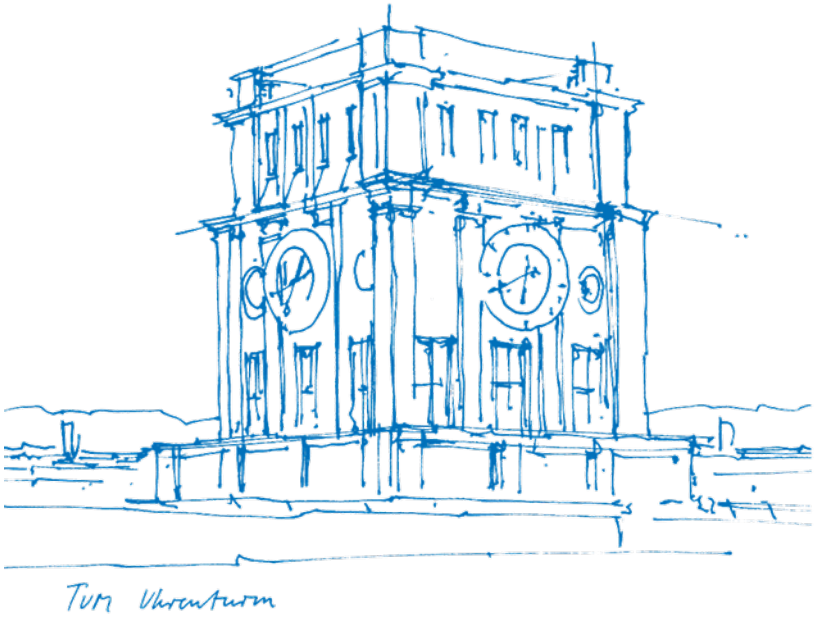
Parallel Programming Tutorial - Sequential Programming

Vincent Bode, M.Sc.
Chair for Computer Architecture and Parallel Systems (CAPS)
Technichal University Munich
April 29, 2020



Organization

- Weekly lectures
  - Time slot: Monday 10:15-11:45
  - With integrated quizzes

Organization

- Weekly lectures
  - Time slot: Monday 10:15-11:45
  - With integrated quizzes
- Weekly exercises
  - Time slot: Wednesday 08:30-10:00
  - Schedule subject to change
  - In class programming exercises in teams
  - Homework to be submitted in teams

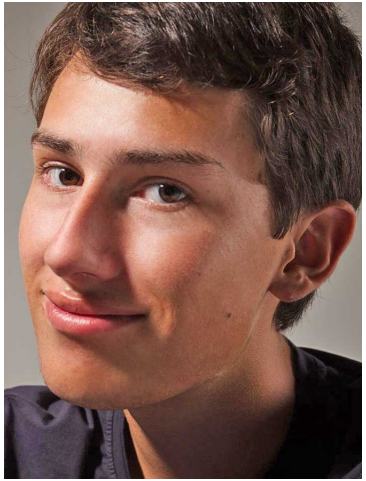
Organization

- Weekly lectures
  - Time slot: Monday 10:15-11:45
  - With integrated quizzes
- Weekly exercises
  - Time slot: Wednesday 08:30-10:00
  - Schedule subject to change
  - In class programming exercises in teams
  - Homework to be submitted in teams
- Contact us anytime:
  - Moodle forum: https://www.moodle.tum.de/mod/forum/view.php?id=1110865
  - E-Mail: parprog@lists.lrz.de
  - Livestream chat during broadcasts
  - TUM RocketChat: https://chat.tum.de/channel/parprog
  - Room: MI, 01.04.035

The Team

Iva Villa
Tutor

Prof. Martin Schulz
Lecturer



Vincent Bode
Central Tutorial

Aldo Kacorri
Tutor

Assignments

- In class exercises
  - Entirely optional
  - In teams
  - Relevant to homework
  - Can also be solved after session

Assignments

- In class exercises
  - Entirely optional
  - In teams
  - Relevant to homework
  - Can also be solved after session
- Homework
  - Approximately 1/week
  - Usually 1 week of time to solve
  - Success on 80% $\rightarrow$ 0.3 bonus
    - Only applies for a passed exam (original grade $\leq$ 4.0)
  - Online submission: https://parprog.caps.in.tum.de
  - We check: correctness (output, threads, synchronization), speedup, memory leaks, plagiarism
  - Solutions discussed in next exercise session

