Database Systems on Modern CPU Architectures

Moritz Sichert, André Kohn
Technical University Munich
Chair for Database Systems

sichert@in.tum.de
kohna@in.tum.de
Organisation

- Lecture + Tutorial: Tuesdays from 14:00 – 17:00. (Next week: C++ crash course)
- Programming assignments every 2 weeks starting now. (probably 7)
- Announcements on the website & Mattermost.
- Assignments managed via GitLab (CI). (Due Tuesdays at 14:00)
- No teams.
- Bonus system:
  - 0.3 / 0.4 grade bonus on final exam. (>= 75% of all points)
  - +4 points for correct implementation of assignments.
  - +1 points for code quality per assignment. (>= 1 point for implementation)
Chair for Database Systems

Welcome to our GitLab

You can register an account using an email address from the following domains:

- in.tum.de
- tum.de
- mytum.de

We reserve the right to disable and eventually delete accounts after a semester ended.

For unrelated projects, please use the GitLab from the LRZ: https://gitlab.lrz.de/

Register

Full name
Insert Name Here

Username
insert-username-here
Username is available.

Email
name@in.tum.de

Email confirmation
name@in.tum.de

Password
******
Minimum length is 8 characters

Register

Didn't receive a confirmation email? Request a new one.
Welcome to GitLab
Code, test, and deploy together

Create a project
Projects are where you store your code, access issues, wiki and other features of GitLab.

Create a group
Groups are the best way to manage projects and members.

Explore public projects
There are 63 public projects on this server. Public projects are an easy way to allow everyone to have read-only access.

Learn more about GitLab
Take a look at the documentation to discover all of GitLab's capabilities.
A group is a collection of several projects.

If you organize your projects under a group, it works like a folder.

You can manage your group member’s permissions and access to each project in the group.
ModernDBs 2018

Your request for access has been queued for review.

ModernDBs 2018

Withdraw Access Request  Global

A group is a collection of several projects.
If you organize your projects under a group, it works like a folder.
You can manage your group member's permissions and access to each project in the group.
Fork project
A fork is a copy of a project. Forking a repository allows you to make changes without affecting the original project.

Click to fork the project
Insert Name Here

https://gitlab.db.in.tum.de/modern dbs-18/task-external-sort/forks?namespace_key=222
Forking in progress

Please wait while we import the repository for you. Refresh at will.
### External Sort

Implement the external sort algorithm. It should take an input file that contains unsigned 64 bit integers, the number of values the input file contains, an output file, and a memory size value. In the end, the output file should contain all values from the input file in ascending order.
# Task External Sort

```
# MODIFIEDS

stages:
- build
- test
- cleanup

make:
  stage: build
  script:
  - mkdir -p build
  - cd build
  - cmake -DMAKE_BUILD_TYPE=Debug ..
  - make
  - cache
  key: "CI_PIPELINE_ID"
  paths:
  - build/
  - policy: pull-push
tags:
- "clang-5.0"
- "cmake"
```

Initial commit
Moritz Sichert authored about 17 hours ago

This GitLab CI configuration is valid. Learn more
External Sort

Implement the external sort algorithm. It should take an input file that contains unsigned 64 bit integers, the number of values the input file contains, an output file, and a memory size value. In the end the output file should contain all values from the input file in ascending order.
External Sort

Implement the external sort algorithm. It should take an input file that contains unsigned 64 bit integers, the number of values the input file contains, an output file, and a memory size value. In the end the output file should contain all values from the input file in ascending order. Your implementation must not use more bytes of heap memory than the number given as memory size. You should use a k-way-merge that merges k individual runs which each fit into memory. To sort runs that fit into memory, you can use std::sort(). For the k-way-merge it may be useful to use std::priority_queue or the heap functions std::make_heap(), std::push_heap(), and std::pop_heap().

Add your implementation to the file src/external_sort.cc. The signature for the external_sort() function is given, do not change it. It may be useful to look into include/modernbs/file.h to understand the file API.

You can test your implementation by using the external_sort tool (see external_sort --help for more information) or the tests in test/external_sort_test.cc. To run them, compile the project and then execute the tester binary.

Your implementation should pass all of the tests starting with ExternalSortTest. Tests starting with AdvancedExternalSortTest are optional and are not required to pass to get full points for this task. We will of course check your code for correctness. Passing the normal tests is a requirement but does not automatically mean that you will get full points.

Additionally, your code will be checked for code quality. You can run the checks for that by using the lint CMake target (e.g. by running make lint). When your code does not pass those checks, we may deduct a small amount of points.
# External Sort

5. Implement the external sort algorithm. It should take an input file that contains unsorted 64-bit integers, the number of values the input file contains, an output file, and a memory size value. In the end, the output file should contain all values from the input file in ascending order.

