
# Comparison of the Three CPU Schedulers in Xen

Lucy Cherkasova (HPLabs)
Diwaker Gupta (UCSD)
Amin Vahdat (UCSD)

# Motivation

- Effective management of virtualized IT environments relies on
  - dynamically resizing VMs
  - migrating VMs to different nodes in response to changed conditions
- The capacity management methods should work in ensemble with underlying resource allocation mechanisms
- Three CPU schedulers in Xen:
  - What are required features for management tools like VSE and gWLM?
  - Application performance sensitivity to different CPU schedulers and their parameters

# History of three CPU schedulers in Xen

- BVT: Borrowed Virtual Time
  - *Lack of non-work-conserving mode*
- SEDF: Simple Earliest Deadline First
  - *Lack of global load balancing*
- Credit: a fair share proportional scheduler
  - *Is this an ideal scheduler?*

- Understanding CPU scheduler features and scheduler performance is critical for efficient resource management in the consolidated environment

# BVT: Borrowed Virtual Time

- ***Proportional share*** via setting different domain weights
  - Example:
    - Dom1: weight 1 (20%)
    - Dom2: weight 3 (80%)
- ***Work conserving***: if only one domain has work to do – it can get all the CPU (i.e. we can not limit the CPU usage to, say, 50% only)
- ***Low latency support*** for I/O intensive and real-time applications: analogy of "priority".
- Fair-share scheduler based on the concept of virtual time
- Context-switch allowance C: real time, flexible (default: 5 ms period).
- ***Optimally-fair***

# SEDF: Simple Earliest Deadline First

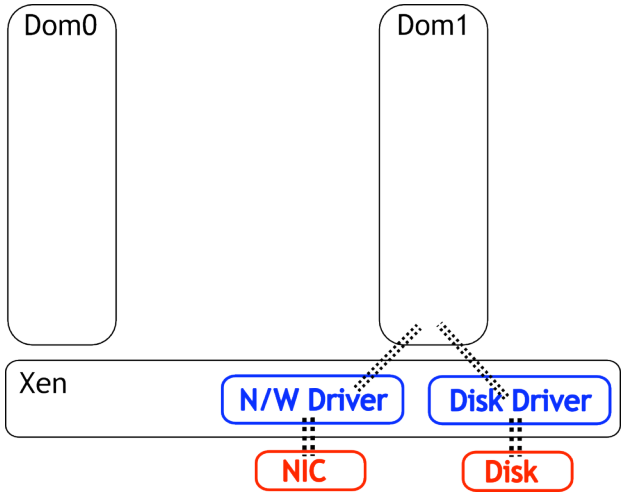
- Support for both ***work-conserving*** and ***non work-conserving*** modes
  - now we can allocate, say, 50% of CPU to the domain and not more.
- ***Real-time*** apps support: based on the notions of slice and period (flexible)
  - CPU allocation: 20 ms slice in each 100 ms period;
  - CPU allocation: 2 ms slice in each 10 ms period.
- Preemptive
- ***Limited fairness*** within the period.
- Per CPU scheduler: ***no global load balancing***
  - CPU1: dom1 – 80% CPU usage
  - CPU2: dom2 --- 80% CPU usage
  - dom3 with 30% CPU usage can not be allocated and served, each CPU has only 20% of available CPU share.

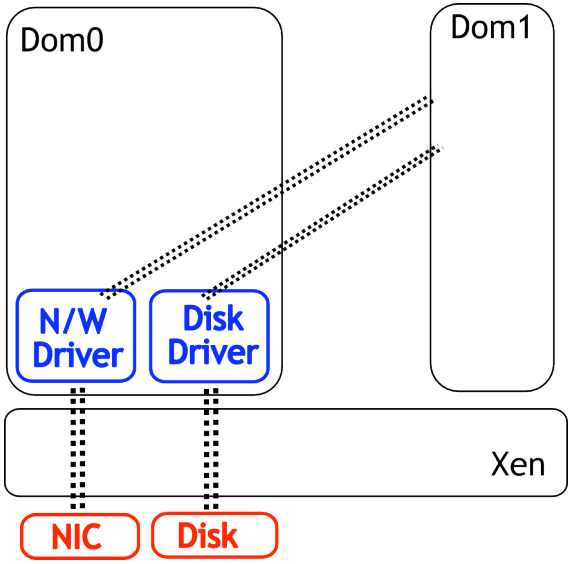
# Credit scheduler

- Fair-share scheduler
- Support for both ***work-conserving*** and ***non work-conserving*** modes using weights
- ***Global load balancing***
  - Now we can allocate dom1 (80%), dom2 (80%) and dom3 (30%) on 2-way CPU machine.
- Scheduling period is 30 ms (hard coded).
- Non-preemptive

# Two Popular I/O models

- Device drivers are hosted and executed within a hypervisor (VMM), e.g., the first Xen implementation, the VMware model
- Unmodified device drivers are hosted and executed in the privileged management domain: *Domain0,* e.g., the current Xen implementation.
- *I/O* processing is done by two components: *Domain0* and the guest domain.
- Performance of I/O intensive applications depends on CPU allocation to *Domain0* and the guest domain.



# Challenges

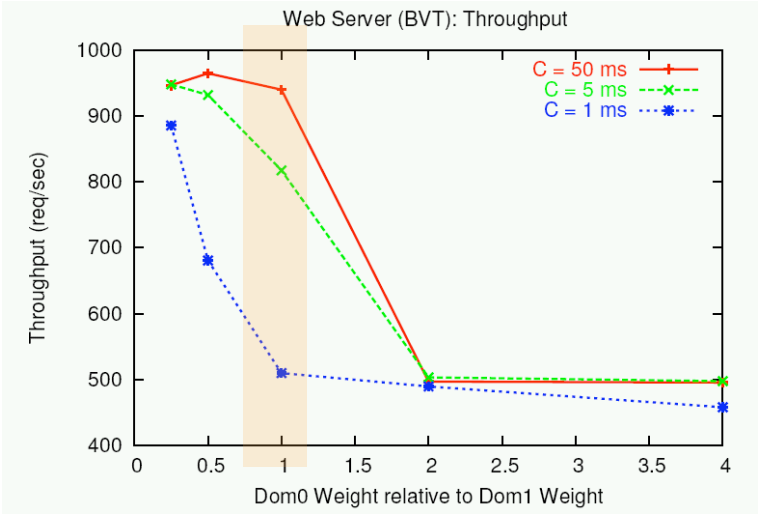
- How does one estimate the application CPU requirements and project them into two components: Dom0 and the guest domain's shares?
- How sensitive are I/O intensive applications to the amount of CPU allocated to Dom0?
- How significant is the impact of scheduler parameters?
- Does allocation of a larger CPU share to Dom0 mean a better performance for I/O intensive apps?

