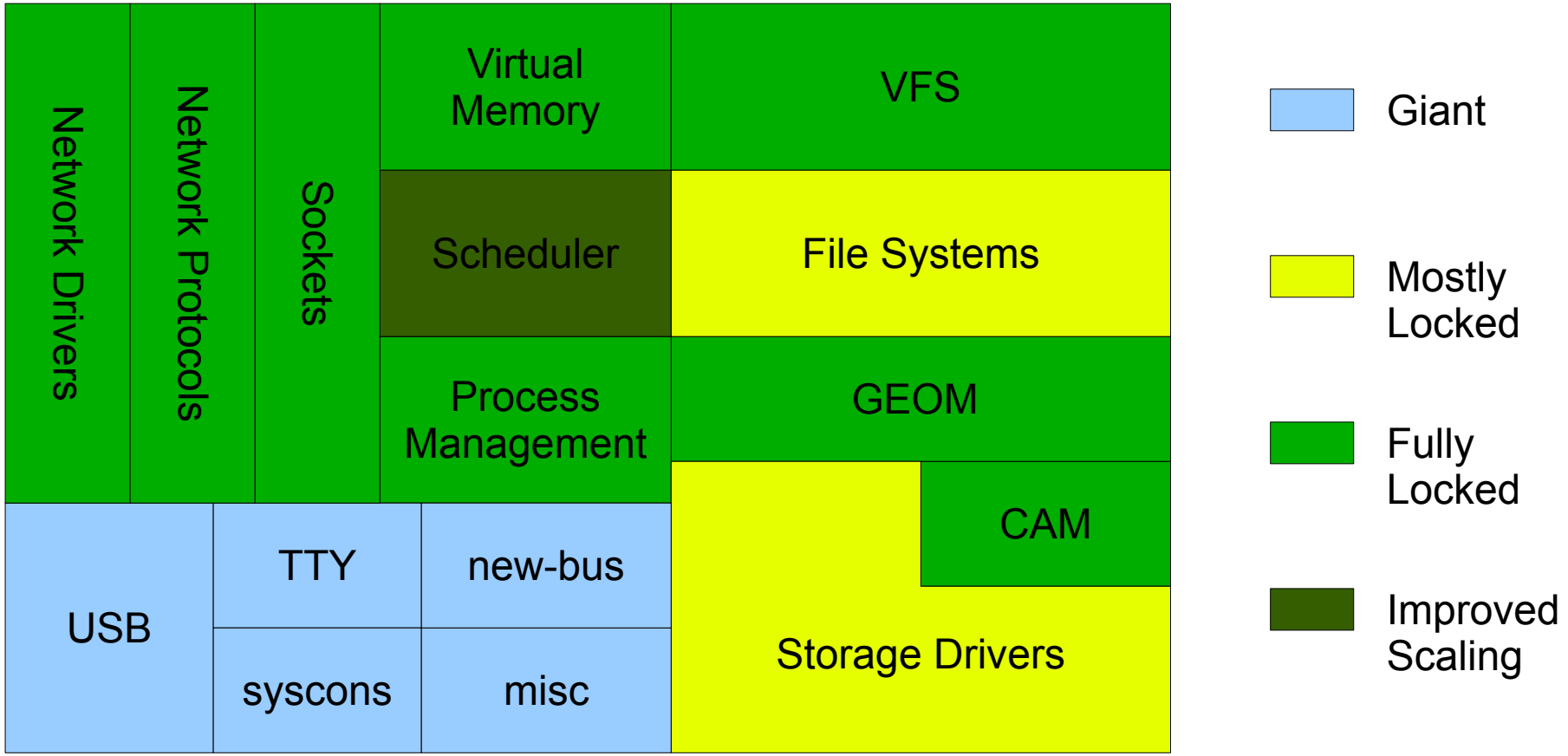
# FreeBSD 5.4


Network Drivers
Network Protocols
Sockets
Virtual Memory
VFS
Scheduler
File Systems
Process Management
GEOM
CAM
TTY
new-bus
USB
syscons
misc
Storage Drivers
Giant
Mostly Locked
Fully Locked
Improved Scaling

