# Performance Tuning the Linux Kernel

Christoph Lameter, Ph.D.
cl@linux-foundation.org

# Overview

- Performance tuning goals
- Network performance
- VM performance
- Multi threading the kernel: Lock contention
- CPU resources: cpu caches etc.
- Challenges due to increasing kernel complexity
- Future trends

# Tuning goals

- Throughput improvements
  - TCP throughput in mb/secs? (netperf)
  - Number of system ops per sec? (AIM9)
  - Number of tasks per sec? (AIM7)
- Latency improvements
  - Responsiveness
  - OS interrupt hold off / scheduling holdoff
- Variability improvement
  - Replicate runs, reduce outliers
- Memory footprint
  - Embedded solutions

# Throughput or Latency

- Throughput improvement requires batching
- Tradeoff between throughput and latency
- Performance and memory
- Performance and hardware resources
- Parallelism / Multicore issues
- Code efficiency

# Network performance

- Throughput per second
- Packets per second
- Latency of packets
- Special hardware features
  - GSO (Generic Segmentation Offload)
  - TSO (TCP Segmentation offload)
  - LRO (Large Receive Offload)
  - TOE (TCP Offload Engine)
  - VNIC (Virtualized NICs)

# Optimizing per cpu resources

- Per cpu cache hierachy
- Siblings share lower cpu caches
- MESI type cache coherency protocols
- Attempt to optimize cache footprints
- Per cpu areas
- Future per cpu allocator / per cpu operations
- cpu_alloc / cpu_ops

# Kernel Complexity

- Performance is reducing over time
- Hardware compensates
- Large cache footprint (both data and instruction)
- Complex percpu cache management
- Larger word size (32bit->64bit transition)
- Challenge to keep performance up to par

# Future Trends

- Increasing cores (6 cores per socket now, 8 cores per socket soon)
- Increasing cache sizes per core (10-50 MB!)
- Memory increases (fractions of a Terabyte)
- Multiple concurrent memory paths (->NUMA effects)
- Multiple concurrent I/O paths
- 2015: 128 cores, ~1TB Ram, 100-256 MB per cpu cache

# Performance Regression

- Regular performance degradation on release.
- 7 patches so far (not merged yet)
- May be addressed in 2.6.28 development cycle.
- Earlier kernels = faster