Cloud Computing

Winter Term 2019/2020

Tutorial Session 2

Anton Gulenko
Complex and Distributed IT-Systems

anton.gulenko@tu-berlin.de

Assignment 1

- Results will be available on ISIS within the next weeks
- Typical mistakes:
  - Benchmark script should output to stdout, not to a file:

```
echo "$t_time,$cpu,$mem,$diskRand,$diskSeq" >> sysfile.csv
```

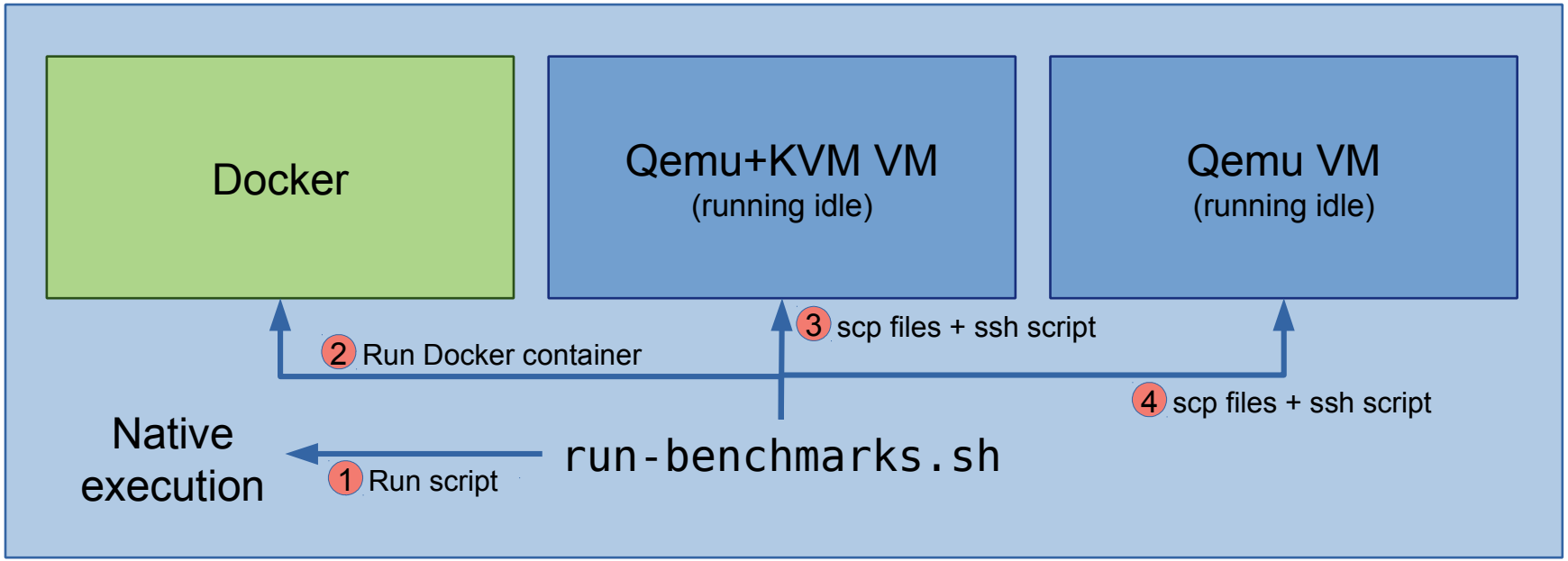
  - Single-line script is not readable:

```
echo  $(date +%s)","$(sysbench cpu run --time=60 | grep "events per second:"
 | awk "{ print $ 4 }")","$(sysbench memory run --memory-block-size=4K --mem
ory-total-size=100T --time=60 | grep "transferred" | awk "{ print $ 4 }" | c
ut -b 2-)$(sysbench fileio prepare --file-num=1 --file-total-size=1G --file-
extra-flags=direct >/dev/null)","$(sysbench fileio run --file-test-mode=rndr
d --file-total-size=1G  --file-extra-flags=direct --file-num=1 --time=60 | g
rep "read, MiB/s:" | awk "{ print $ 3}")","$(sysbench fileio run --file-test
-mode=seqrd --file-total-size=1G  --file-extra-flags=direct --file-num=1 --t
ime=60 | grep "read, MiB/s:" | awk "{ print $ 3}") $(sysbench fileio cleanup
 >/dev/null) | tee -a /home/results.csv
```

Practical Assignment 2

- Due: 19.12.2019
- Summary:
  - Work on 1 host machine
    - Preferably your laptop (physical machine). If you don’t have Linux, use a VM.
  - Prepare virtualization environments:
    - Qemu/KVM
    - Docker
  - Write 2 new benchmarks:
    - Forksum
    - Iperf3 (uplink speed)
  - Execute benchmarks on different virtualization platforms
- Public cloud platforms are not mandatory for this assignment, but do not yet delete your accounts (shut down your VMs if not used for the assignment)

Benchmark Setup



Experiment host: Laptop or Public Cloud VM

Virtualization Platforms

- Native execution
  - Simply execute `./benchmark.sh`
- Docker
  - Write Dockerfile that executes benchmark script
  - The container image must contain all tools and files for all benchmarks, execute the benchmark when started without parameters, and exit after printing the results
- Qemu (with and without KVM)
  - Use an Ubuntu 18.04 cloud image
  - Either work directly with qemu-system-* executable, or use a management program such as Libvirt