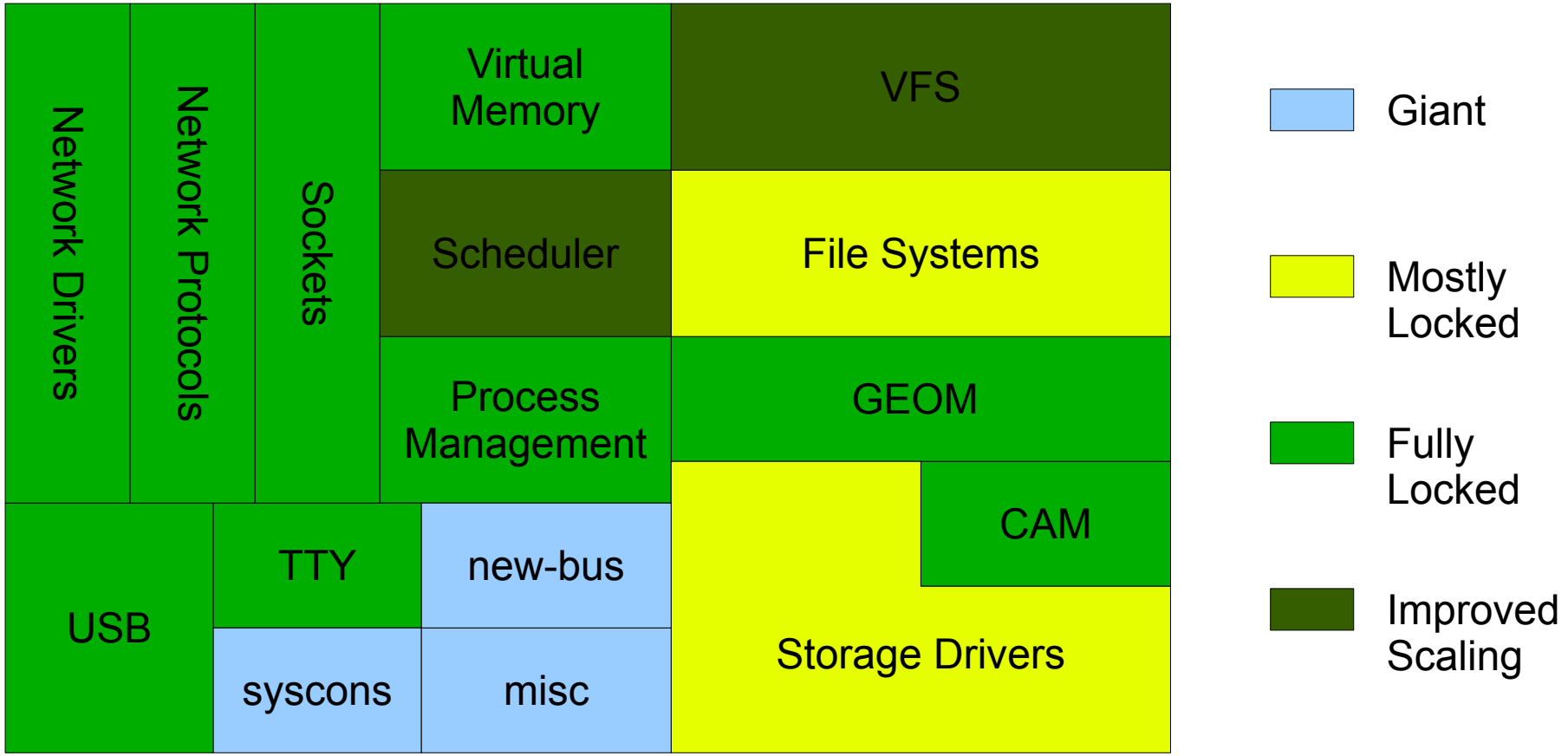
# FreeBSD 6.0


Network Drivers
Network Protocols
Sockets
Virtual Memory
VFS
Scheduler
File Systems
Process Management
GEOM
CAM
USB
TTY
new-bus
syscons
misc
Storage Drivers
Giant
Mostly Locked
Fully Locked
Improved Scaling

