Resource Limiting Strategies in Verio's Virtual Private Server

Fred Clift

Manager VPS Development

801-437-7471

fclift@verio.net

Build on us.

VERIO

An NTT Communications Company

Who is Verio? Who am I?

- **Internet Hosting Pioneer**
- **Business Unit of NTT**
- **What do we use FreeBSD for?**
  - *Virtual/Managed Private Server (VPS/MPS)*
  - *Signature Hosting line (traditional shared hosting)*
  - *Infrastructure*
  - *CPS (even our power strips run FreeBSD!)*
- **Manage Dev team for VPS/MPS products**

Overview – why are you here?

- **Not directly about Jail(8) (yet!) Verio's background.**
- **Why limit?**
- **User/Software Perception + examples**
- **Techniques**
- **What Verio can do for you**

Intro

Buzzword of the week: Virtualization, Multi-tennancy, Software as a Service, Virtual Appliances, Platform as a Service

What do they all have in common? Share a computer with uncoordinated, competing applications. (compare to big-iron running a single app)

Examples: Traditional Internet Hosting (FAMP), server consolidation, virtual dev/test environments, preconfigured SaaS application deployment

Why resource limit ?

3 Virtual partitions on your server

From the Application/User perspective:

- As an application, how do you handle being out of RAM? Disk space? Life sucks.
- Less performance
- The flipside: Predictable performance for all
- “Large Startup” apps
- Burstiness! The Magic Bullet

From the Physical Server/Provider side

- Try to share physical resources fairly, or better, unfairly aka “proportional”.
- Large Startup apps - e.g. JVM - You can’t set memory limits usefully low enough (little shared code space, large absolute usage)
- For a limit to be useful, you need steady-state, long-term to be restrictive
- Overcommit (statistics or application knowledge help!)
- Burstiness! The Magic Bullet

Burstiness – The Magic Bullet

Example: Disk Bandwidth

Ensure each of 30 virtual FreeBSD instances has <u>some</u>

- **30 MB/Sec (mediocre hardware…)**
  - *Split this between 30 Virtual FreeBSD boxes*
- **Naive way - Low limits- Limit each instance to 1MB/sec**
  - *Achieves desired effect*
  - *Performance always terrible*
  - *My 10 year old ATA drive does better!*
- **Better way - Overcommit- limit instance to 10 MB/sec**
  - *Achieves desired effect*
  - *Performance seems mediocre, but passable*
  - *My 10 Year old ATA drive does better!*
  - *Can't "Ensure" performance - best effort – 3 instance have to all be hogs to saturate – look at stats*

Burstiness – The Magic Bullet

Even better way - Burst limits

- Allow applications to burst
- Limit long-term/steady-state to 10%
- Achieves desired effect
- End User Perspective usually good
- Takes advantage of natural burstiness in applications
- Still prevents Resource Starvation for long-term abusers
- How?

Examples of Burstiness

- Any periodic process - pop/imap of email
- Temporal Locality in website access
- Builds on servers ‘make buildworld’
- Incoming mail with ClamAV/SpamAssassin

What to limit?

- **Anything people/applications use, or abuse**

- **Traditional ones (man getrlimit):**
  - *CPU time*
  - *Disk space*
  - *Per-proc Memory usage*
  - *File descriptors*
  - *nproc*

- **Others**
  - *Disk I/O BW*
  - *Network BW*
  - *Syscall rate-limits - e.g. mysql runnaways*
  - *Mail queue injection limits (qmail) - spam spam spam*
  - *Multi-level quotas*

- **“small application tuning” - e.g. mysql/innodb**

HOW? Techniques

- **First, generally only limit virtual instances - leave physical server stuff unlimited, or even give it a preference.**
- **Figure out what to measure and calculate**
  - *Sleep the thread if the account needs to be limited*
- **Takes statistics and care**
  - *Will cause problems.*
  - *Signature NTT backup example*
    - 30 virtual instance of FreeBSD, 30 Gig disk quotas, 300G of usable space.
  - *Syscall rate-limit example.*
  - *Disk-IO/nproc example*

Techniques…

- **Modify limiting system to use some bursting measure, combine with overallocation - Burstiness**
  - *Still need to understand your users/applications*
  - *Need stats, but it’s more forgiving*

