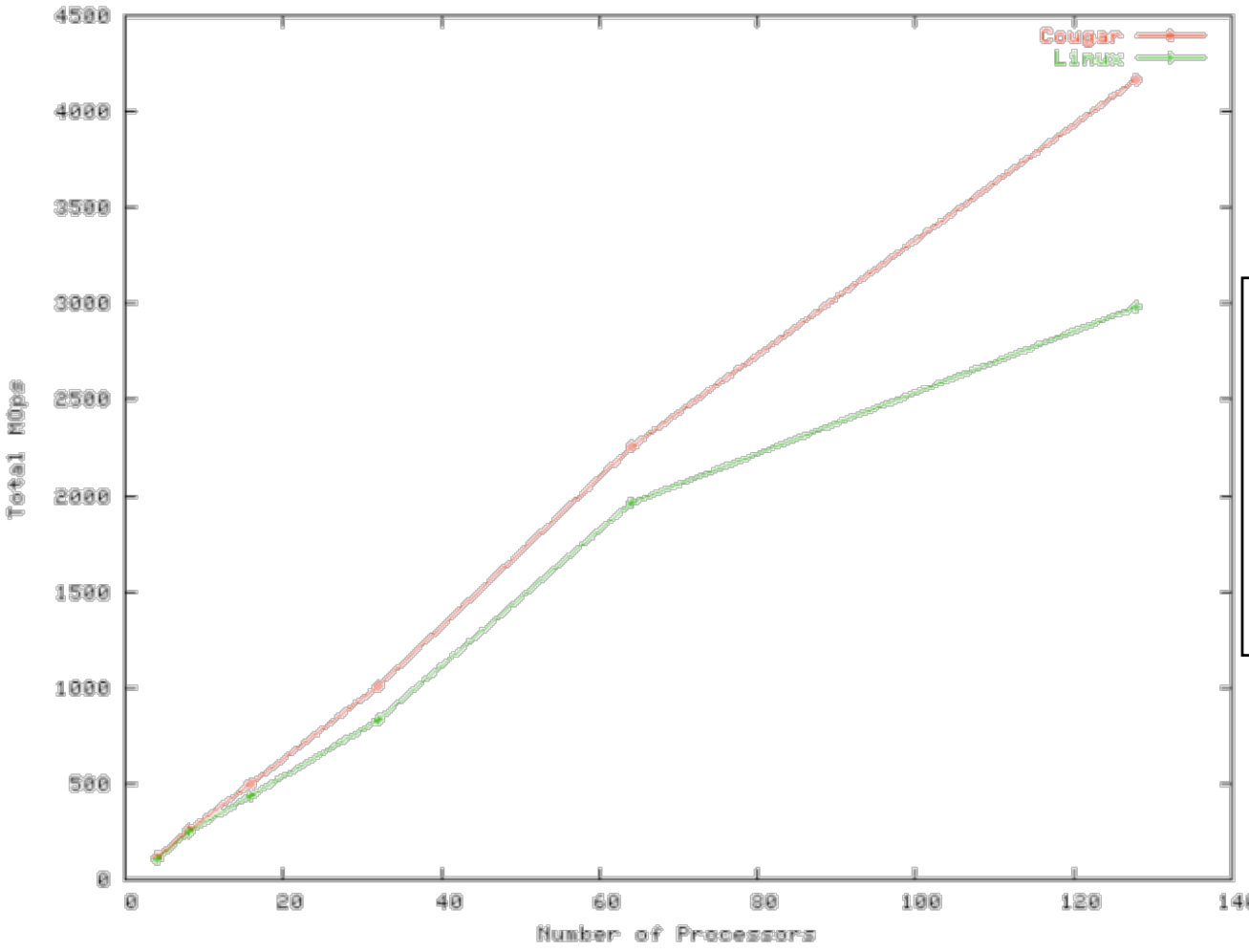
System Software

Recipe for a good MPP

1. Select Best Microprocessor
2. Surround it with a balanced or “bandwidth rich” environment
3. “Scale” the System
   - Eliminate Operating System Interference (OS Jitter)
   - Design in Reliability and Resiliency
   - Provide Scaleable System Management
   - Provide Scaleable I/O
   - Provide Scaleable Programming and Performance Tools
   - System Service Life (provide an upgrade path)