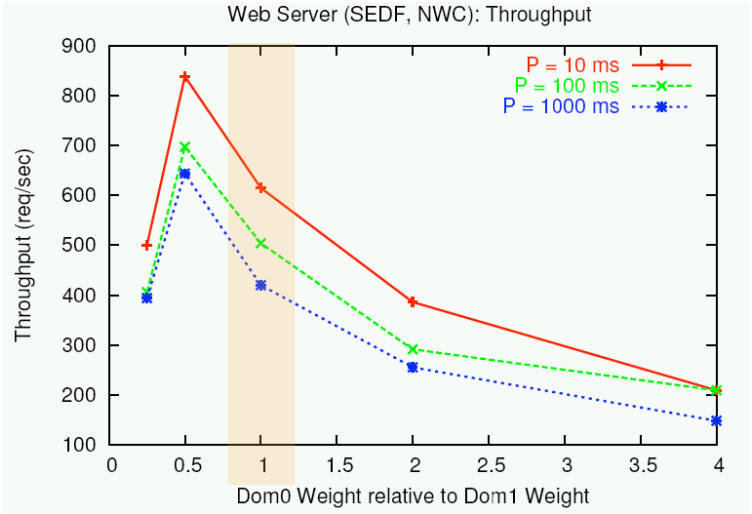
# Scheduler Parameters and Dom0 Weights

- I/O intensive applications are highly sensitive to Dom0 CPU share
- Scheduler parameters significantly change the scheduler behavior and application performance
- Analysis with XenMon:

| CPU Scheduler | $Dom_0$ Util | $Dom_1$ Util | ex/sec | i/o per ex | Tput req/sec |
|---|---|---|---|---|---|
| SEDF, P=10 ms | 27% | 50% | 451 | 34.6 | 615 |
| SEDF, P=100 ms | 35% | 50% | 4635 | 2.8 | 504 |
| SEDF, P=1000 ms | 40% | 50% | 7292 | 1.5 | 419 |

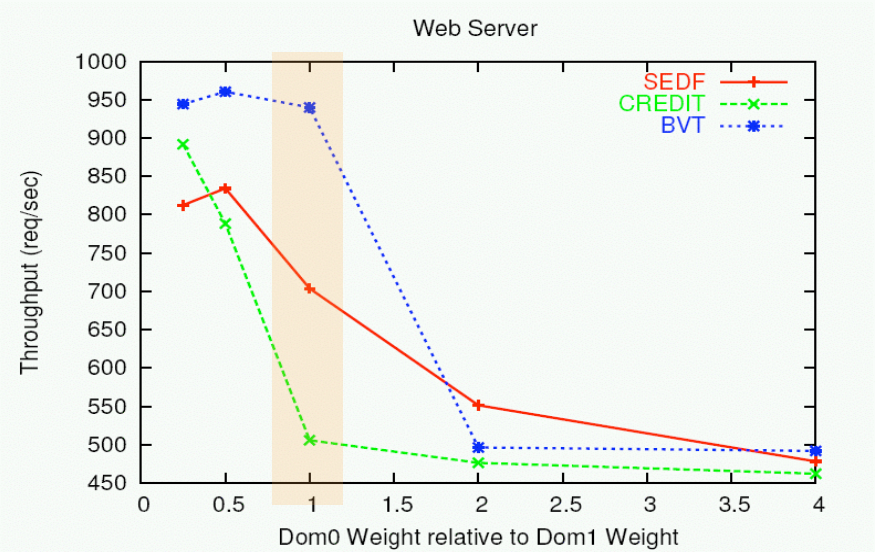
- Dom0 is scheduled much more often but it performs mush less “useful” work
- It leads to a high context switch overhead and worse application performance

