深入单机性能极限

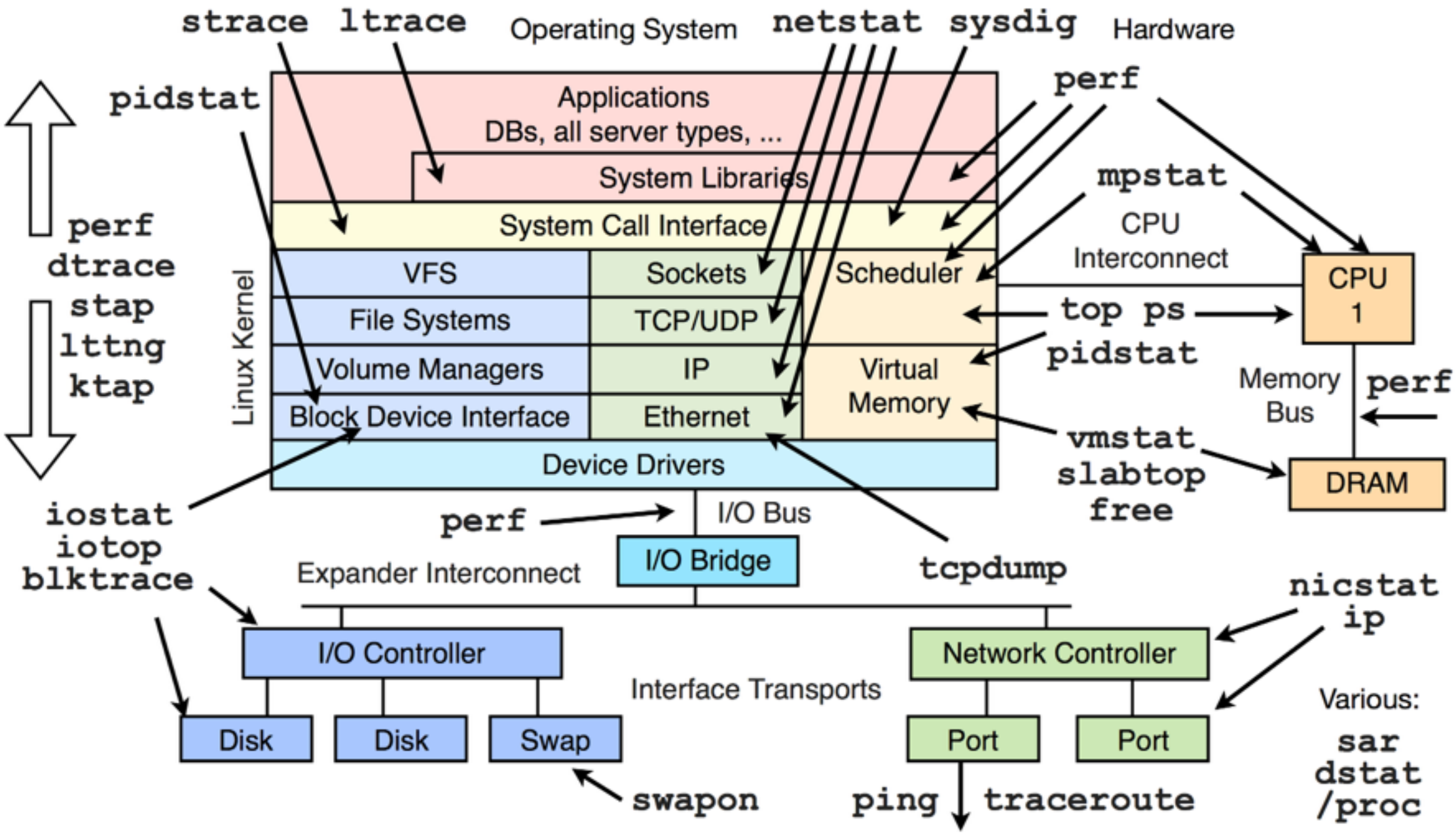
张景埕 <diogin@gmail.com> @diogin

PHPCon2014

/about

- 张景埕，@diogin
- 原360工程师，服务端技术委员会委员
- 现百车宝技术中心总经理
- 杂食程序猿 && 业余性能爱好者

Linux性能.....


strace
ltrace
Operating System
netstat
sysdig
Hardware
perf
pidstat
Applications
DBs, all server types, ...
System Libraries
System Call Interface
VFS
Sockets
Scheduler
File Systems
TCP/UDP
Volume Managers
IP
Virtual Memory
Block Device Interface
Ethernet
Device Drivers
Linux Kernel
perf
dtrace
stap
lttng
ktap
mpstat
CPU Interconnect
CPU 1
top ps
pidstat
Memory Bus
perf
vmstat
slabtop
free
DRAM
iostat
iotop
blktrace
perf
I/O Bus
I/O Bridge
Expander Interconnect
tcpdump
nicstat
ip
I/O Controller
Network Controller
Interface Transports
Disk
Disk
Swap
Port
Port
Various:
sar
dstat
/proc
swapon
ping
traceroute

一、硬件和软件特征

服务器硬件特征

1. 所有组件都越来越牛X，组件的个数越来越多(Scalability)
2. CPU：QPI, NUMA, 超线程, Turbo Boost, 深度流水线, 多级缓存(数据,指令,TLB), 高级指令
3. 内存：DDR4, ECC
4. 存储：PCI-E SSD(高效主控, MLC/SLC), SAS HDD
5. 网络：PCI-E, 10Gbps
6. 各级 Cache 和 Buffer