Assignments

- In class exercises
  - Entirely optional
  - In teams
  - Relevant to homework
  - Can also be solved after session
- Homework
  - Approximately 1/week
  - Usually 1 week of time to solve
  - Success on 80% $\rightarrow$ 0.3 bonus
    - Only applies for a passed exam (original grade $\leq$ 4.0)
  - Online submission: https://parprog.caps.in.tum.de
  - We check: correctness (output, threads, synchronization), speedup, memory leaks, plagiarism
  - Solutions discussed in next exercise session
- Q&A Sessions
  - Live interaction session with our tutors.
  - Meant for issues with homework.

Moodle Exercise

Moodle Quiz

Using the submission tool

https://parprog.caps.in.tum.de

Teambuilding

Go find a team!

Up to 3 students per team.

Sequential Programming

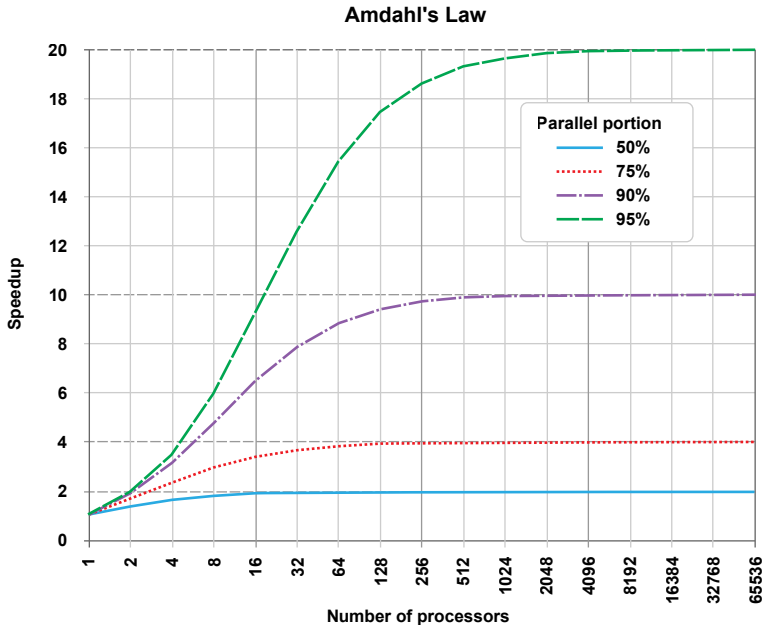


Figure 1: Life is sad sometimes (Source).

Assignment 1: “Varys’ Very Advanced Encryption Standard (VV-AES)”

Assignment: VV-AES

Figure 2: A venerable spider[1].

[1] From https://en.wikipedia.org/wiki/File:Varys-Conleth_Hill.jpg

[2] From https://www.cbr.com/max-von-sydow-joins-game-of-thrones-as-three-eyed-raven/

[4] From: https://gameofthrones.fandom.com/wiki/Pycelle

Assignment: VV-AES

Figure 2: A venerable spider[1].

Figure 3: Just your average westerosi postman[2].

[1] From https://en.wikipedia.org/wiki/File:Varys-Conleth_Hill.jpg

[2] From https://www.cbr.com/max-von-sydow-joins-game-of-thrones-as-three-eyed-raven/

[4] From: https://gameofthrones.fandom.com/wiki/Pycelle

Assignment: VV-AES

Figure 2: A venerable spider[1].

Figure 3: Just your average westerosi postman[2].