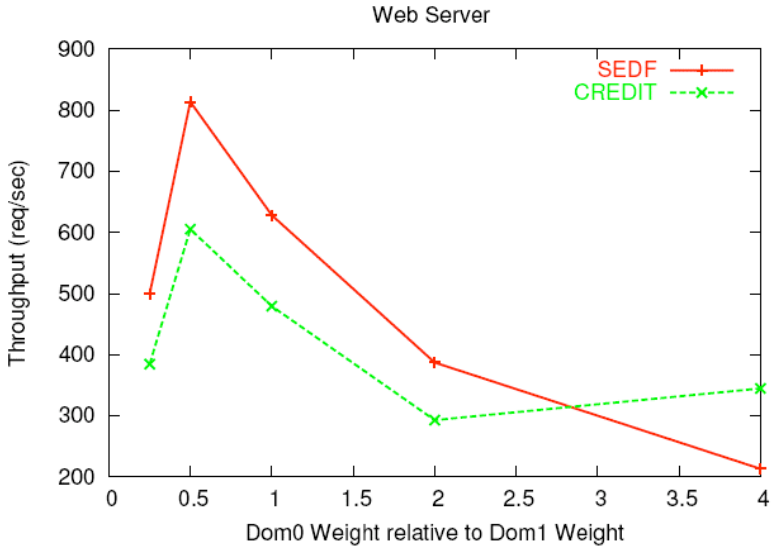


# Application Performance Sensitivity to Different Schedulers: Web Server

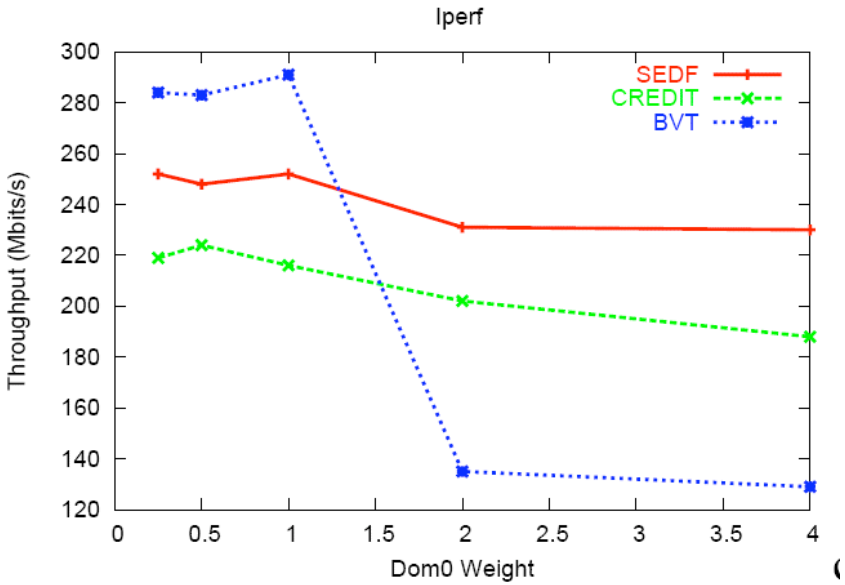
- Application performance can be significantly different under the different schedulers (*with the same CPU allocation share!*)
- These results vote in favor of more *homogeneous* environment when considering VM migration
- *Optimizing Xen scheduler performance for nwc-mode:* this mode is equired by HP management tools



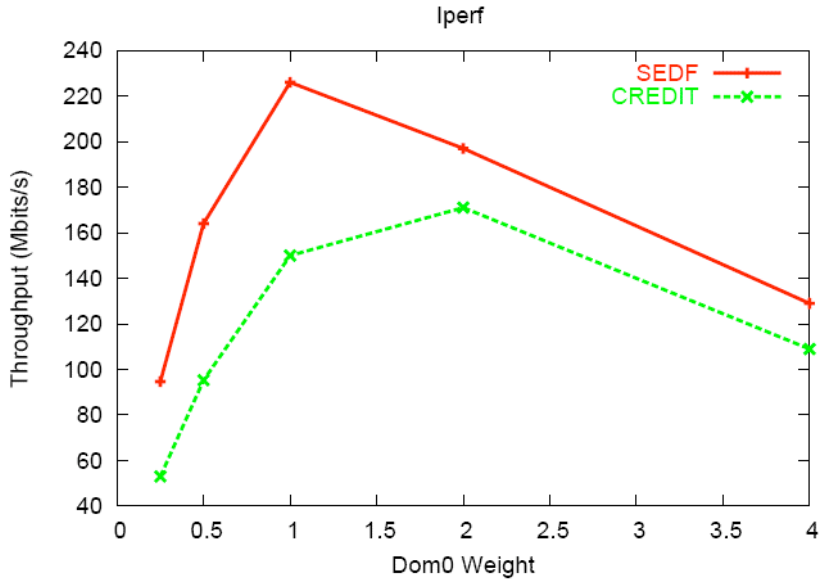



# Iperf

- Relatively flat *iperf* performance under SEDF and Credit in wc-mode

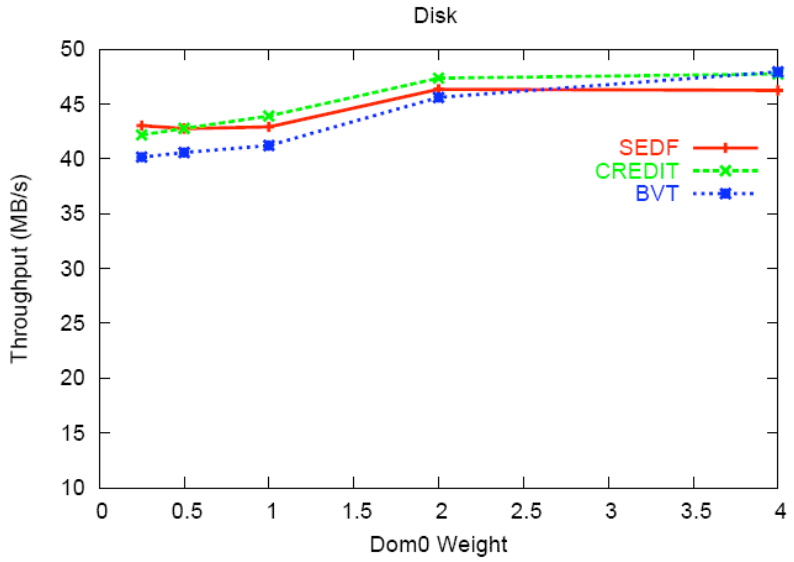



- *iperf* performance is very sensitive to Dom0 weight under SEDF and Credit in nwc-mode.

