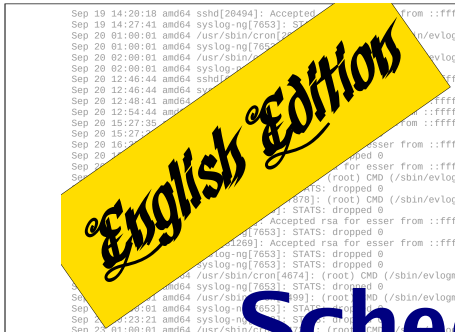


```
Sep 19 14:20:18 amd64 sshd[20494]: Accepted          from ::ffff:87.234.201.207 port 61557
Sep 19 14:27:41 amd64 syslog-ng[7653]: ST
Sep 20 01:00:01 amd64 /usr/sbin/cron[2               in/evlogmgr -c "severity=DEBUG")
Sep 20 01:00:01 amd64 syslog-ng[765
Sep 20 02:00:01 amd64 /usr/sbin/c                     evlogmgr -c 'age > "30d"')
Sep 20 02:00:01 amd64 syslog-n
Sep 20 12:46:44 amd64 sshd[                        ff:87.234.201.207 port 62004
Sep 20 12:46:44 amd64 sys
Sep 20 12:48:41 amd64                             :ffff:87.234.201.207 port 62105
Sep 20 12:54:44 amd                              m ::ffff:87.234.201.207 port 62514
Sep 20 15:27:35                                from ::ffff:87.234.201.207 port 64242
Sep 20 15:27:
Sep 20 16:                                   esser from ::ffff:87.234.201.207 port 63375
Sep 20 1                                  pped 0
Sep 2                                  a for esser from ::ffff:87.234.201.207 port 63546
Se                                    (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
                                    ATS: dropped 0
                                 7878]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
                                 ]: STATS: dropped 0
                              ]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63397
                              g[7653]: STATS: dropped 0
                              1269]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64391
                          slog-ng[7653]: STATS: dropped 0
                          syslog-ng[7653]: STATS: dropped 0
                        64 /usr/sbin/cron[4674]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
S                     amd64 syslog-ng[7653]: STATS: dropped 0
Sep                 01 amd64 /usr/sbi    499]: (root)  MD (/sbin/evlogmgr -c  age > "30d"')
Sep        0:01 amd64 syslog-n  7653]: STATS: drop d 0
Sep 2   0:23:21 amd64 syslog-ng
Sep 23 01:00:01 amd64 /usr/sbin/cr
Sep 23 01:00:01 amd64 syslog-ng[7653
Sep 23 02:00:01 amd64 /usr/sbi
Sep 23 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:05 amd64 sshd[6554]: Accepted publickey for esser from ::ffff:192.168.1.5 port 59771 ssh2
Sep 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:34 amd64 sshd[6606]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62093
Sep 24 01:00:01 amd64 /usr/sbin/cron[12436]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 02:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64456
Sep 24 11:15:48 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 13:49:08 amd64 sshd[23197]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61330
Sep 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: snd_seq_midi_event: unsupported module, tainting kernel.
Sep 24 15:42:07 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: snd_seq_oss: unsupported module, tainting kernel.
Sep 24 20:25:31 amd64 sshd[29399]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62566
Sep 24 20:25:31 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 01:00:02 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 25 02:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64183
Sep 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64253
Sep 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62029
Sep 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:05:37 amd64 sshd[11554]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62822
Sep 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:06:10 amd64 sshd[11586]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62951
Sep 25 14:07:17 amd64 sshd[11608]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63392
Sep 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63709
Sep 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62778
```

Scheduling (5)

Threads

- When handling threads: Treat execution aspect and the other process properties separately
- Application can consist of a number of threads which cooperate and simultaneously run in the same address space
- Threads (of one process) that run on different CPUs can dramatically increase process performance

Multiprocessor Thread Scheduling (1)

- Load Sharing
  - no (permanent) assignment of CPUs to processes
- Gang Scheduling
  - a number of threads (that belong together) gets an according number of CPUs by the scheduler simultaneously

Multiprocessor Thread Scheduling (2)

- Dedicated CPU allocation
  - threads are a assigned to a specific CPU
- Dynamic scheduling
  - number of threads can change during program execution

Load Sharing

- distribute compute-load uniformly across all CPUs
- no centralized scheduler needed
- uses global queue(s)