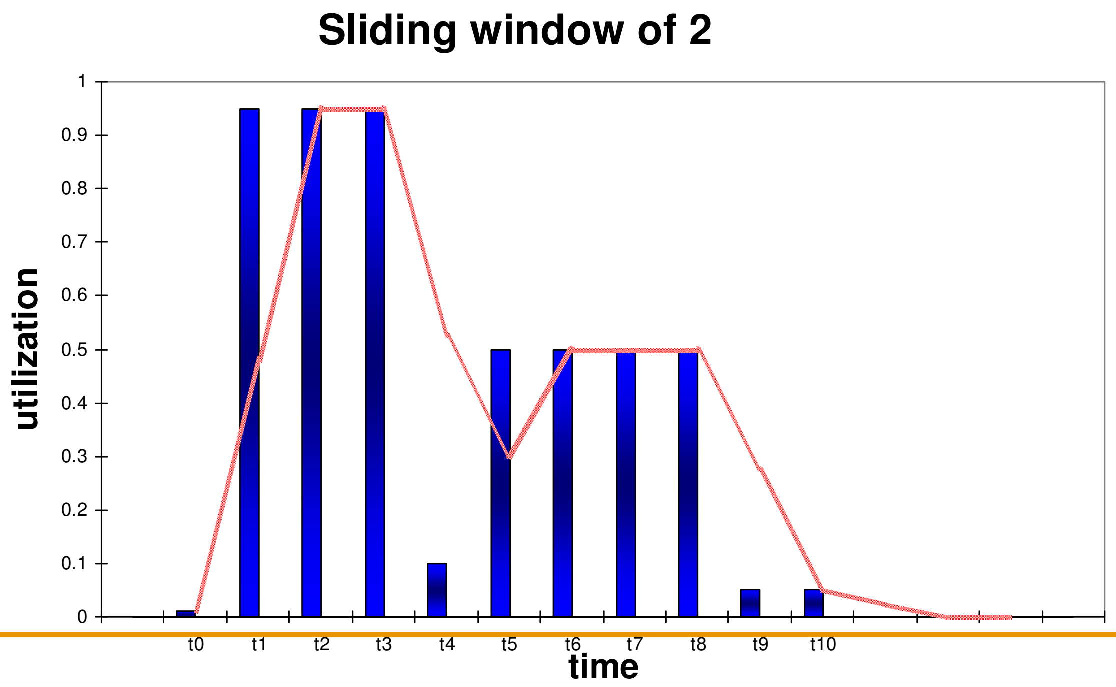
- **Two ways we do burst-based limits “shaping”.**
  - *“load average” bursting*
  - *Variants on Token-Bucket*