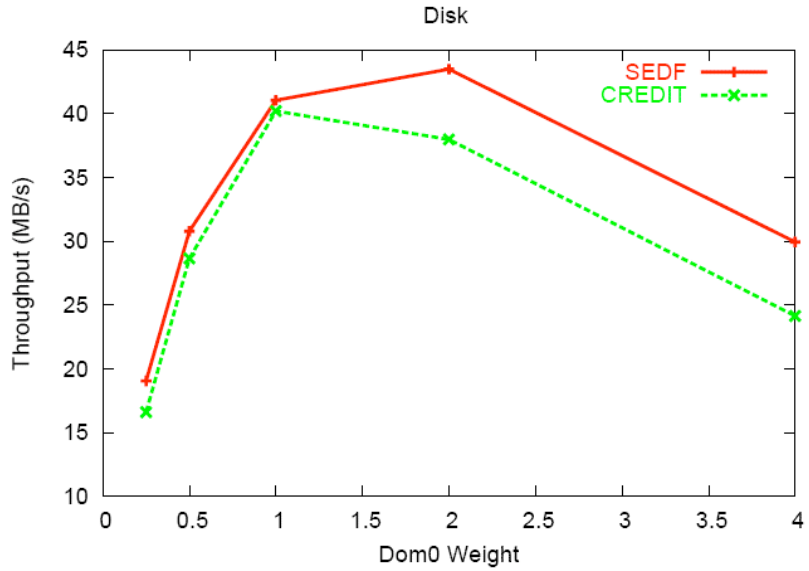


# Disk

- Similar *disk* throughput under all the three schedulers in wc-mode

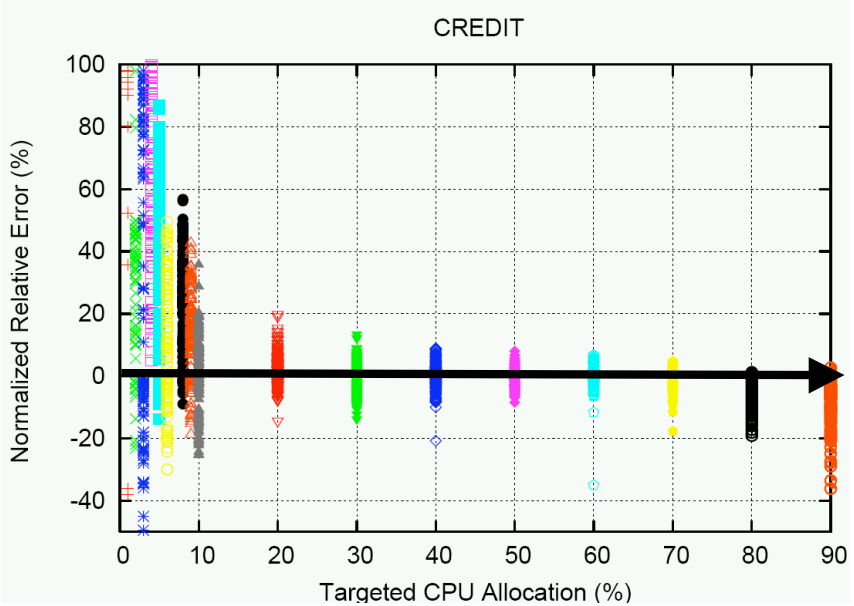


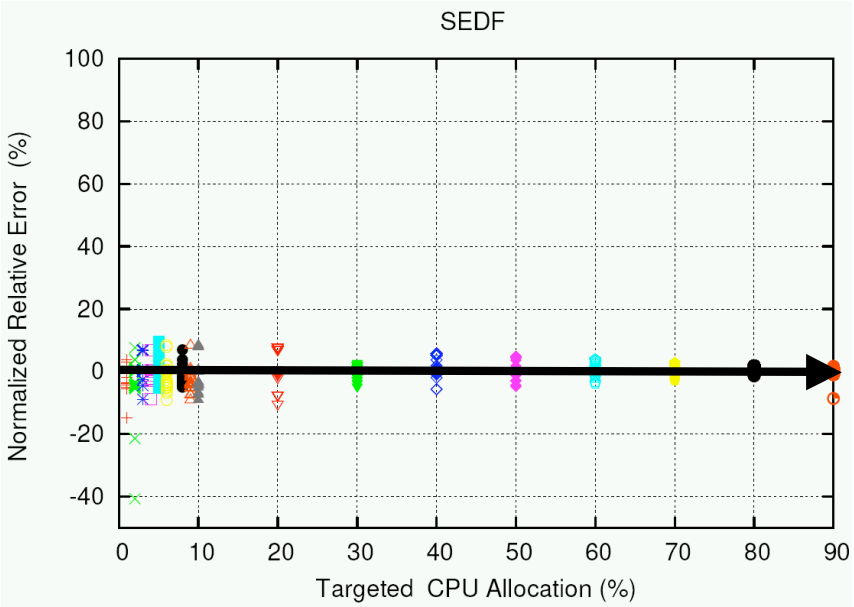

- *disk* performance is very sensitive to Dom0 weight under SEDF and Credit in nwc-mode.



# CPU Allocation Error

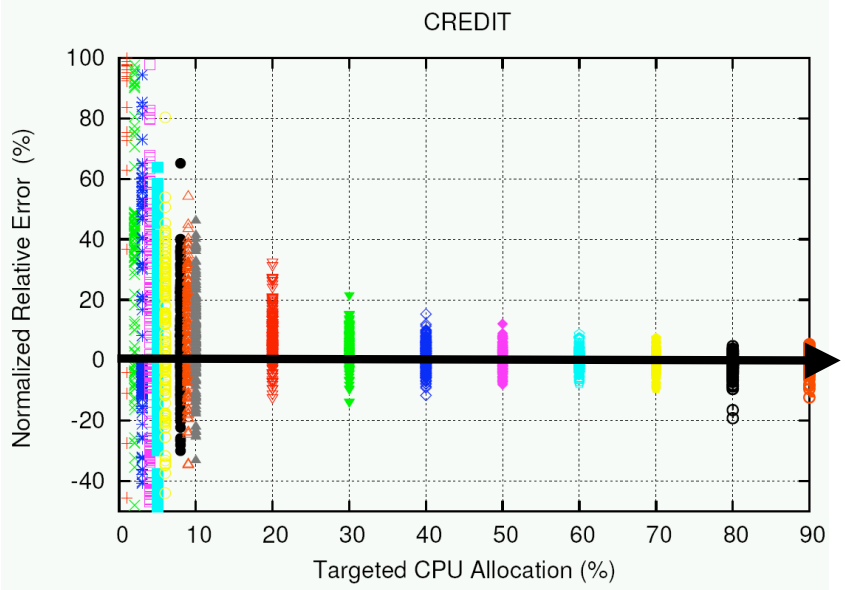
- ALERT: ALlocation ERror Test
  - Dom0 is allocated 6% (using cap mechanism)
  - Dom1 is allocated 1%, 2%, 3%,…,10%, 20%,….,90%
  - Dom1 executes the “slurp” program (tight CPU loop).
- Expectations for Dom1:
  - If X% CPU is allocated then X% of CPU should be measured during the test
- Credit scheduler shows high CPU allocation errors
  - Even longer time averages (3min) had 10% error
- Such high errors complicate the management
- Can lead to unstable controller behavior in gWLM

