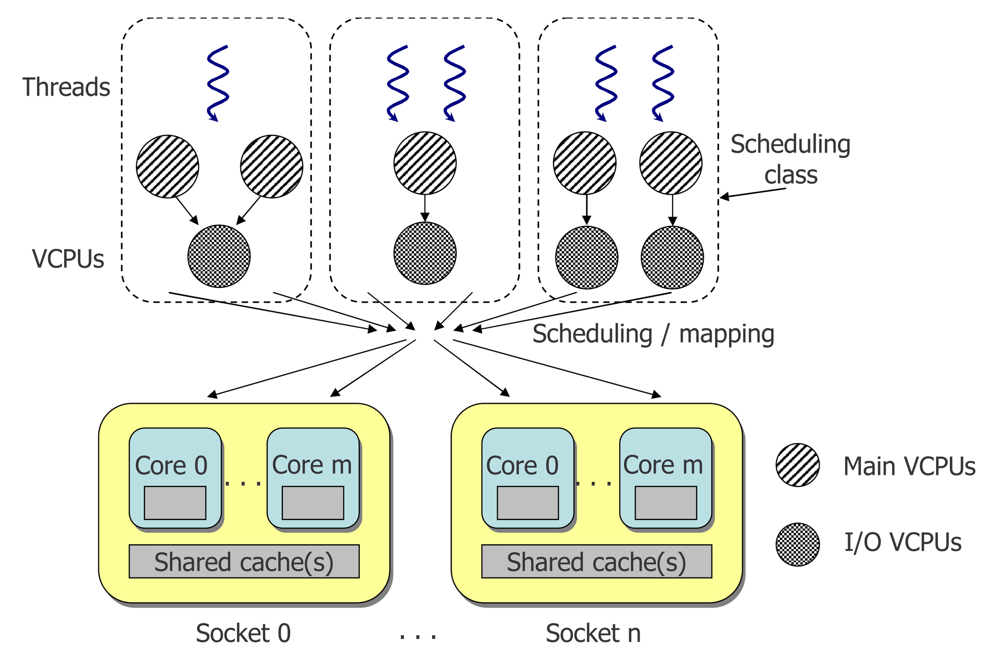
Virtual CPU Scheduling in the Quest Operating System

Matthew Danish, Ye Li, and Richard West

Introduction

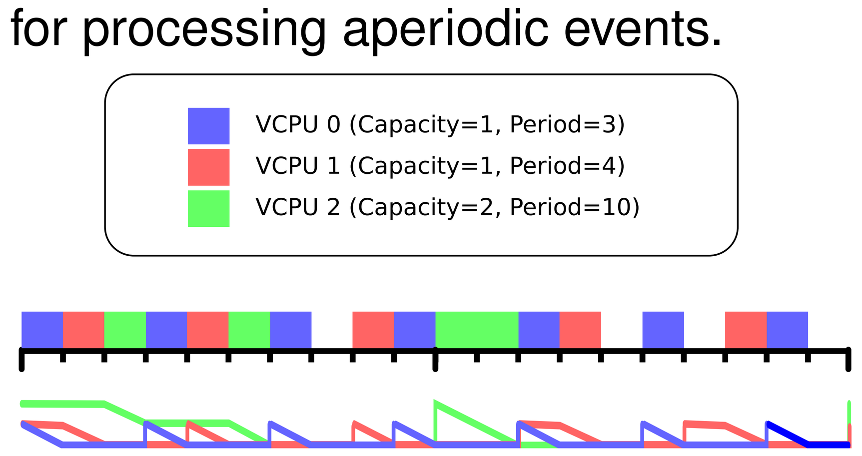
- Real-time scheduling is about ensuring predictability and temporal isolation.
- Quest is a research operating system developed at Boston University.
- Physical CPU (PCPU) resources are allocated to Virtual CPUs (VCPUs).
- Program threads are assigned to VCPUs.



- Main VCPUs are associated with normal thread execution.
- I/O VCPUs perform interrupt handling on behalf of Main VCPUs, for I/O processing.

Scheduling

- Main VCPUs are **Sporadic Servers** (SS) [Sprunt 89]: they have a budget that is replenished after an interval.
- Good for processing aperiodic events.