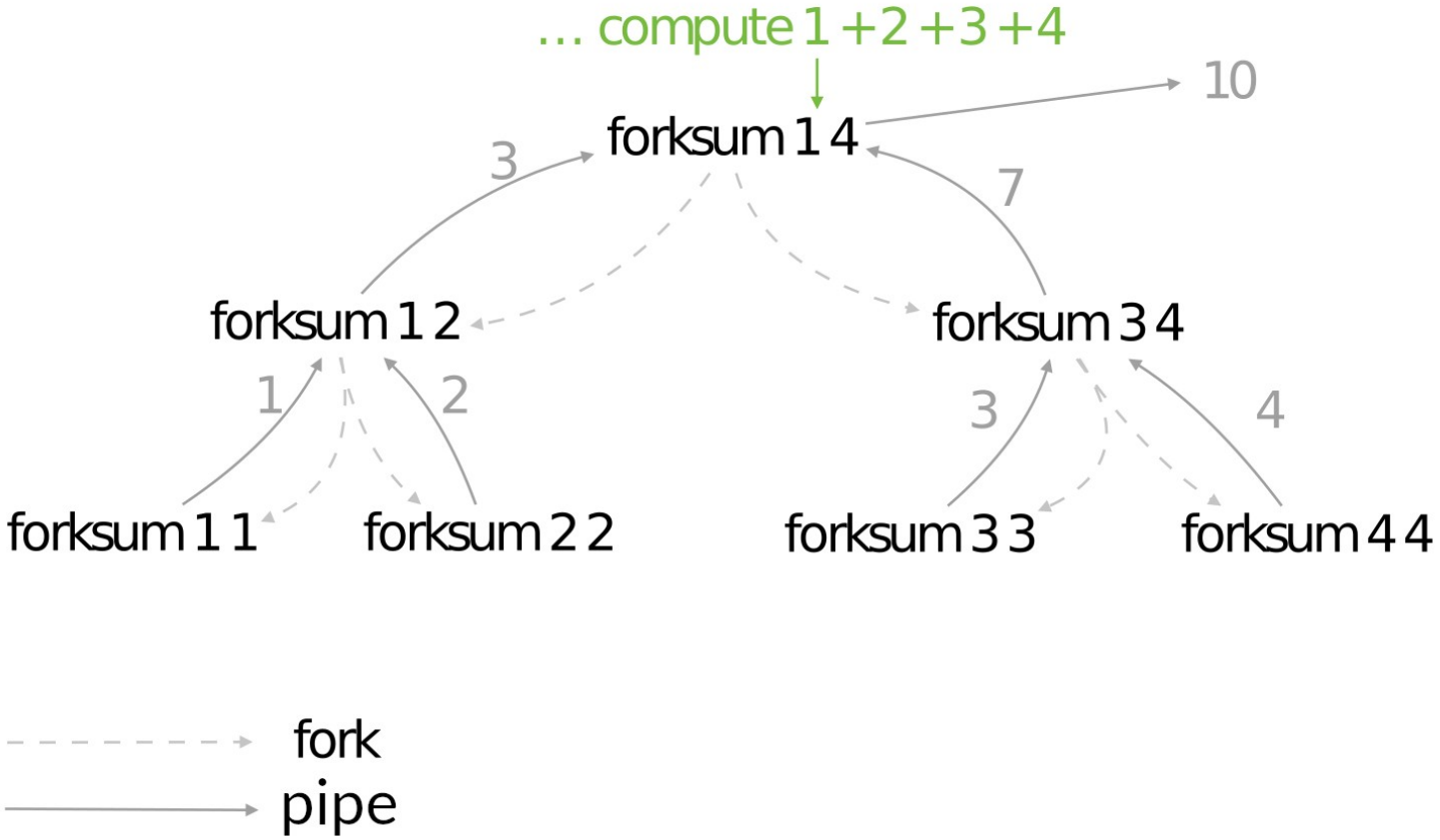
Benchmarks

- Basic resources: CPU, RAM and disk access
  - Reuse from assignment 1
  - Option: use the benchmark.sh script provided on ISIS
- New benchmark: forksum
  - Simulates creation of many parallel processes
  - Main benchmark target: system calls
- New benchmark: Iperf3
  - Measure uplink speed to a server
  - Main benchmark target: network performance

Forksum

- Program receives 2 parameters: start and end of integer range
  - Task: compute sum of all integers within the range
  - Example: **`./forksum 100 1000`** should print **`495550`**
- Every sum is executed by a separate child process
  - If start == end: output value (end of the recursion)
  - Else: spawn 2 child processes: one for lower sub-range and one for upper sub-range
    - After child-processes return their results, parse them and output the sum

Forksum: Example



Forksum: required C system calls

- **`fork()`**: continue program as two separate processes
  - Return value of `fork()` tells the program if it is the child or parent process
- **`pipe()`**: Create a bidirectional pipe that can be used to write in the child process, and to read in the parent process
- **`fdopen()`**: Open file descriptor as a stream for reading and writing
- **`fprintf()`**: Write formatted text to a stream, can be used to write to stderr
- **`getline()`**: Read a line of text from a stream
- Other useful functions: wait(), perror(), read(), strotl(), printf(), close(), exit()

Iperf3 Uplink Benchmark

- Iperf3 measures network performance
- Requires a server to establish a connection
  - You can use the Iperf3 server hosted by us (IP in assignment sheet)
  - If it does not work, use a public Iperf3 server, e.g. speedtest.wtnet.de