Load Average based shaping

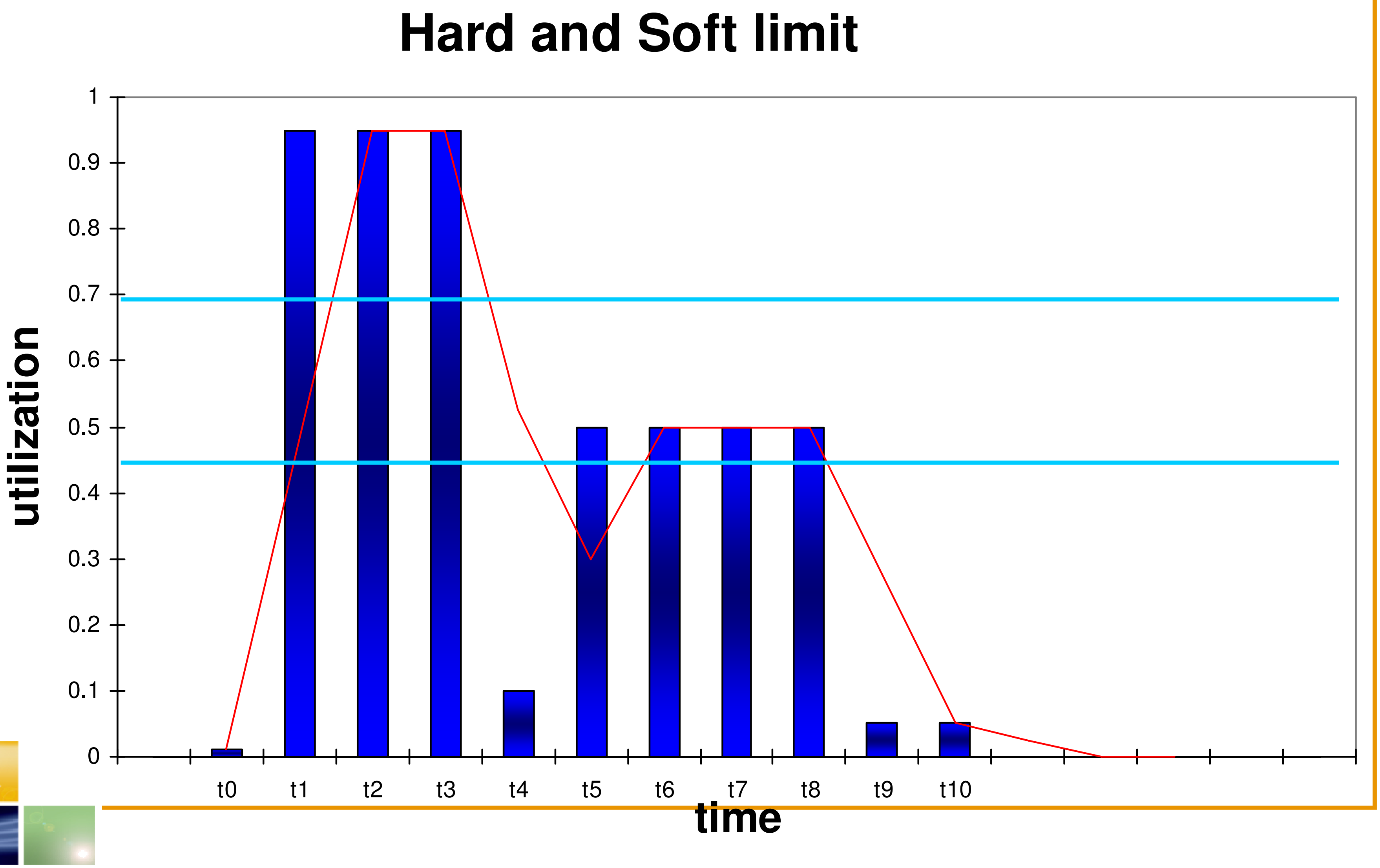
Uses same “exponential decay sliding-window average” that FreeBSD uses to calculate load average

