CSaP - Project AA 2018-2019

- Goal: “Grid FTP” client-server application
- The “server side” will be made of different processes/subsystems, possibly running on differen hosts:

  A “Metadata Server” (MDS)

  One or more “Data Repositories” (DR)

CSaP – MDS Server

- Handles initial connections with clients, maintains a map where files have been stored, allocated blocks, free lists, etc.
- Upon startup it contacts all known DR (using a configuration file) to get updated allocation maps
- Waits for requests from clients

  Auth <Userid> <Password>, returns (Key, NOK)

  GetDR, returns (<#,IP> pairs, NOK)

  Put <File> <Size>, returns (<#DR,BlockRange> pairs, NOK)

  Get <File>, returns (<#DR,BlockRange> pairs, NOK)

  Remove <File>, returns (OK,NOK)

CSaP - DR

- Store data blocks on local file(s), keeping an hash and a key for each block of data
- Upon startup rebuild block maps, collaborating with MDS
- Receive requests from clients:

  Get <Block#> key, returns (Block,NOK)

  Put <Block#> key, returns (OK,NOK)
- Receive requests from MDS

  Remove <block#>, returns (OK,NOK)

  Transfer <block#> <clientIP> <key>, returns (OK,NOK)
- Aknowledge completed/interrrupted/failed operations to MDS/Clients

CSaP - Clients

- Connect to MDS to authenticate ad get the map of servers/blocks for the file to store/retrieve/delete
- For a store/retrieve operation, contact each DR as indicated in the list, possibly with multiple parallel processes performing the operation

Evaluation Scorecard

- The code works (reliably)
- Race conditions
- Robustness under unexpected situations
    - Misbehaved clients
    - Communication errors
    - Reboots/crashes of MDS and DR
    - Disk errors

Project Collaterals

- Source Code, including for each function/global variable/data structure:
  - Purpose
  - Parameters, side effects and return value (for functions)

- **DOCUMENTATION** describing:
  - Design Choices (and the reasoning behind them)
  - Macro modules and their interaction
  - Test cases
  - Release notes, including limitations, known errors, etc

Suggestions

- Be creative: evaluate possible alternatives
- Apply the KISS principle (keep it simple .. :^)
- Start prototyping early
- Few days before deadline, send a draft of code and documentation for review/interaction