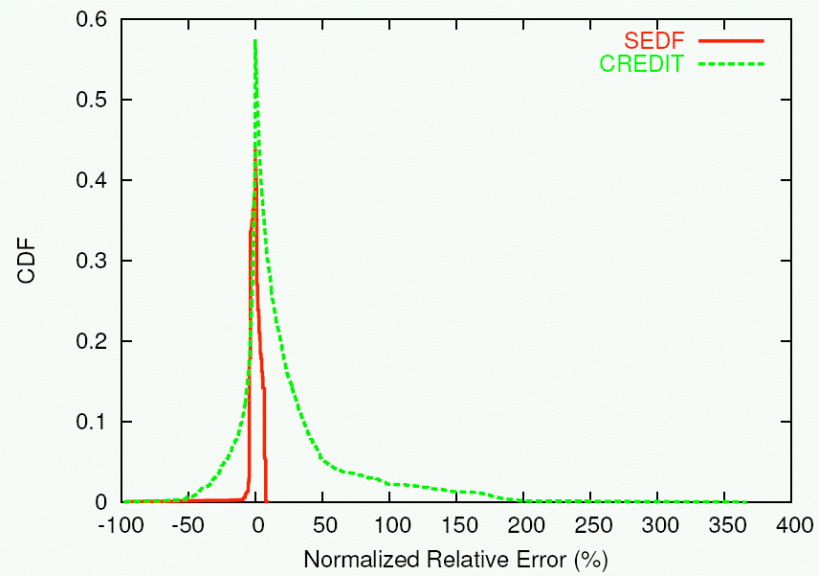


# Credit Scheduler Improvement

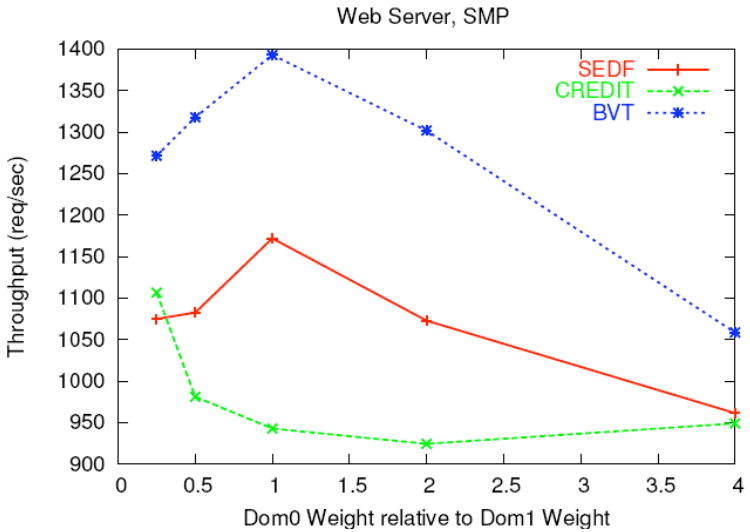
- Current Credit implementation has lower allocation errors.
- For CPU allocation <30% the Credit errors are still high, more improvement is needed
- Credit still has a much higher error compared to SEDF
- Longer time averages are significantly improved

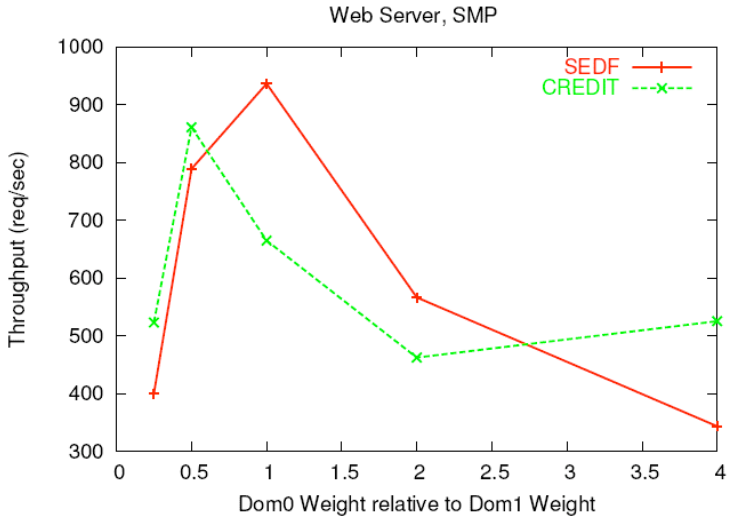


# SMP case: Web Server

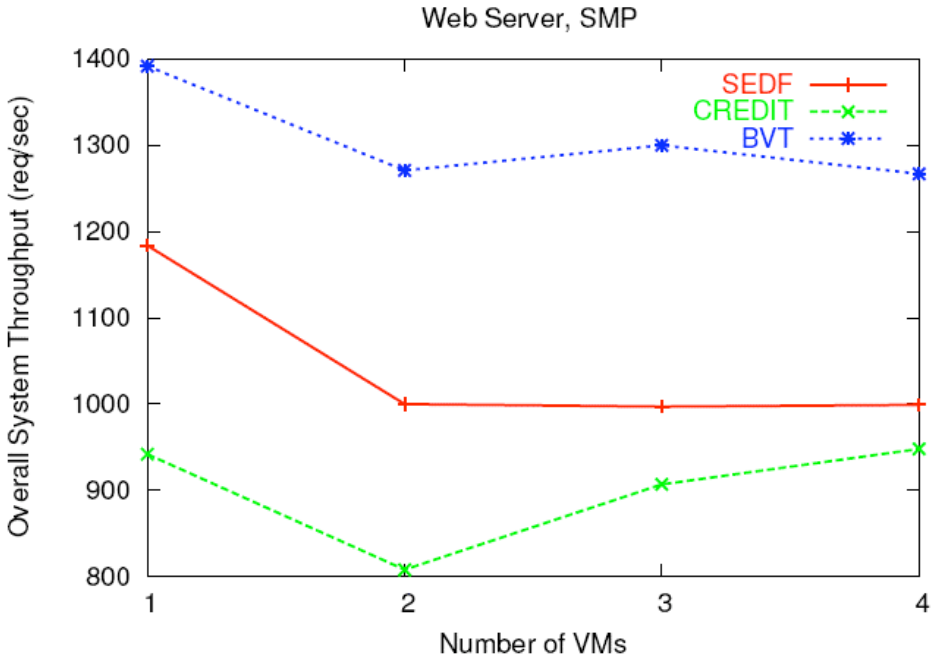
- Each domain (Dom0 too) is configured with 2 VCPUs
- Better load-balancing features of Credit can not “overwrite” its basic behavior for processing I/O intensive workloads
- Web server throughput increase compared to 1 CPU case:
  - BVT: 45%
  - SEDF: 30%
  - Credit:24%
- Web server performance is less sensitive to Dom0 weight in wc-mode
- Web server performance is very sensitive to Dom0 weight in nwc-mode



# Multiple Web Servers

- Small scaling experiment with multiple VMs: each is running a web server
- Small drop in aggregate throughput with 2 VMs
- Credit shows an improved performance for aggregate throughput with higher number of VMs (due to globa load balancing?..)