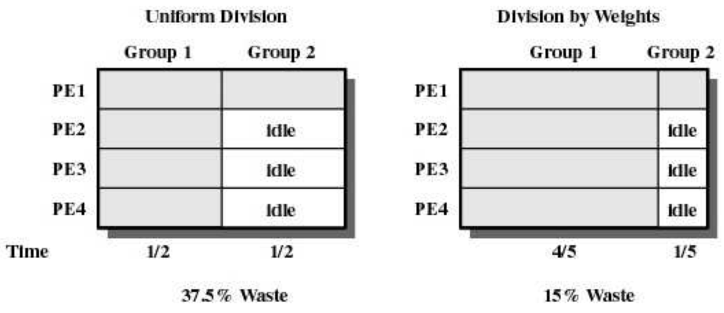
Disadvantages of Load Sharing

- central queue requires *mutual exclusion*
  - can be a bottleneck when more than one CPU searches for work at the same time
- preemptive threads are likely to change the CPU often
  - cache usage is less efficient
- when all threads are in the global queue, it is unlikely that all threads of one process are scheduled simultaneously

Gang Scheduling

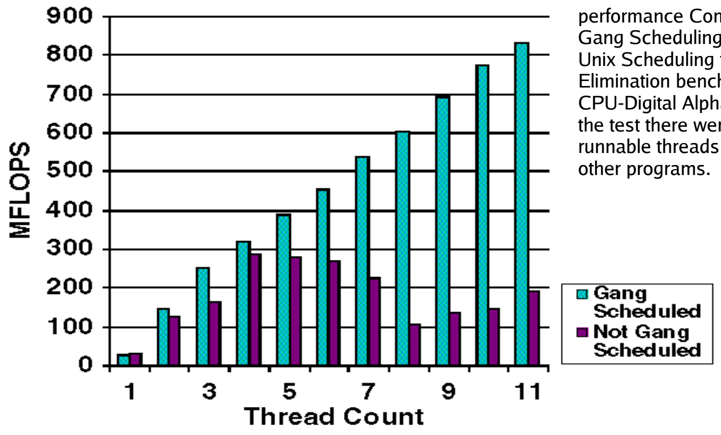
- simultaneous scheduling of Threads which belong to one process
- useful for applications in which performance suffers immensely when any part (thread) is not active
- threads must often synchronize

Scheduling Groups



Scheduling groups with 4 / 1 threads

Gang Scheduling: Performance



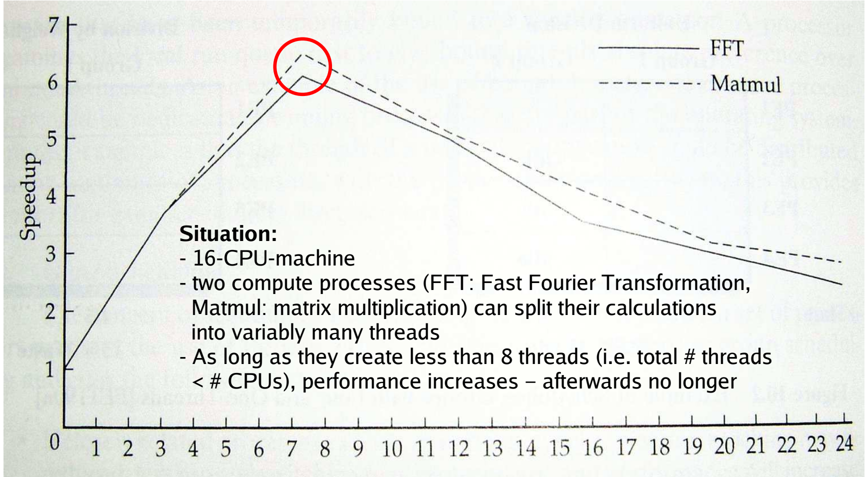
performance Comparison of Gang Scheduling and normal Unix Scheduling for a Gauss Elimination benchmark on a 12-CPU-Digital Alpha 8400. During the test there were approx. 20 runnable threads belonging to other programs.