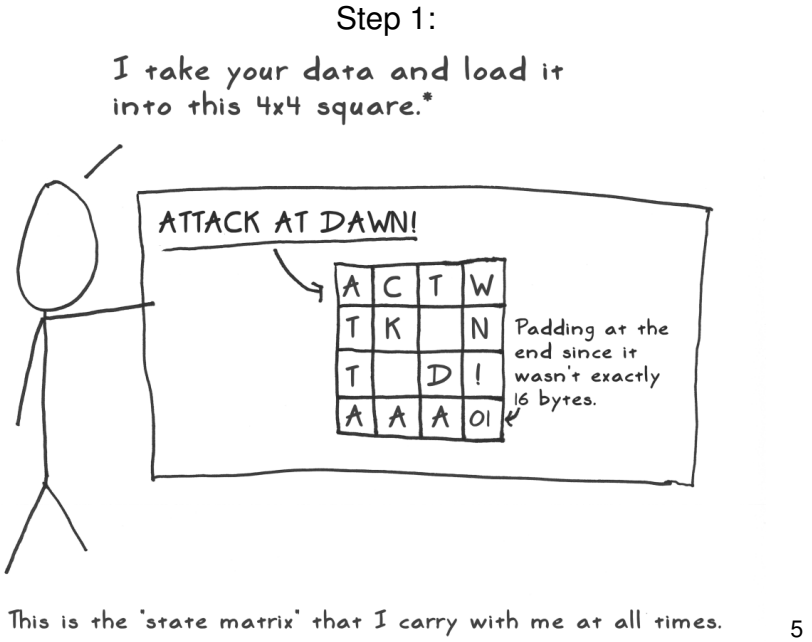
Figure 4: One of the brightest minds this side of the narrow sea[4].

[1] From https://en.wikipedia.org/wiki/File:Varys-Conleth_Hill.jpg

[2] From https://www.cbr.com/max-von-sydow-joins-game-of-thrones-as-three-eyed-raven/

[4] From: https://gameofthrones.fandom.com/wiki/Pycelle

Explanation: VV-AES

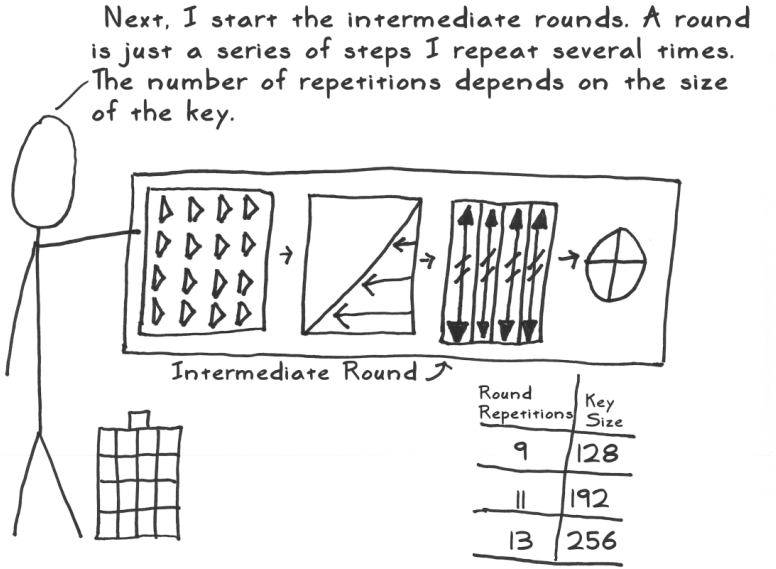


5

[5] Courtesy of Jeff Moser's A Stick Figure Guide to the Advanced Encryption Standard (AES)

Explanation: VV-AES

Step 2:

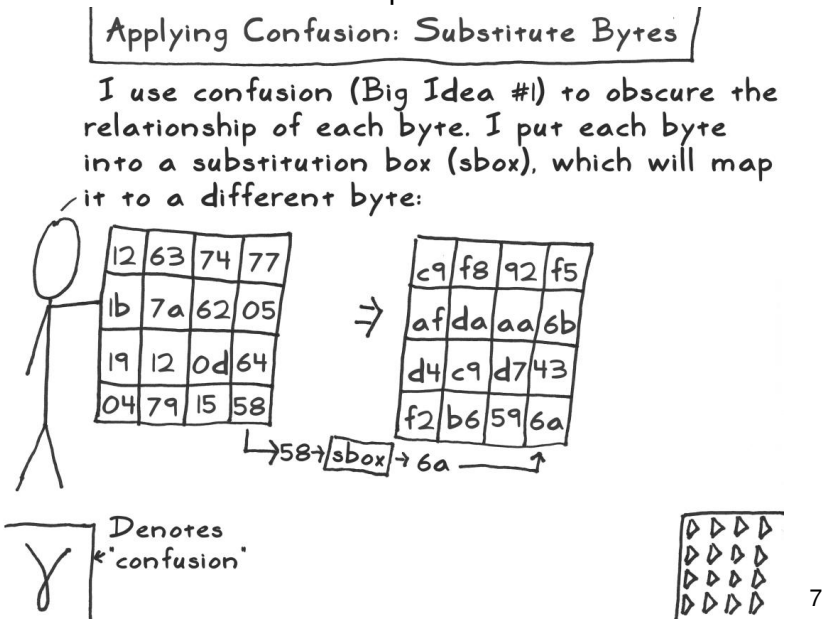


[6]