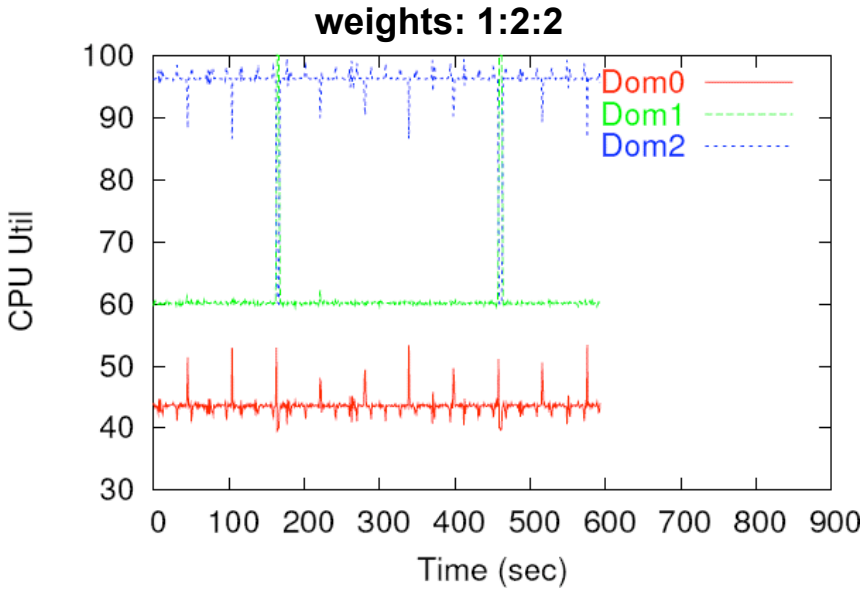
# BVT: SMP case

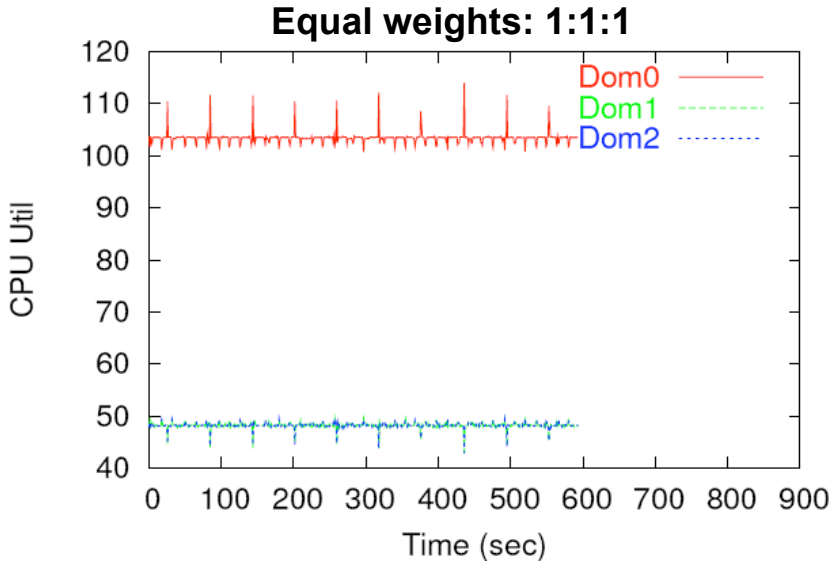
Experiments with slurp (single process)

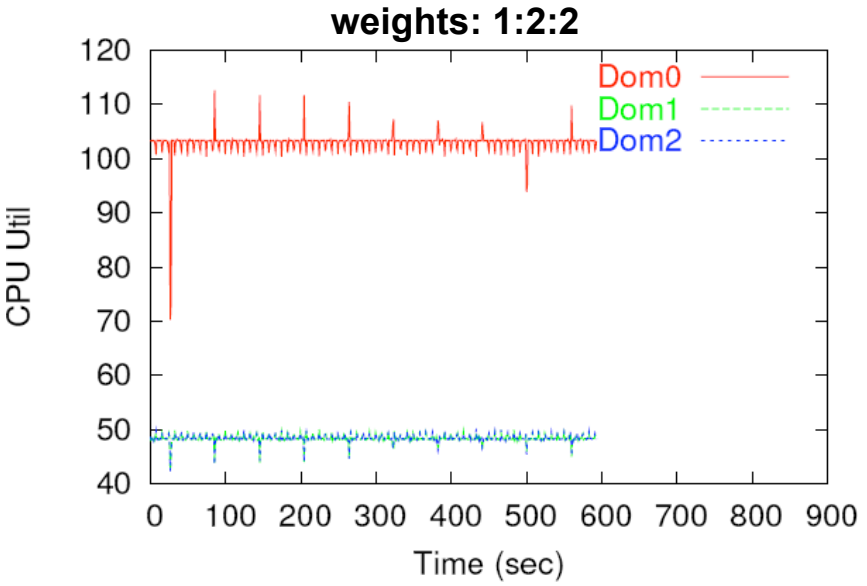
All the domains (including Dom0) run Slurp (tight CPU loop).

Current BVT implementation does not support global load balancing

VMs (VCPUs) are randomly assigned to physical machines and CPU allocation Is computed after that for each CPU.