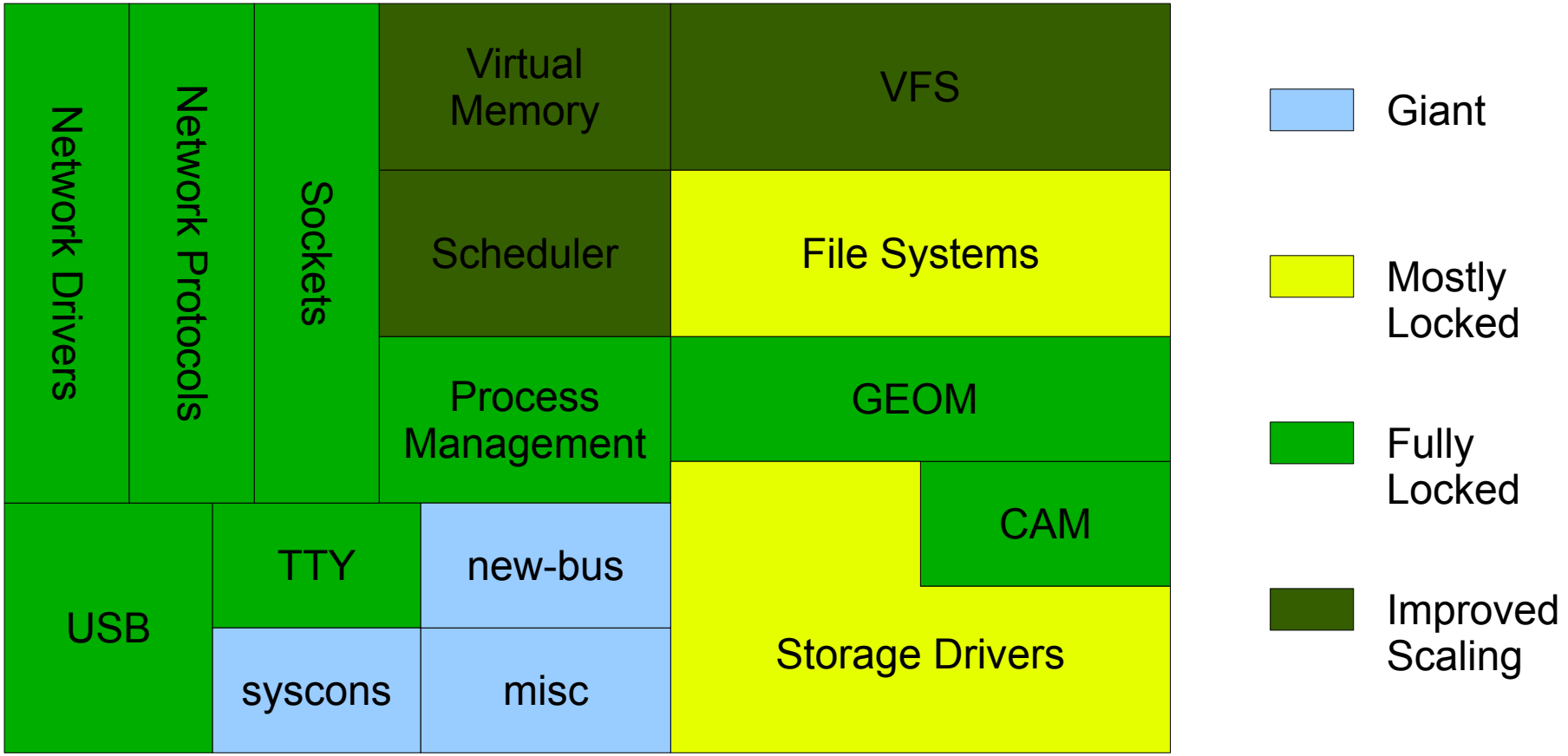
# FreeBSD 7.0


Network Drivers
Network Protocols
Sockets
Virtual Memory
VFS
Scheduler
File Systems
Process Management
GEOM
CAM
USB
TTY
new-bus
syscons
misc
Storage Drivers
Giant
Mostly Locked
Fully Locked
Improved Scaling