8. Your implementation must not use more bytes of heap memory than the number given as memory size. You should use a k-way-merge that merges k individual runs which each fit into memory. To sort runs that fit into memory, you can use `std::sort()`. For the k-way-merge it may be useful to use `std::priority_queue` or the heap functions `std::make_heap()`, `std::push_heap()`, and `std::pop_heap()`. 

16. Add your implementation to the file `src/external_sort.cc`. The signature for the `external_sort()` function is given, do not change it. It may be useful to look into `/include/mergesort/file.h` to understand the file API.

20. You can test your implementation by using the `external_sort` tool (see `external_sort --help` for more information) or the tests in `test/external_sort_test.cc`. To run them, compile the project and then execute the `test` binary.

25. Your implementation should pass all of the tests starting with `ExternalSortTest`. Tests starting with `AdvancedExternalSortTest` are optional and are not required to pass to get full points for this task. It will of course check your code for correctness. Passing the normal tests is a requirement but does not automatically mean that you will get full points.

30. Additionally, your code will be checked for code quality. You can run the...
should contain all values from the input file in ascending order. Your implementation must not use more bytes of heap memory than the number given as memory size. You should use a k-way-merge that merges k individual runs which each fit into memory. To sort runs that fit into memory, you can use `std::sort()`. For the k-way-merge it may be useful to use `std::priority_queue` or the heap functions `std::make_heap()`, `std::push_heap()`, and `std::pop_heap()`. Add your implementation to the file `src/external_sort.cc`. The signature for the `external_sort()` function is given, do not change it. It may be useful to look into `include/moderndb/file.h` to understand the file API.

You can test your implementation by using the `external_sort` tool (see `external_sort --help` for more information) or the tests in `test/external_sort_test.cc`. To run them, compile the project and then execute the `test` binary.

Your implementation should pass all of the tests starting with `ExternalSortTest`. Tests starting with `AdvancedExternalSortTest` are optional and are not required to pass to get full points for this task. We will of course check your code for correctness. Passing the normal tests is a requirement but does not automatically mean that you will get full points. Additionally, your code will be checked for code quality. You can run the checks for that by using the `lint` (_make target (e.g. by running `make lint`). When your code does not pass those checks, we may deduct a small amount of points. Commit message

Update README.md

Target Branch

master

Commit changes

Cancel
External Sort

Implement the external sort algorithm. It should take an input file that contains unsigned 64 bit integers, the number of values the input file contains, an output file, and a memory size value. In the end the output file should contain all values from the input file in ascending order. Your implementation must not use more bytes of heap memory than the number given as memory size. You should use a k-way-merge that merges k individual runs which each fit into memory. To sort runs that fit into memory, you can use std::sort(). For the k-way-merge it may be useful to use std::priority_queue or the heap functions std::make_heap(), std::push_heap(), and std::pop_heap().

Add your implementation to the file src/external_sort.cc. The signature for the external_sort() function is given, do not change it. It may be useful to look into include/modernbs/file.h to understand the file API.

You can test your implementation by using the external_sort tool (see external_sort --help for more information) or the tests in test/external_sort_test.cc. To run them, compile the project and then execute the tester binary.

Your implementation should pass all of the tests starting with ExternalSortTest. Tests starting with AdvancedExternalSortTest are optional and are not required to pass to get full points for this task. We will of course check your code for correctness. Passing the normal tests is a requirement but does not automatically mean that you will get full points.

Additionally, your code will be checked for code quality. You can run the checks for that by using the list command with --check Narrow list. When your
Job #23094 triggered less than a minute ago by Insert Name Here.
### Team Statistics for imlab17

Only data for the chosen team is calculated. Excludes posts made in direct message channels, which are not tied to a team.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Users</td>
<td>18</td>
<td>Public Channels</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Private Channels</td>
<td>1</td>
<td>Total Posts</td>
<td>1076</td>
<td></td>
</tr>
</tbody>
</table>
GitLab Mattermost

All team communication in one place, searchable and accessible anywhere
You need to sign in or sign up before continuing.

Chair for Database Systems

Welcome to our GitLab

You can register an account using an email address from the following domains:

- in.tum.de
- tum.de
- mytum.de

We reserve the right to disable and eventually delete accounts after a semester ended.

For unrelated projects, please use the GitLab from the LRZ:
https://gitlab.lrz.de/

Sign in | Register

Username or email

Password

Remember me

Forgot your password?

Sign in

Didn't receive a confirmation email? Request a new one.
GitLab Mattermost

All team communication in one place, searchable and accessible anywhere

Teams you can join:

ModernDBs 2018

Create a new team
Welcome to:
Mattermost

Your team communication all in one place, instantly searchable and available anywhere.

Keep your team connected to help them achieve what matters most.
Welcome to:

Mattermost

Your team communication all in one place, instantly searchable and available anywhere.

Keep your team connected to help them achieve what matters most.
Questions?