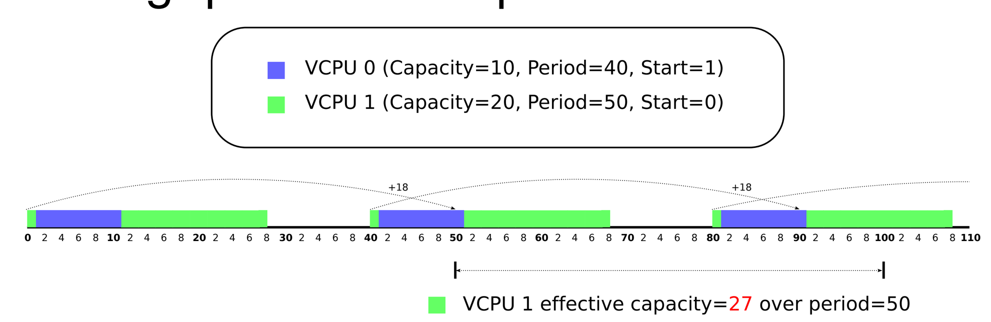
Rate-Monotonic Property Priority of VCPU V_i is inversely proportional to period T_i.

Liu-Layland bound If $\sum \frac{C_i}{T_i} \le n\left(\sqrt[n]{2}-1\right)$ then temporal isolation is guaranteed.

Example: $\frac{1}{3}+\frac{1}{4}+\frac{2}{10} \overset{?}{\le} 3\left(\sqrt[3]{2}-1\right)$

Blocking

- Programs may choose to wait for arbitrary reasons, or finish early.
- POSIX Sporadic specification is exploitable through this means. [Stanovich 10]
- A clever program can exceed its budget by causing "premature replenishment."



- Solution: Split partly-used replenishment upon blocking. [Stanovich 10]
- Splits can later be merged if they wind up back-to-back.

Overhead

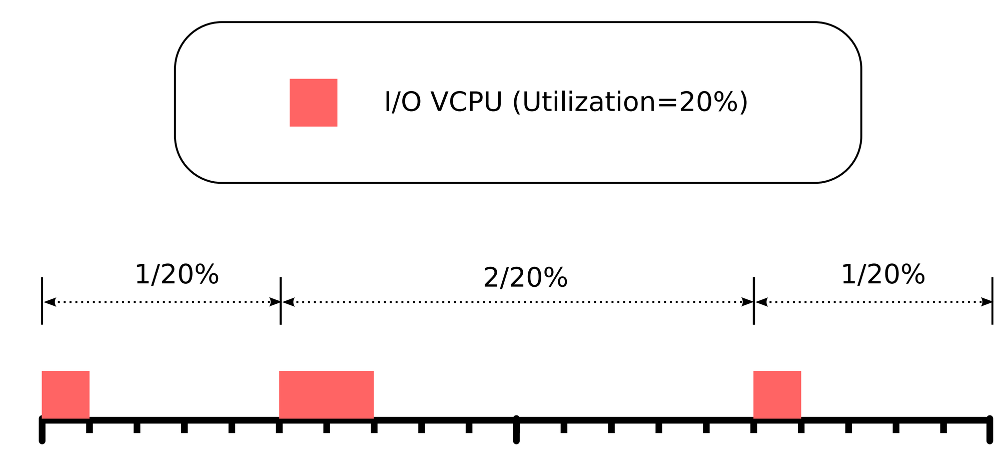
- Managing splits and merges is time and space consuming.
- For efficiency, replenishment list size is hard-capped.
- As a trade-off, reaching the cap causes the VCPU to lose effective capacity.

I/O

- Most I/O only uses short bursts of CPU.
- I/O tasks block quite often, usually while waiting for hardware.
- Sporadic Servers doing I/O hit the replenishment cap easily.
- Our alternative for I/O VCPUs: **Priority-Inheriting Bandwidth-preserving Servers** (PIBS).
- Different VCPUs can co-exist in the same scheduling framework.
- Many different mappings of I/O VCPUs and I/O devices can be designed. Examples:
  - one I/O VCPU system-wide,
  - an I/O VCPU per device, or
  - an I/O VCPU per device per Main VCPU.

