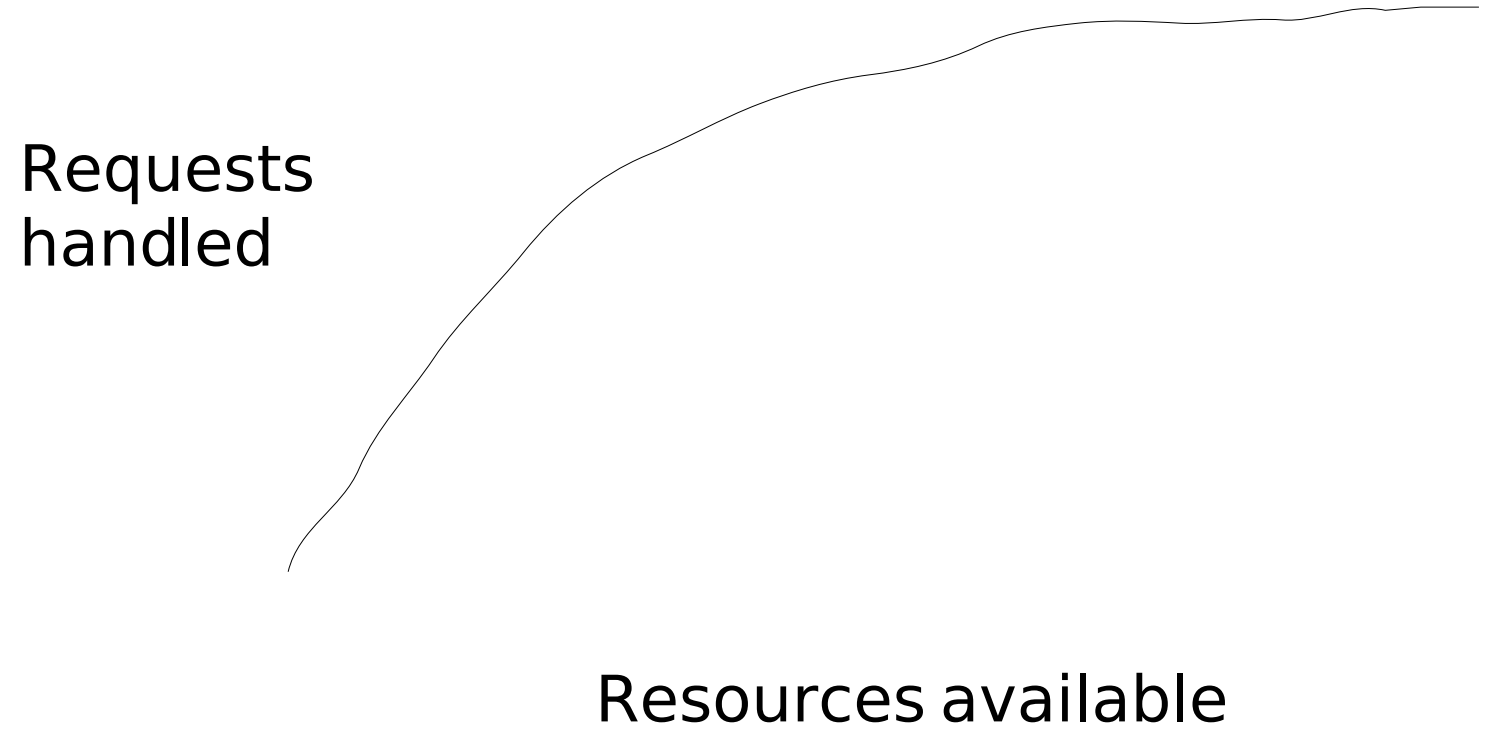
Advanced Unix System Administration

Lecture 10
March 17, 2008

Steven Luo
<sluo+decal@OCF.Berkeley.EDU>

Remarks on Performance

- The generic performance curve:



Remarks on Performance

- It's not worth optimizing one particular thing past the point when it's not the bottleneck anymore
- Hence your job is usually to identify the bottleneck, widen it, and repeat until satisfied with performance
- Performance tuning requires instrumentation, understanding, a bit of thought, and patience

Remarks on Performance

- Hardware effects
  - Choosing the right hardware can have a profound impact on performance
  - Components have different effects on performance
- More/faster CPUs
  - Linear scaling up to the concurrency limit/CPU processing limit of the application
  - Helps the tail in the limit of very high load

Remarks on Performance

- More memory
  - Can improve concurrency, extends the peak performance of the system by allowing more cache
  - Helps the tail in the limit of very high load
- Faster disks
  - Increases initial performance, overload performance by speeding up cache misses
- Faster network
  - Allows more connections, more data pushed

Remarks on Performance

- Performance tuning is always ultimately a tradeoff
  - Increasing one setting (i.e. concurrent processes) may come at the detriment of other performance attributes
  - What's best depends heavily on the workload, the resources of your system, and the bottlenecks that the system is encountering

Remarks on Performance

- Identifying performance bottlenecks
  - Basic instrumentation tools
    - System instrumentation (top, vmstat, iostat, etc.) can show you what subsystems and processes are most busy
    - Tracing can uncover high numbers of calls to one syscall/library function
    - strace and truss support displaying time spent information

Remarks on Performance

- Identifying performance bottlenecks
  - Profiling
    - gprof – allows you to show which functions a program is spending most of its time in
      - requires recompile and code knowledge
    - OProfile (Linux) – uses performance counting hardware to show which functions and syscalls take most time
    - These techniques may require a bit of digging into the code you're profiling – try simpler ideas first!

Networking Intro

- The OSI model
  - Seven layers that conceptually separate the different functions of a network stack very cleanly
  - No practical modern network stacks actually implement the full separation model
- Physical layer (layer 1)
  - Specifies the actual communications hardware
  - Fiber, copper twisted pairs, wireless, SCSI . . .

Networking Intro

- Data link layer (layer 2)
  - Specifies details of over-the-wire communication and error correction between physical hosts
  - We'll focus on Ethernet in this class
- Network layer (layer 3)
  - Routes traffic from Point A to Point B, possibly with QoS considerations
  - IP is the only important example nowadays

Networking Intro

- Transport layer (layer 4)
  - Provides facilities for a link between hosts, such as flow control, error correction
  - TCP and UDP are the most important
- Session layer (layer 5)
  - Provides the link between hosts (connections, ports, state)
  - TCP performs the functions of the session layer; applications running on UDP must provide their own session facilities

Networking Intro

- Presentation layer (layer 6)
  - Data representation for the application
  - Almost always performed by the application nowadays; protocols are usually considered part of layer 6
  - We'll cover some important examples
- Application layer (layer 7)
  - Provides a useful service to users
  - The application implementing a protocol is in layer 7