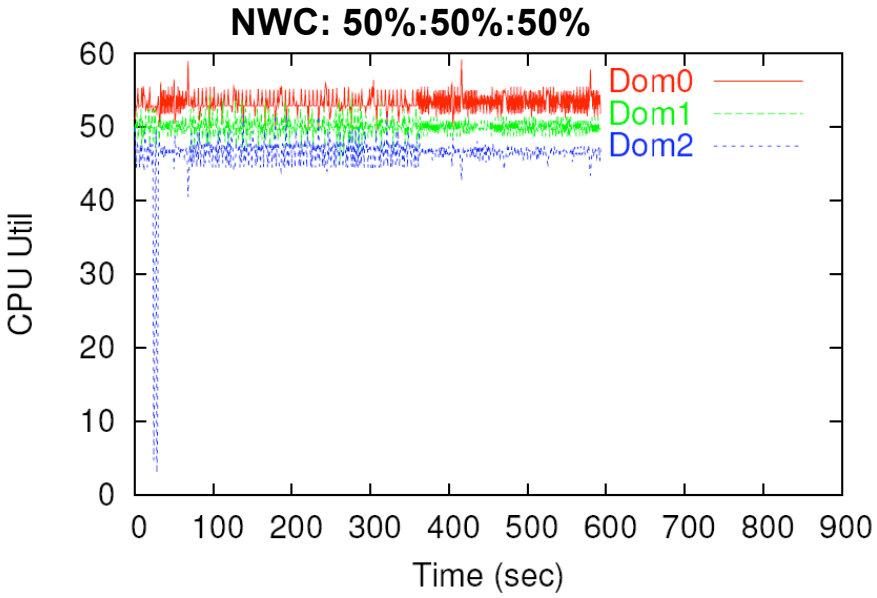


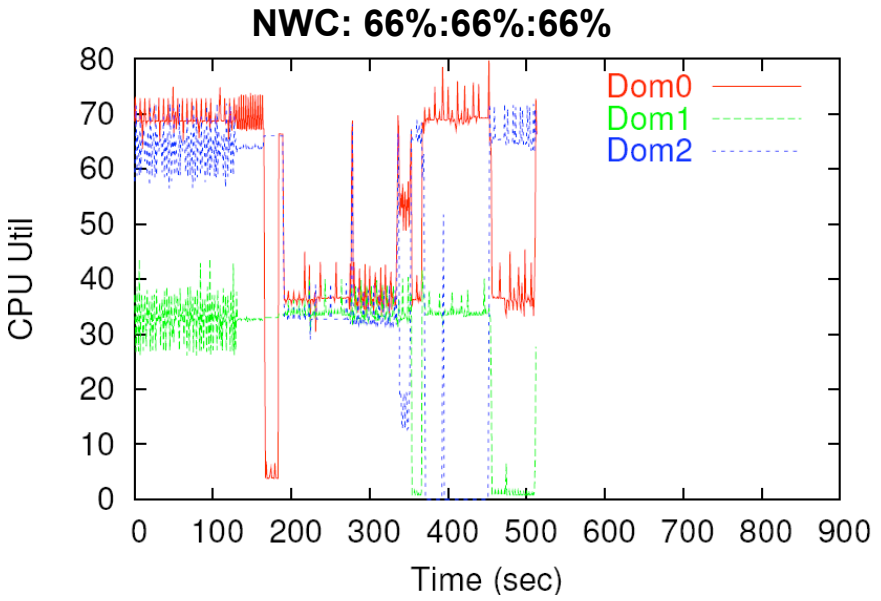



# SEDF: SMP case

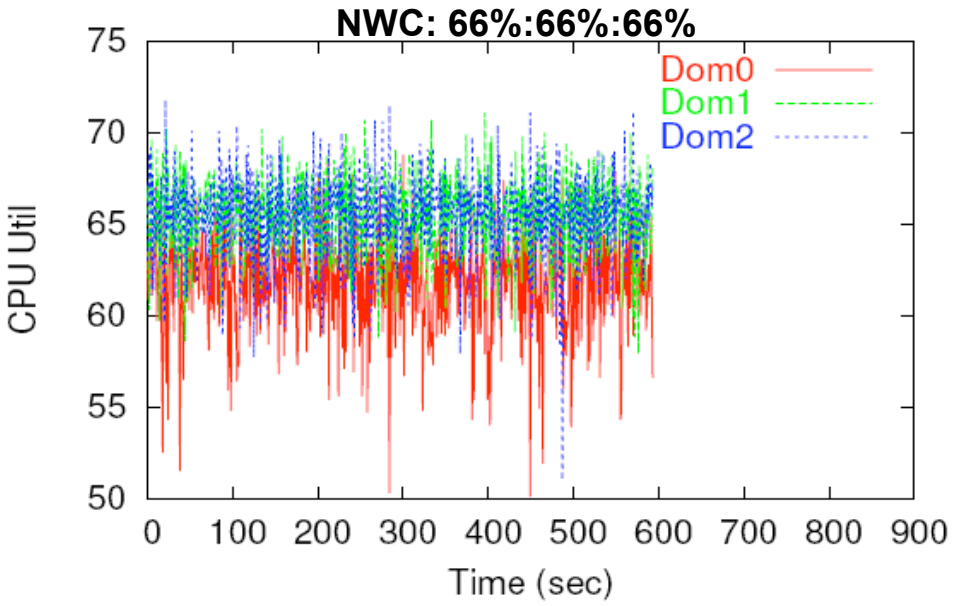
- SEDF has a lack of global load balancing
- It is especially apparent under nwc-mode (capped).
- Similar to BVT, it assigns VCPUs randomly to physica CPUs and tries to support CPU allocation at the CPU level.
- Clearly, 2 processes can not get 66% each at a single CPU.