Simple to calculate estimate of recent usage

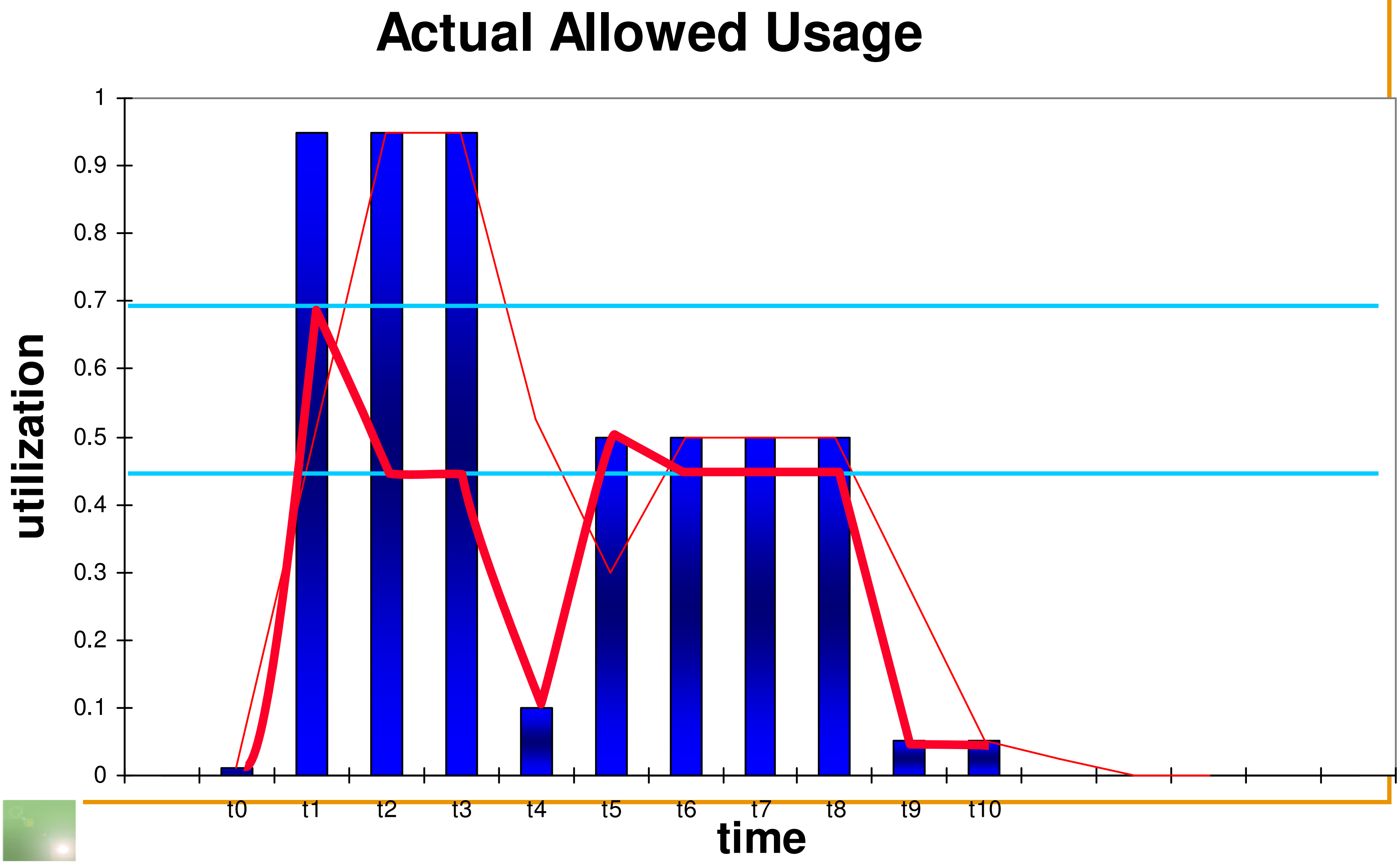
Sort of Integration/Area under curve of samples in a specific time window