[6] Courtesy of Jeff Moser's A Stick Figure Guide to the Advanced Encryption Standard (AES)

Explanation: VV-AES

Step 2.1:

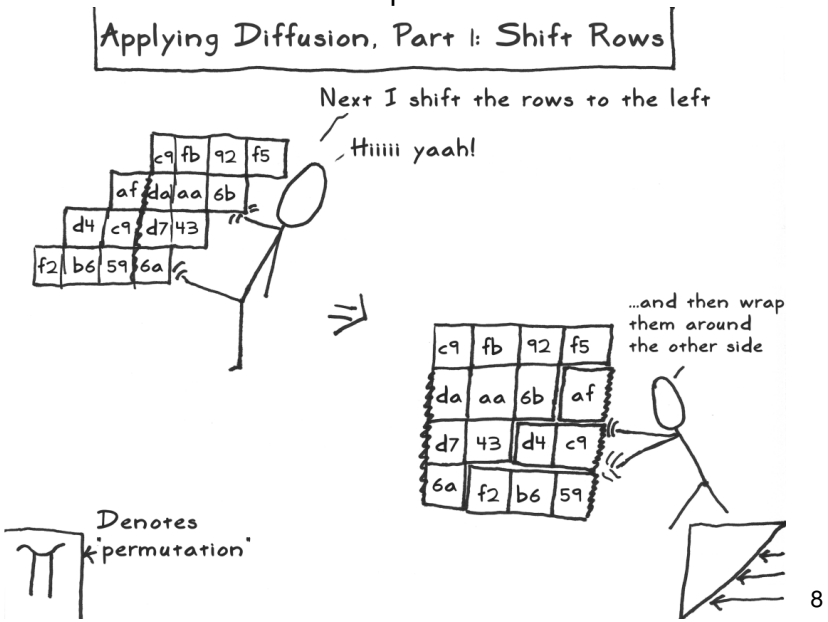


[7]

[7] Courtesy of Jeff Moser's A Stick Figure Guide to the Advanced Encryption Standard (AES)

Explanation: VV-AES

Step 2.2:

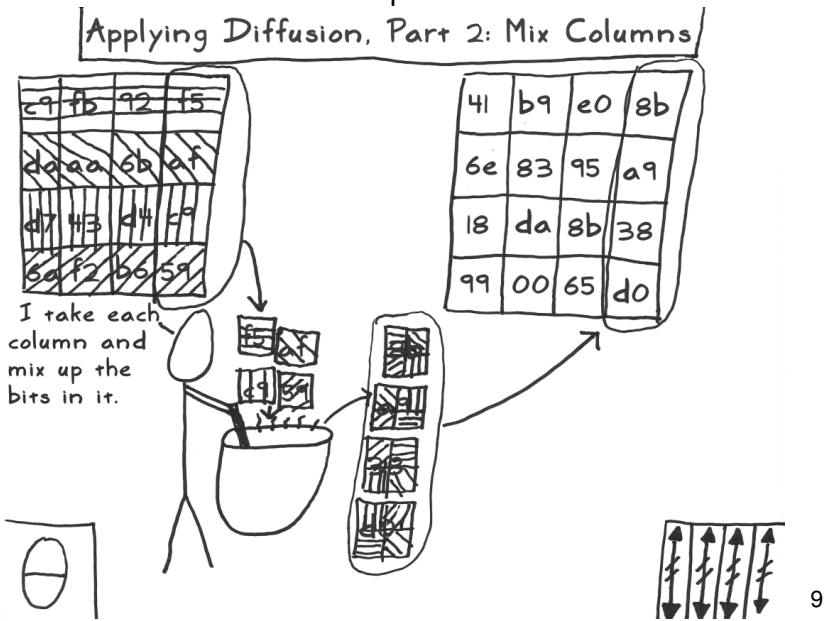


[8]