潜在的硬件瓶颈

- CPU使用率(指令, I/O等待, 软硬中断, 上下文切换)
- 内存容量及频率
- 存储总线带宽和频率(SATA, SAS, PCI-E)、设备的IOPS(随机I/O、顺序I/O、平均I/O，HDD，SSD)
- 网卡中断不均衡、吞吐量、总线带宽、网络带宽

服务器硬件极限

- CPU所有核心都100%忙于处理指令（用户态或核心态）、中断（软中断和硬中断）、等待I/O或进行上下文切换
- 全部物理内存帧均已保留或映射并标为cache，swap到存储的部分也全满
- 存储总线带宽吃满或IOPS达到上限值
- 网卡中断吃满CPU或开始丢包或带宽吃满

服务器软件特征

- 内核优化力度大，同时可调性非常高
- CPU密集、I/O密集，用户CPU/内核CPU
- 高并发 + 并行、线程CPU绑定、高级指令的使用
- 非阻塞I/O、异步I/O、事件通知
- 贯穿各级的 Cache 和 Buffer
- 内存拷贝最少化、指令数最少化
- 对网络栈特性的深入利用（慢启动、no_delay等）
- JIT技术可以得到比静态代码更高效的性能

潜在的软件瓶颈

- 驱动程序性能不够高，不支持高级特性
- 内核版本低，缺乏高级特性
- 文件系统在部分特定用法下性能很差
- 使用的语言环境运行效率不高
- 服务器模型扩展性差，浪费现象严重
- 网络协议栈调整得不好，导致网络并发上不去

服务器软件极限

- 内核施加各种限制
- 进程占用的虚存大小受限制
- 进程驻留内存的页受限制、栈帧大小受限制
- 进程可使用的CPU时间受限制
- 进程可打开的文件描述符受限制
- 进程可用的管道、队列、文件锁受限制

二、基本调优工具

一

- mpstat - 查看CPU使用率信息
- vmstat - 报告虚存统计数据
- iostat - 报告CPU状态和存储I/O统计数据
- netstat - 报告网络栈各项参数
- ps - 系统上运行的进程及其状态
- top - 报告占用CPU和内存最大的进程
- sar - 报告系统活动状态
- pidstat - 检查具体某个进程的CPU和内存情况
- perf - 综合性能检查工具

二

- sysctl - 调整内核参数
- strace - 跟踪进程的执行
- dtrace - 综合跟踪工具
- stap - 综合跟踪工具
- taskset - 绑定执行CPU
- pmap - 报告进程内存映射信息
- free - 报告物理帧和虚拟帧信息
- dmesg - 系统启动信息，包括对各设备的监测
- /proc - 各项系统参数读写接口

三

- /sys - 额外系统参数读写接口
- iotop - 类似top，但用于检测I/O
- ionice - 类似nice，但用于设置进程的I/O优先级
- ifconfig - 监测与配置网络接口
- tcpdump - 捕获某个网络接口上的流量
- nicstat - 监测网卡状态

三、通用性能调优

基本调优策略

- 使用高效的算法和操作，避免不必要的开销
- 缓存开销大的操作，缓冲开销大的写操作
- 利用(处理和存储)局部性、存储器金字塔
- 保证全部进程的全部内存页都在物理内存，很少换页
- 对一些可预测到的数据，可以进行预取、预暖

性能调优误区

- CPU使用率越高系统越慢(空闲的时候CPU就睡觉)
- 剩余内存越少系统越慢(可能用在了文件系统缓存上)
- 多个线程一起accept tcp连接会更快
- 凭直觉

硬件调优

- CPU启用 Turbo Boost
- 编译适合CPU的程序
- 将线程/进程绑定到CPU核
- 让内存工作在最高频率
- 多设备并行I/O
- 组件连接到对应速率最高的接口

Linux调优：进程

- kernel.pid-max
- kernel.threads-max

Linux调优：虚存

- vm.swappiness=0，保证有物理内存时不swap
- huge page?

Linux调优：存储

- ext4/xfs/reiserfs/brtfs/zfs …
- fs.file-max
- fs.nr_open

Linux调优：网络

- 本地端口数不够用
- time wait 桶、时间缩小、回收与复用
- 自适应 read/write buffer
- syn_backlog, somaxconn, net_dev_backlog

Linux应用调优

- /etc/security/limits.conf，控制各种应用级上限
- nice
- 使用高效的内存分配库
- tcp buffer, listen backlog

四、特定系统调优

nginx

- 使用适合CPU核数的worker进程数
- 设置进程的CPU亲缘性
- 设置worker的优先级
- aio
- pcre-jit
- open_file_cache
- tcp_nopush
- 启用各种cache(如fastcgi cache, …)
- 各种 timeout, buffer

fpm

- process.max
- process.priority
- rlimit_files
- listen.backlog
- pm, pm.*

mysql

- 太多了……
- <<High Performance MySQL>>

redis

- O(1) > O(logN) > O(N) > O(N^2) > O(2^N)
- max clients
- max memory
- timeout, buffer…
- tcp no delay
- rdb/aof
- 多进程

php代码

- opcache
- 明确路径的 require/include
- 一次运行需加载的文件数尽量少
- cache开销比较大的操作
- 减少远程调用数
- 减少浪费
- ……还有哪些?

这些只是开始...

参考资料

- Systems Performance: Enterprise and the Cloud
- Wikipedia
- Google

Q & A

张景埕 <diogin@gmail.com> @diogin

PHPCon2014