Scalable Software Architecture:
Why it matters for Capability Computing

NPB Result: MG
Standard Linux vs. Microkernel

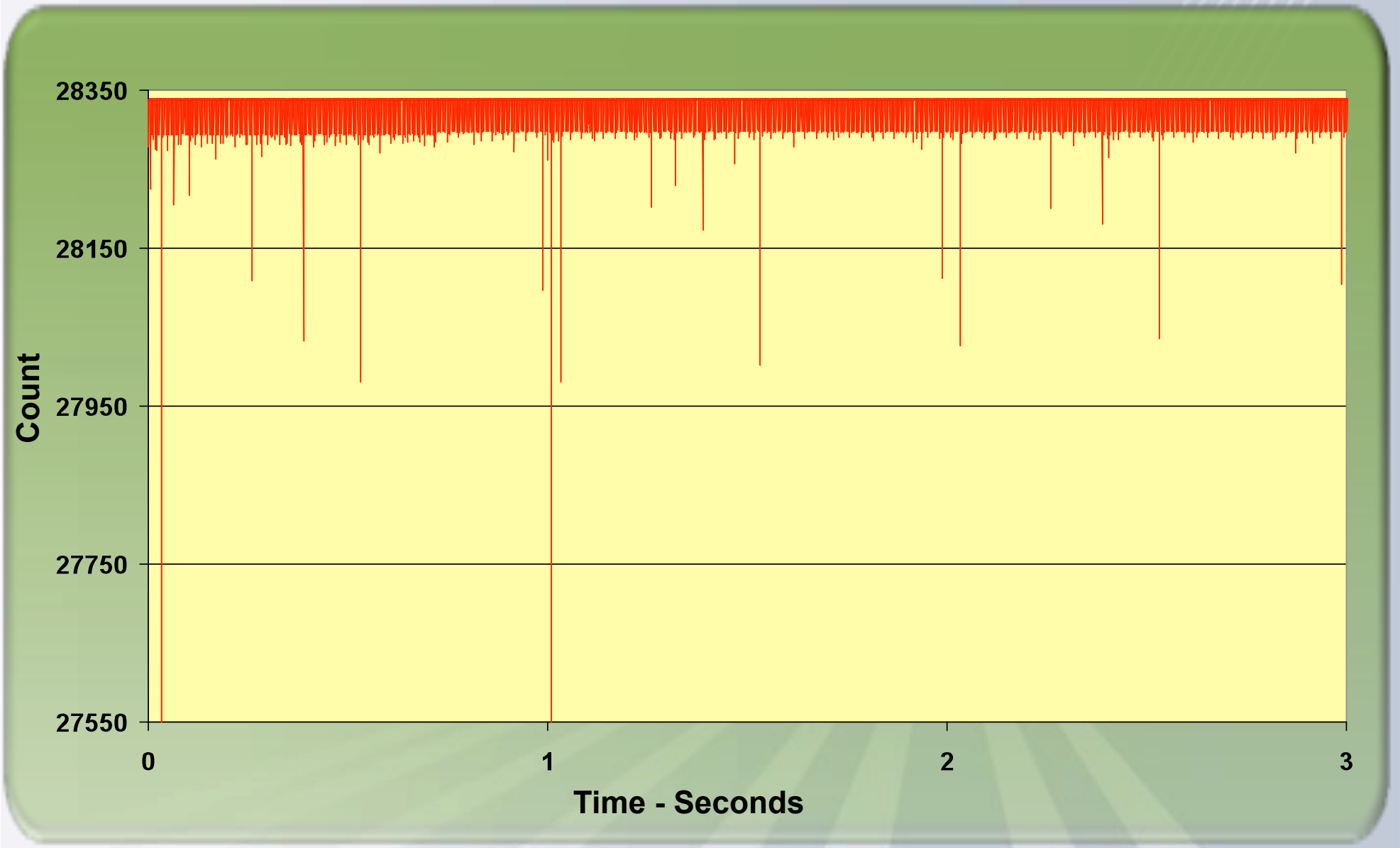


Results of study by Ron Brightwell, Sandia National Laboratory comparing Lightweight Kernel vs. Linux on ASCI Red System