[8]Courtesy of Jeff Moser's A Stick Figure Guide to the Advanced Encryption Standard (AES)

Explanation: VV-AES

Step 2.3:

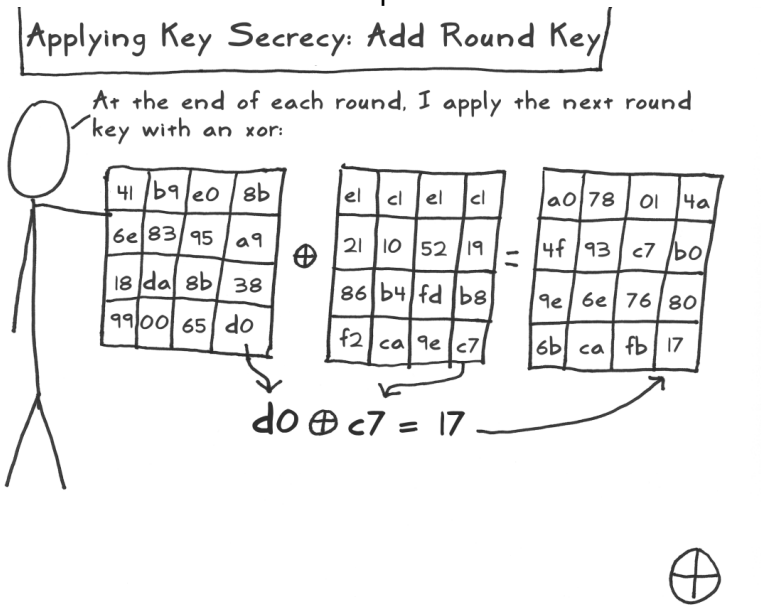


[9]

[9] Courtesy of Jeff Moser's A Stick Figure Guide to the Advanced Encryption Standard (AES)

Explanation: VV-AES

Step 2.4:



10

[10] Courtesy of Jeff Moser's A Stick Figure Guide to the Advanced Encryption Standard (AES)

Assignment: VV-AES

Assignment: VV-AES

Figure 5: A most generous Master of Coin.[11]

Assignment: VV-AES

Figure 5: A most generous Master of Coin.[11]

Constraints

- Only a single Personal Street Urchin (PSU).
- Locked in a container.
- Is eventually killed when he takes too long to do his job.
- Reads plain text messages from a file called *stdin*.
- Writes encrypted messages to a file called *stdout*.
- Can file complaints as they like to *stderr*.
- Can only remember $\sim 10^9$ pieces of information at a time without writing them down.
- Lacks access to the network of whispers.
- Any paper trail is burned after every batch of encrypting.

Assignment: VV-AES

Your job: Optimize the Personal Street Urchin's Workflow

- Speedup of 25x in relation to the provided implementation.
- Note: This exercise does not count towards the grade bonus (0 points).
- Note: VV-AES is a slightly simplified version of AES.
  - More information on real AES http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html.

Hints

Typical performance problems:

- Frequent memory allocation/deallocation
- Unnecessary copying or conversion of data
- Lack of cache awareness
- **Bad algorithms**
- **Doing it yourself**

Building and Running the Program

Developing

- Copy “sequential_implementation.cpp” to “student_submission.cpp”
- Write your code in “student_submission.cpp”

Build the program

- Makefile:
  ```
  $> make all
  ```

Usage of the program

- Sequential:
  ```
  $> ./sequential_implementation
  ```
- Your solution:
  ```
  $> ./student_submission
  ```

No Linux? No Problem.

You have several options for getting access to a linux environment.

- Install linux in a virtual machine (e.g. VirtualBox)
  - Don’t forget to assign multiple cores to the virtual machine
- Use Rechnerhalle
  - Accessible at the workstations or remotely.
  - Remote ssh access: ssh <rbg-id>@lxhalle.in.tum.de
  - Your RBG-Id is the part in front of your @in.tum.de or @ma.tum.de email address.
- Ask the tutors; they will be more than happy to help you.

Questions

?

Please like, subscribe, and fill out my feedback form.