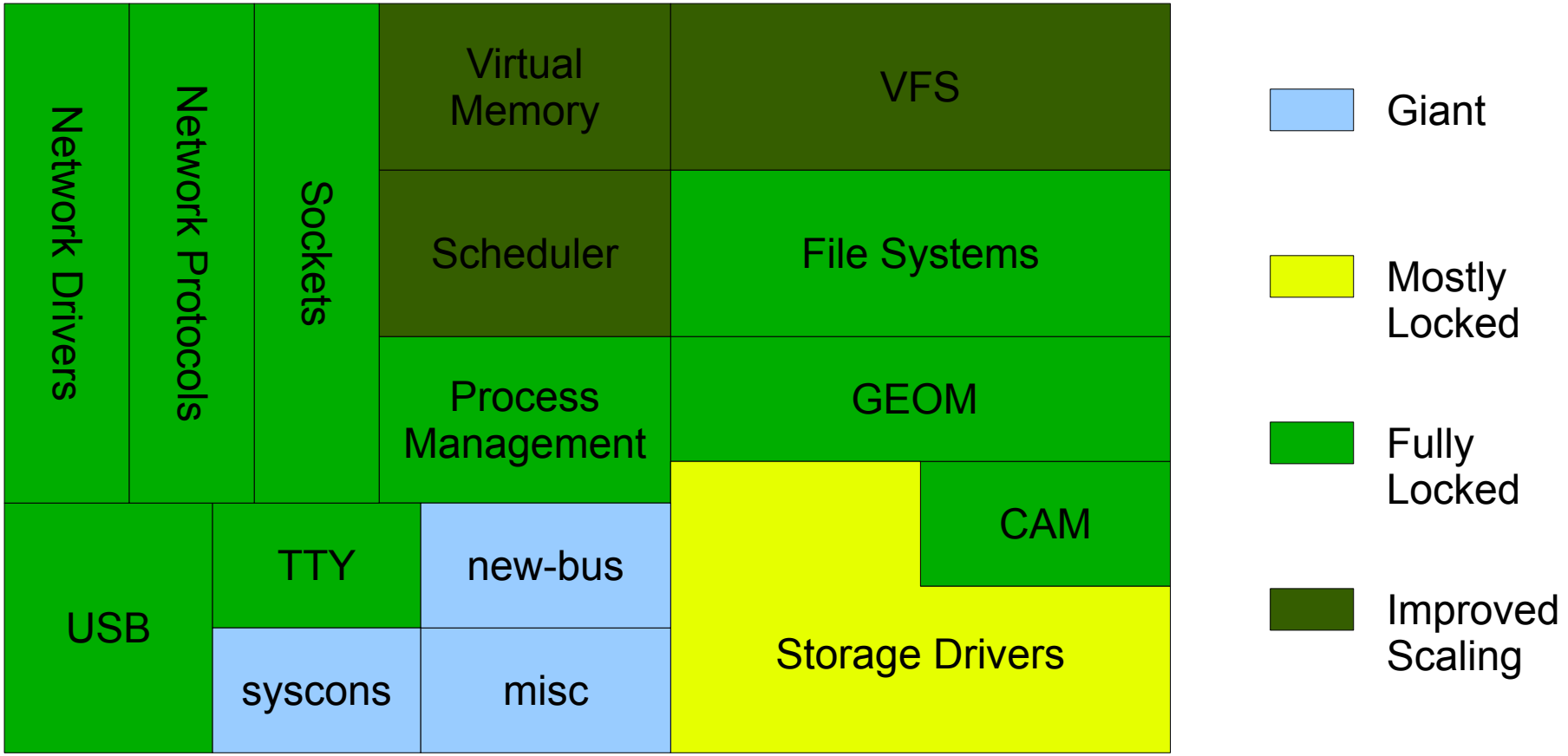
# FreeBSD 8.0


Network Drivers
Network Protocols
Sockets
Virtual Memory
VFS
Scheduler
File Systems
Process Management
GEOM
CAM
USB
TTY
new-bus
syscons
misc
Storage Drivers
Giant
Mostly Locked
Fully Locked
Improved Scaling

# FreeBSD 9.0


Network Drivers
Network Protocols
Sockets
Virtual Memory
VFS
Scheduler
File Systems
Process Management
GEOM
CAM
USB
TTY
new-bus
syscons
misc
Storage Drivers
Giant
Mostly Locked
Fully Locked
Improved Scaling

# FreeBSD 10.0


Network Drivers
Network Protocols
Sockets
Virtual Memory
VFS
Scheduler
File Systems
Process Management
GEOM
CAM
USB
TTY
new-bus
syscons
misc
Storage Drivers
Giant
Mostly Locked
Fully Locked
Improved Scaling

# Remaining Giant Uses

- Boot Time Initialization
- Suspend/Resume
- Module Event Handlers
- new-bus
- Some Storage Drivers (aha(4), dpt(4))
- syscons
- Miscellaneous (e.g., certain non-trivial sysctls)

# Hardware Tangent

- SATA NCQ (previously only for SCSI)
- Multiple RX/TX Queues on NICs
  - Nearly mandatory for 10G
- X86 Memory Controllers Moved on Die

# Finally

- http://www.freebsd.org/~jhb/papers/smp/
- Questions?