Sandia National Laboratories

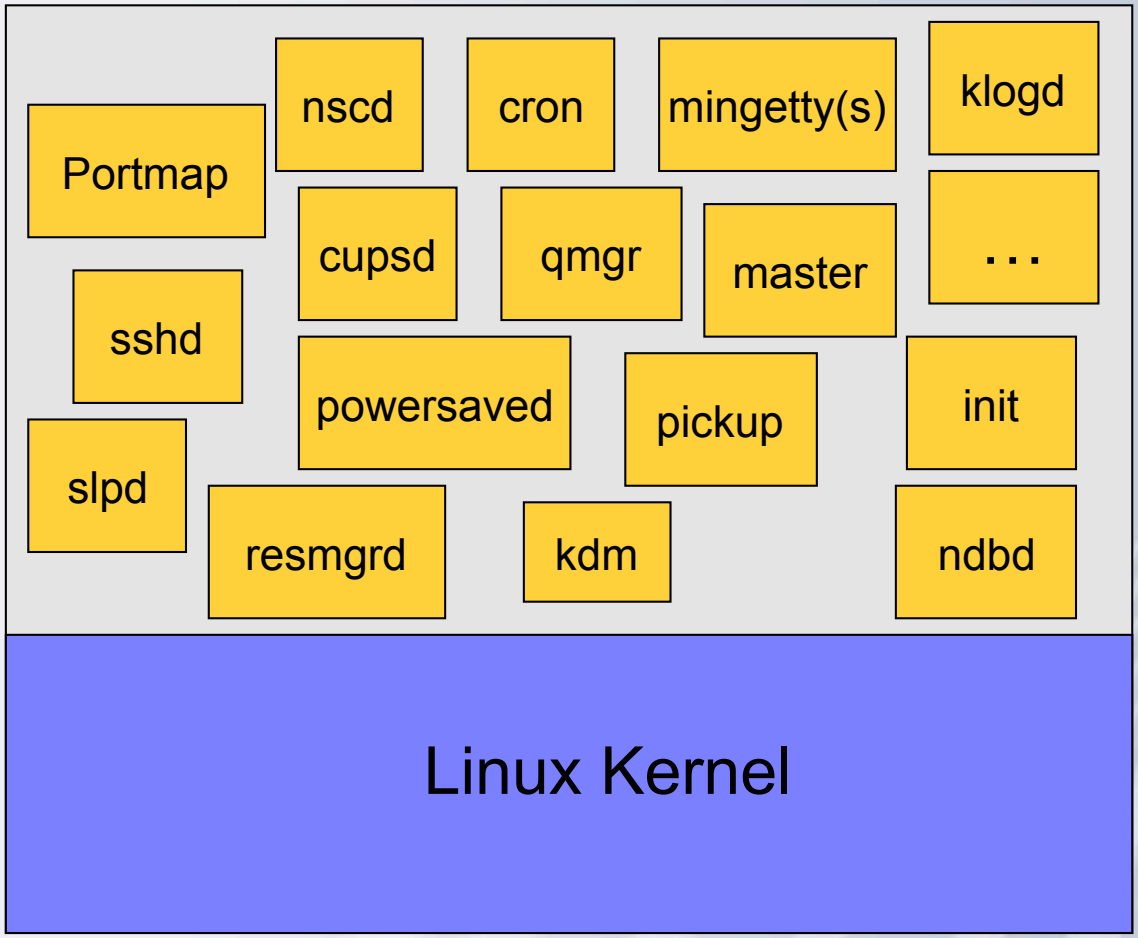
FTQ Plot of Catamount Microkernel

FTQ Plot of Stock SuSE (most daemons removed)