Load Average based shaping



Load Average Based Shaping

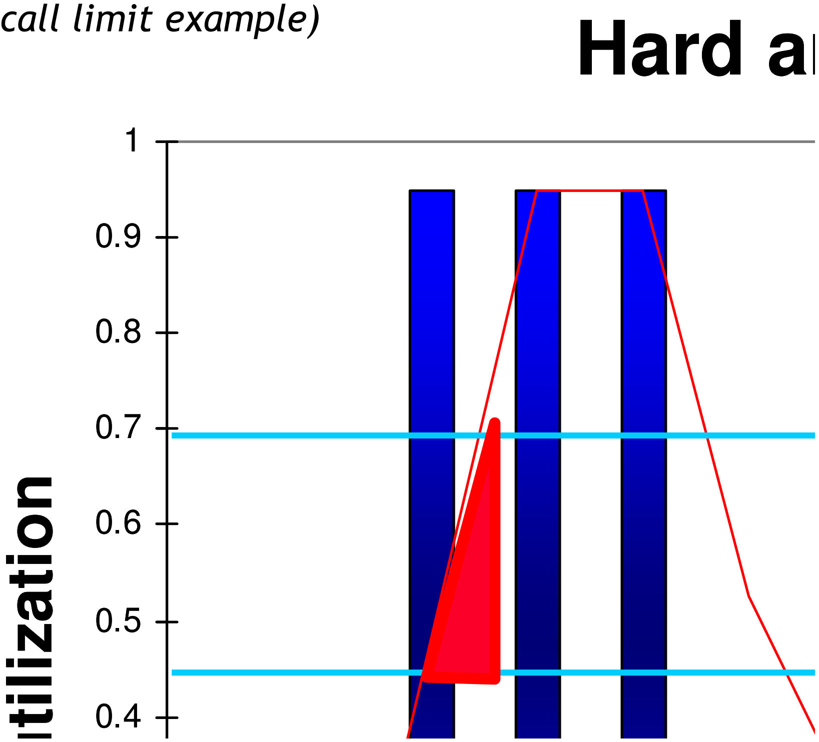


Load Average based Shaping

- Use standard FreeBSD function for calculating usage
- Has been used for Network Bandwidth Disk I/O, Syscall Rate-Limit, kind of CPU
- Specify a Hard limit - can never excede - short term burst to this limit, and a Soft limit - long term steady-state under demand.
- Simple to calculate, hard to know where to insert the checks for shaping - locking.

Load Average based shaping

- **Two main negatives**
  - *Hard to explain/understand/tune*
  - *The burst time is proportional to the ratio of Hard and Soft (syscall limit example)*

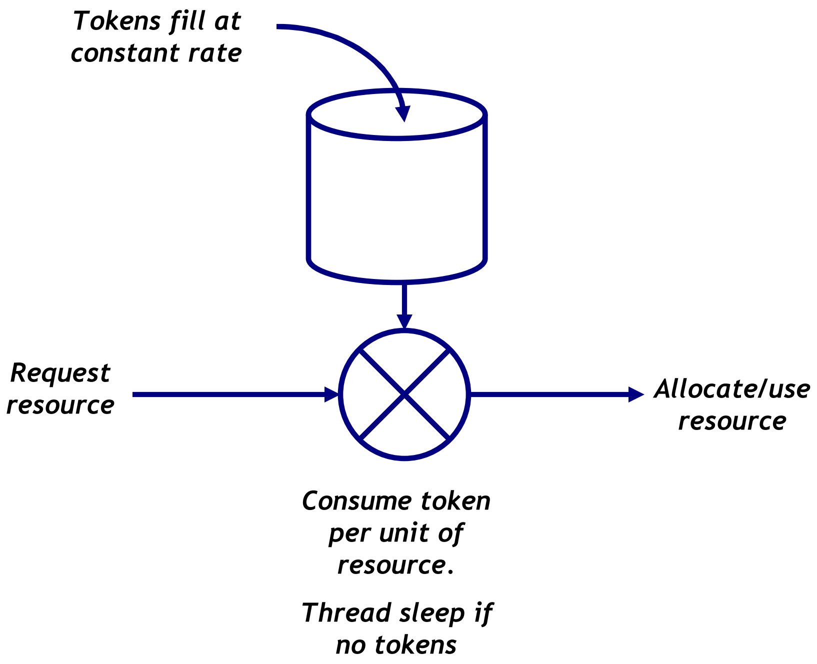


Load Average based shaping

- Possible fix - add third parameter to specify window size (complicates the algorithm, adds a 3rd parameter to tune)
- Possible fix - replace the algorithm with popular Token Bucket implementation
- The primary algorithm we still use - slowly replacing.

Token Bucket based shaping

- Each operation that consumes resources also consumes a token.
- You have a fixed-size bucket being filled at a fixed rate
- If your bucket is full, it ‘overflows’ - tokens discarded
- Two tuneables - Fill rate and Bucket size.
- No limit on short term burst rate
- Long term burst rate dependent on bucket fill rate