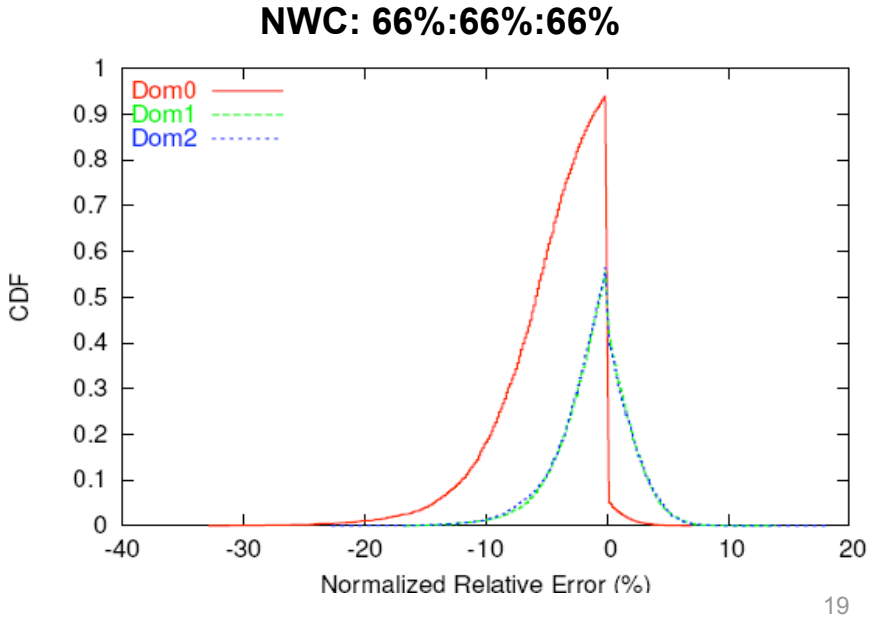


# Credit: SMP case

- Credit is a true winner among the three schedulers for supporting global balancing capabilities
- Still relatively high CPU allocation error
- Dom0 is “under-provisioned” (not clear why)
- These errors mostly introduced by the global load balancing
  - when we ran a similar experiment on 1 CPU machine the allocation error is much smaller.



# Summary

- CPU schedulers should be tested and augmented with ALERT results for different configuration
- Many enterprise management solution rely on the accurate CPU allocation by underlying virtualization layer.
- Challenges: how do we “tune” Dom0 CPU allocation for optimal application performance?
- How do we project application resource demands to a virtual world?
- L. Cherkasova, D. Gupta, A. Vahdat:
  *When Virtual is Harder than Real: Resource Allocation Challenges in Virtual Machine Based IT Environments.*
  **HPL-2007-25,** February, 2007.

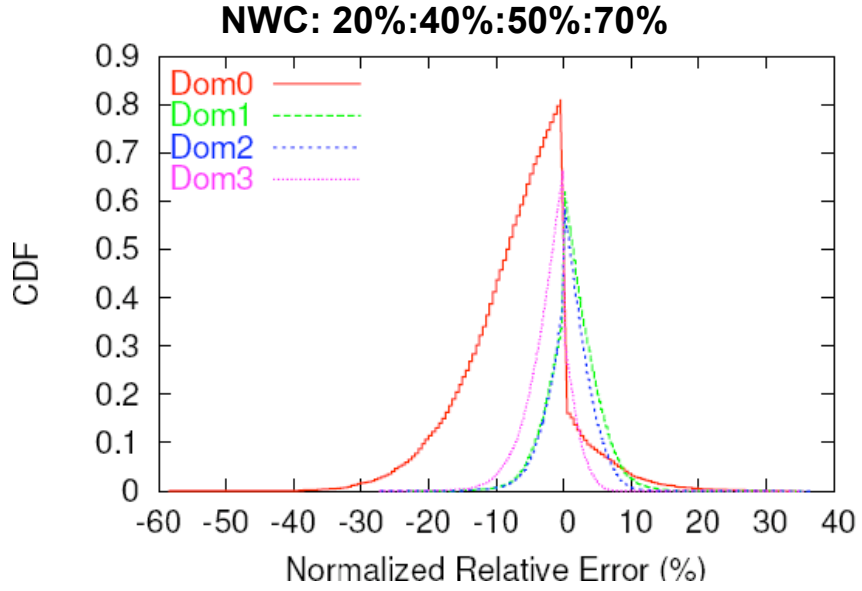
# Acknowledgements:

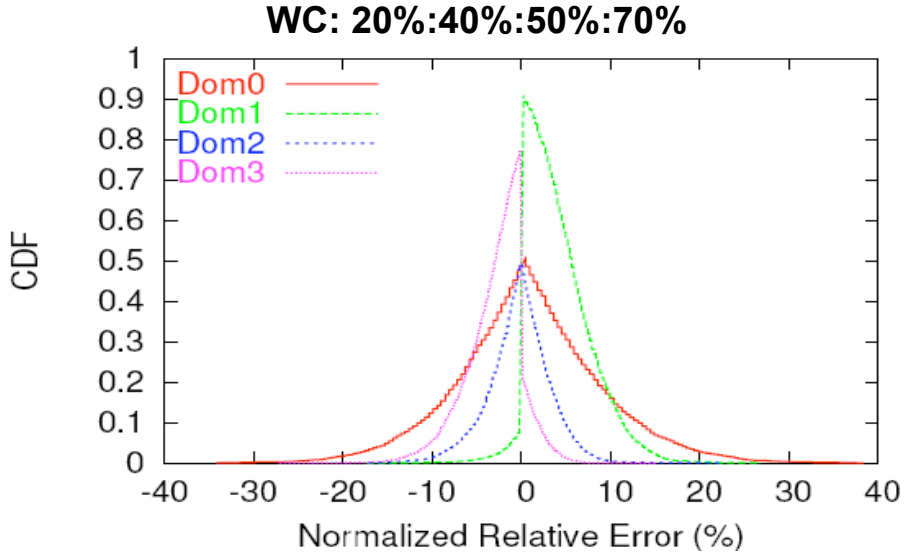

- Stephan Diestelhorst and Timothy Roscoe (SEDF)
- Emmanuel Ackaouy (Credit)

*Questions?*

# Credit (cont)

- A few more examples of CPU allocation error under Credit
- Dom0 is consistently "under-provisioned" when Credit is in nwc-mode (capped).
- Distribution of CPU allocation errors in wc-mode is more symmetric.



# Credit (1CPU case)

- CPU allocation error with multiple domains show much better results
- Why for a single domain tested with ALERT the CPU allocation errors are much higher?