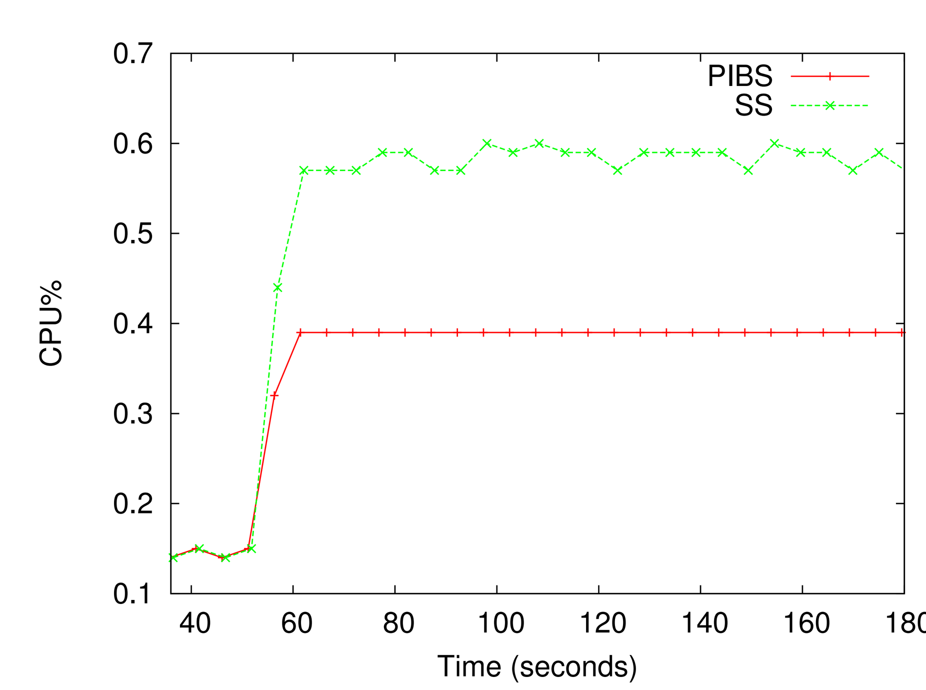
PIBS

- I/O VCPUs run on behalf of a Main VCPU.
- They inherit their current priority from that Main VCPU.
- PIB Servers have a target CPU utilization U.
- When the PIB Server consumes b time units, it must wait b/U before executing again.
- $b \le C_{max}$ where C_{max} is derived from the resources of the Main VCPU requesting the I/O work.



PIBS vs SS

- Experimental evaluation: network ping-flood begins at $t = 50$.
- Compare I/O VCPU algorithm: PIBS vs SS.

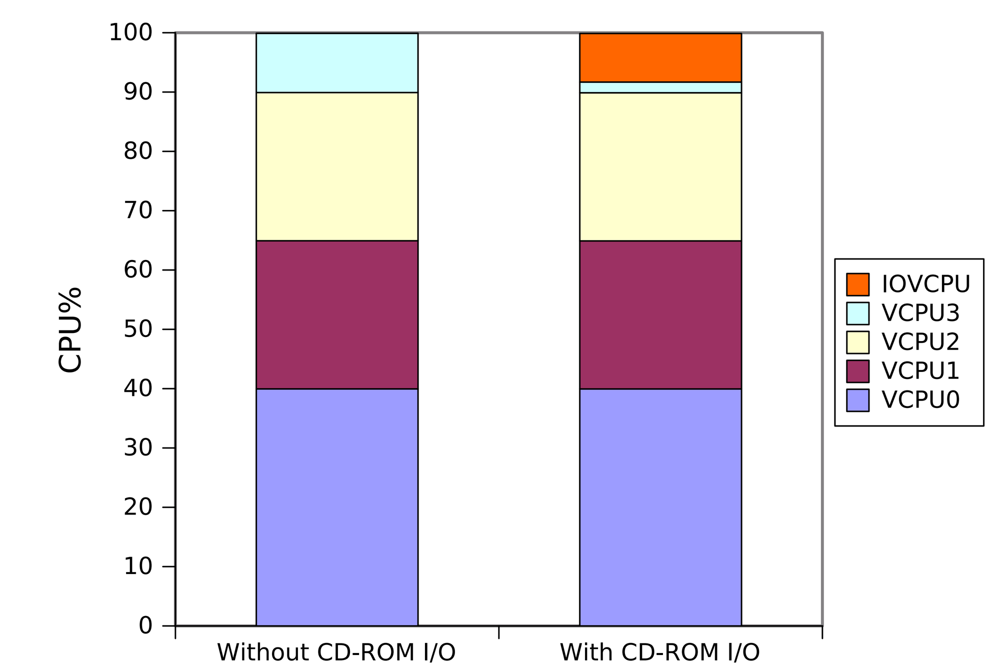


Scheduling Overhead

- PIBS is more efficient and consistent.

I/O Isolation

- In this experiment, `VCPU1` uses `IOVCPU` to read from the CD-ROM drive.
- Only the lowest priority, `VCPU3`, is forced to give up capacity.



Future Work

- Mapping VCPUs to PCPUs is the next important step.
- Cache awareness: when VCPUs get to run, try to avoid memory stalls that reduce number of instructions per clock cycle.
- Inter-process communication: the costs of IPC need to be controlled, particularly communication that spans PCPUs.
- Safety and modularity: separate components using hardware and software techniques. Isolate them with VCPUs.

Further Info

- Danish, Li and West. *Virtual CPU Scheduling in the Quest Operating System*. In Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium, April 11-14th, 2011, Chicago, IL, USA.
- Quest is free software.
- Website: `http://QuestOS.github.com/`

Advisor: Richard West and Hongwei Xi