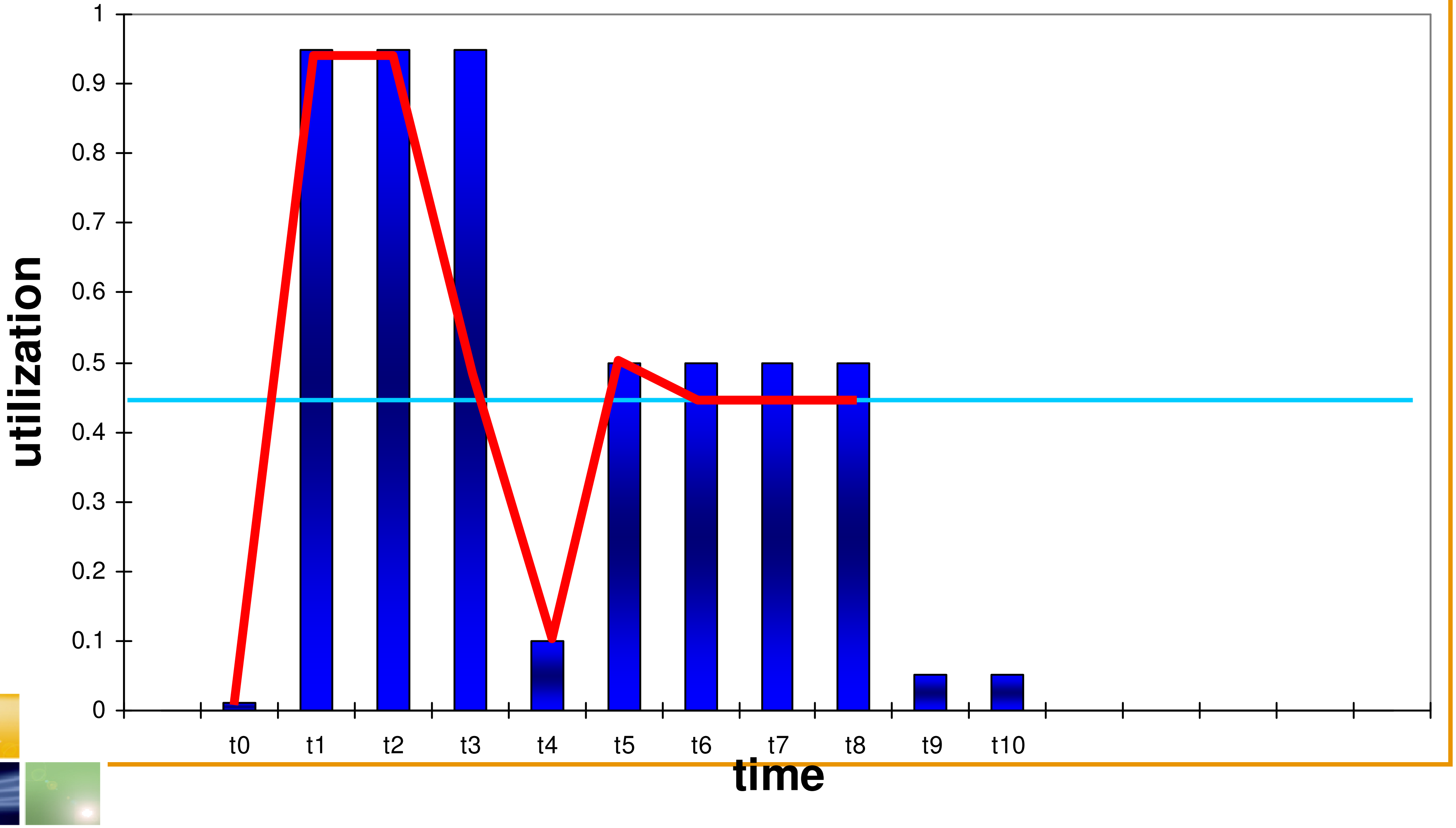
Tokens fill at constant rate
Request resource
Allocate/use resource
Consume token per unit of resource.
Thread sleep if no tokens

Token Bucket based shaping

- Simple calculations
- Easy to explain the metaphor
- Easier to tune than Load Average shaping
- Burst time is dependent on bucket size

- BUT, no short term rate limit (can be extended - use a drain rate at the cost of extra complexity, use leaky-token-bucket)

Token Bucket example



What Verio is Doing

- **BSD license on our Freebsd (4.x, 6.x 7.x) mods**
  - *Waiting on lawyers*
  - *We're (Verio Developers…) eager*
  - *Not useful unless we merge*
- **Merging with (very similar) Vimage framework**
  - *Resource measurement/limits*
  - *Userland framework? Probably need something new*
  - *Virtlink system/virtual mounts – unionFS merge? Fix?*
- **When? RSN**
- **What else are we doing? ISCSI initiator, DTrace, Kernel, Peter Holm's Kernel Stress test suite**

Questions?

- **Get a copy of this at:**
- http://clift.org/fred/bsdcan2008.pdf

- **Contact me:**

  Fred Clift
  fclift@verio.net