Picture: http://www.llnl.gov/asci/pse_trilab/sc98.summary.html

Dedicated CPU Allocation (1)

- when the scheduler picks a process, it schedules each of its threads on one CPU
- some CPUs may be idle
- mo multitasking on the CPUs

Dedicated CPU Allocation (2)



Situation:
- 16-CPU-machine
- two compute processes (FFT: Fast Fourier Transformation, Matmul: matrix multiplication) can split their calculations into variably many threads
- As long as they create less than 8 threads (i.e. total # threads < # CPUs), performance increases – afterwards no longer

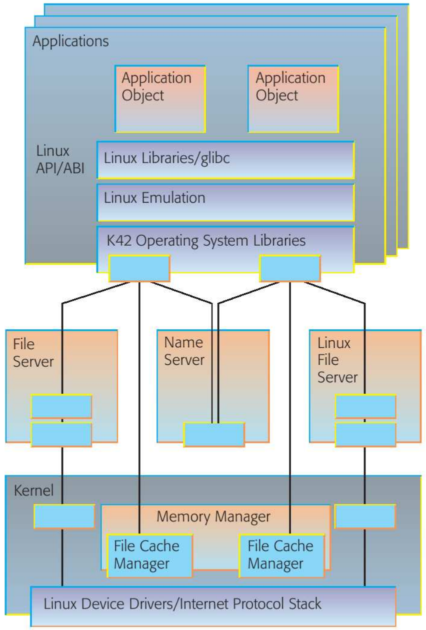
Figure 10.3 Application Speedup as a Function of Number of Processes [TUCK89]

Dynamic Scheduling

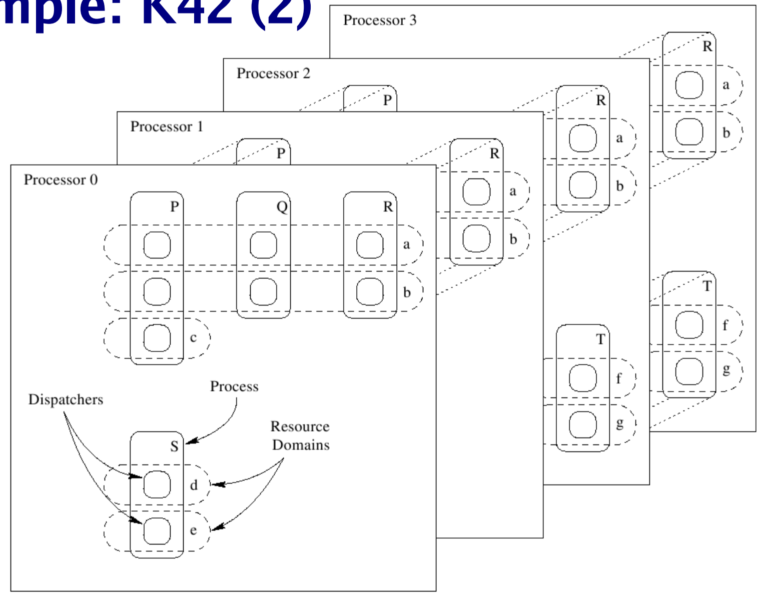
- number of threads (of one process) can change dynamically: process creates new threads and destroys them – *depending on the resources* that the operating system provides
- operating system adjusts the load in order to optimize CPU utilization
  - idle CPUs are scheduled to run threads
  - new processes can receive a CPU that is currently used by a process having several CPUs
  - ignore request until a CPU becomes available
  - new processes receive a CPU with higher priority than already running applications do

Example: K42 (1)

- Linux-compatible OS for machines with hundreds of CPUs
- object-oriented; OS system calls are IPC calls
- two scheduler levels
- thread scheduling: completely handled in user mode
- kernel scheduler
  - manages so called dispatchers which in turn manage the threads (often: one dispatcher per process)
  - runs independently on each CPU


Applications
Application Object
Application Object
Linux API/ABI
Linux Libraries/glibc
Linux Emulation
K42 Operating System Libraries
File Server
Name Server
Linux File Server
Kernel
Memory Manager
File Cache Manager
File Cache Manager
Linux Device Drivers/Internet Protocol Stack

Example: K42 (2)



Example: K42 (3)

- Thread Migration:
  K42 thread scheduler does Load Balancing, by migrating threads from busy dispatchers to idle ones
- kernel may, as an exception, also migrate dispatchers to a different CPU
- kernel scheduler activates resource domains (inside one domain: dispatchers in a circle)
- does Gang Scheduling (see processes P,R)
- dispatcher can cope with single threads blocking (e.g. for page faults or I/O), without also blocking