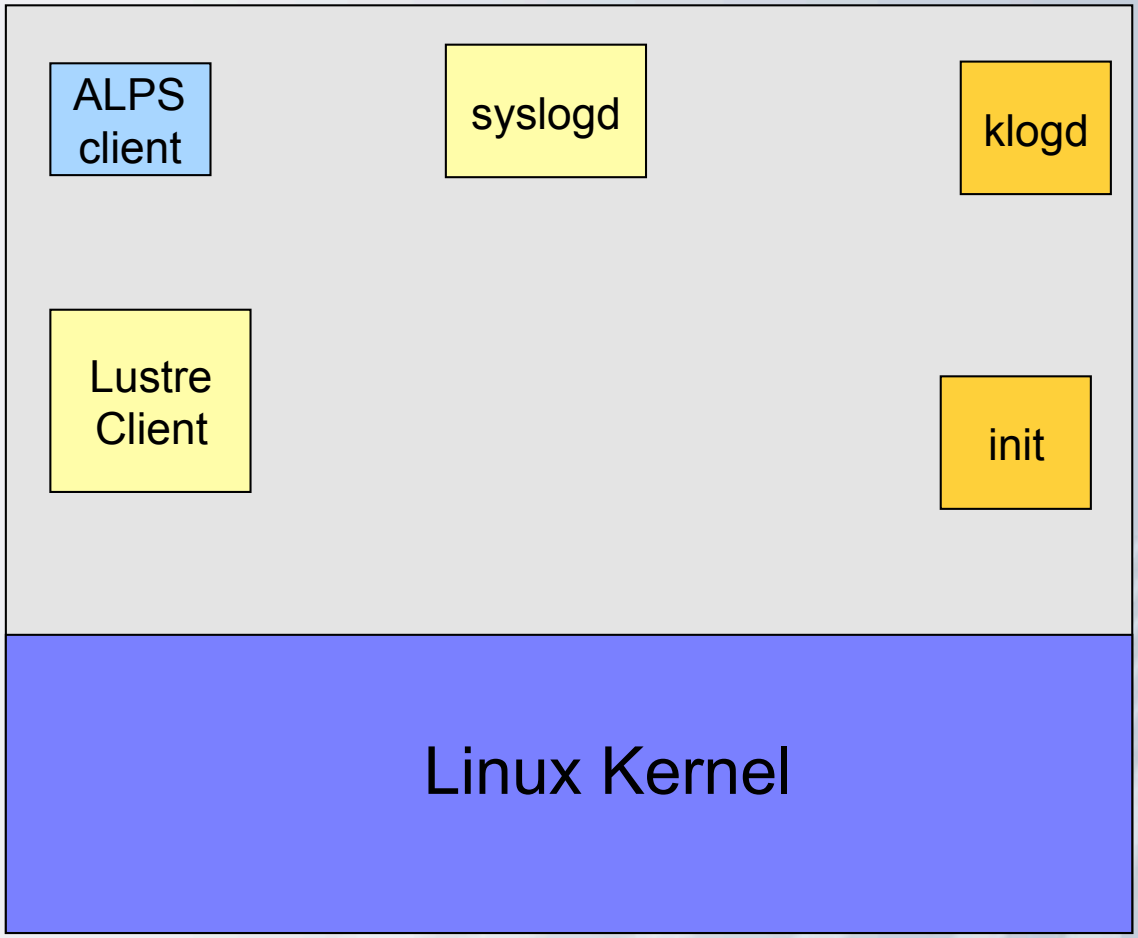


FTQ plot of CNL

Trimming OS – *Standard Linux Server*



Linux on a Diet – *CNL*



Compute Node Linux – Capability vs Capacity

Light on functionality

Heavy on functionality

Capability (Ultra-light Linux Image – with only the services required to support application processes, sockets,OpenMP,most POSIX requests)
Scale >30K nodes

Capacity
(Mid-weight Linux Image – some services and more mounted file systems – dynamic libraries)
Scale < 1K nodes

Shrink-wrap
3rd Party Application
(Full Linux Image and all services)

High scale

Low scale