Example: K42 (4)

- processes have a choice:
  - real multitasking -> use several dispatchers
  - only programming comfort of threads -> one dispatcher is enough
- miscellaneous thread libraries for programmers, including POSIX threads
- K42 scheduler: http://www.research.ibm.com/K42/white-papers/Scheduling.pdf (2002)
- introduction to K42: http://www.research.ibm.com/journal/sj/442/appavoo.pdf

```
Sep 19 14:2[illegible] amd64 sshd[[illegible]]: Accepted rsa for esser from ::ffff:87.234.201.207 port [illegible]
Sep 19 14:27:41 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 01:00:01 amd64 /usr/sbin/cron[29278]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 20 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 02:00:01 amd64 /usr/sbin/cron[30103]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 20 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:46:44 amd64 sshd[6516]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62004
Sep 20 12:46:44 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:48:41 amd64 sshd[6609]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62105
Sep 20 12:54:44 amd64 sshd[6694]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62514
Sep 20 15:27:35 amd64 sshd[9077]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64242
Sep 20 15:27:35 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:37:11 amd64 sshd[10102]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63375
Sep 20 16:37:11 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:38:10 amd64 sshd[10140]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63546
Sep 21 01:00:01 amd64 /usr/sbin/cron[17055]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 21 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 02:00:01 amd64 /usr/sbin/cron[17878]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 21 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:43:26 amd64 sshd[31088]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63397
Sep 21 17:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:53:39 amd64 sshd[31269]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64391
Sep 21 18:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 19:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 01:00:01 amd64 /usr/sbin/cron[4674]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 22 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[5499]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 20:23:21 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 01:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 23 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 02:00:01 amd64 /usr/sbin/cron[25555]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 23 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:05 amd64 sshd[6554]: Accepted publickey for esser from ::ffff:192.168.[illegible] port 59[illegible] ssh2
Sep 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:34 amd64 sshd[6606]: Accepted rsa for esser from ::ffff:87.[illegible] port [illegible]093
Sep 24 01:00:01 amd64 /usr/sbin/cron[12436]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 02:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64456
Sep 24 11:15:48 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 13:49:08 amd64 sshd[23197]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61330
Sep 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: snd_seq_midi_event: unsupported module, tainting kernel.
Sep 24 15:42:07 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: snd_seq_oss: unsupported module, tainting kernel.
Sep 24 20:25:31 amd64 sshd[29399]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62566
Sep 24 20:25:31 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 01:00:02 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 25 02:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64183
Sep 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64253
Sep 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62029
Sep 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:05:37 amd64 sshd[11554]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62822
Sep 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:06:10 amd64 sshd[11586]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62951
Sep 25 14:07:17 amd64 sshd[11608]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63392
Sep 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63709
Sep 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62778
```

Linux O(1) Scheduler

Linux O(1) Scheduler (1)

- change with Linux kernel 2.6:
  new scheduler which remedies some problems of the old 2.4.x scheduler:
  - schedule time was (linearly) dependent on the number of processes, i.e. O(n)
    -> poor performance with very many processes
  - poor performance on SMP machines

Linux O(1) Scheduler (2)

Causes (in kernel 2.4)

- one common queue for all processes on all CPUs;
  no sorting in that queue
- scheduler must search whole queue in order to find the next process to schedule
- one single lock for the runqueue
  -> one CPU accessing the queue blocks access for all further CPUs
- result: schedule action very complex

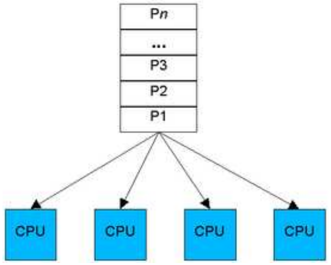


Bild: Linux Journal,
http://www.linuxjournal.com/node/7178/

Linux O(1) Scheduler (3)

Kernel 2.4

- processes not bound to a CPU, assignments random (no processor affinity)
  -> processes change CPUs regularly
  -> poor utilization of CPU caches

Linux O(1) Scheduler (4)

Kernel 2.6: new O(1) Scheduler with the following features:

- O(1) scheduler: time required for selecting the next process (for one CPU) is constant – independent on the number of processes
- CPUs don't block one another in case of simultaneous scheduling decisions
- load balancer distributes compute-load uniformly across several CPUs