Retrospective Sprint 3
of
Group BLOCKS7PG

Block Model Compression Algorithm

a1162576 Karl Asenstorfer

a1806297 Po-Yi Lee

a1804817 Xiaoman Li

a1784375 Yuanpeng Liu

a1782685 Yang Lu

a1797683 Jiaping Qi

a1786785 Hechen Wang

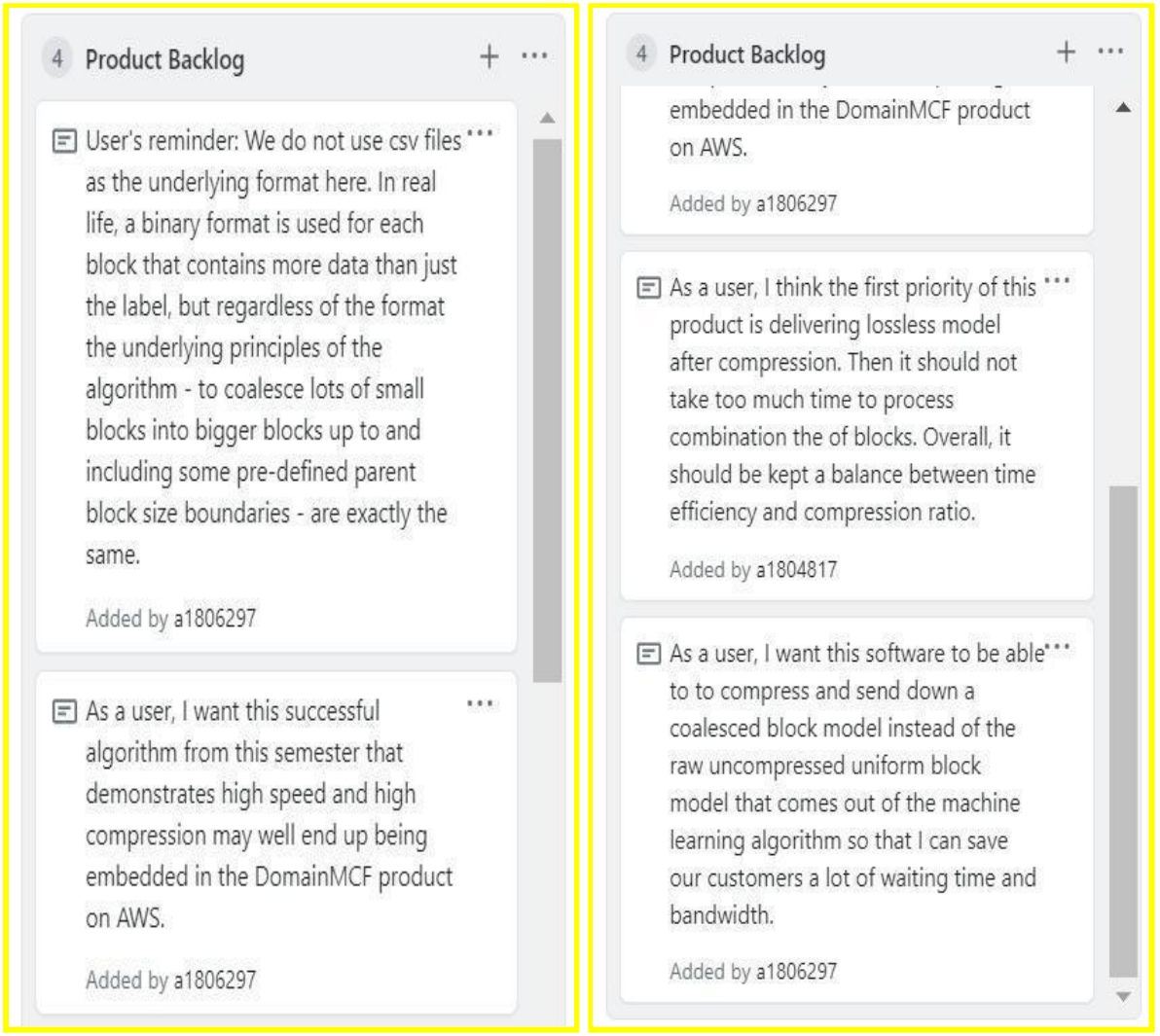
a1784184 Kaiyang Xue

a1811518 Liuyang Yun

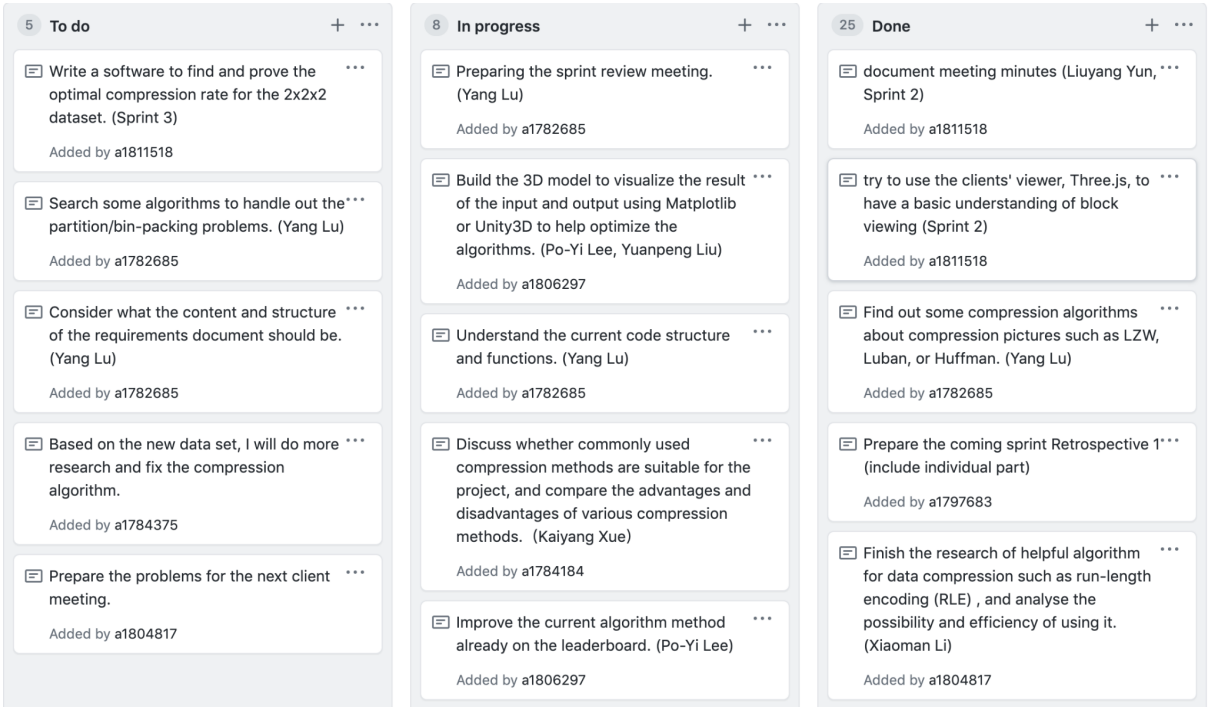
Snapshot Week 7 of Group BLOCKS7PG

Product Backlog and Task Board

Product Backlog:



Task Board:


5 To do
Write a software to find and prove the optimal compression rate for the 2x2x2 dataset. (Sprint 3)
Added by a1811518
Search some algorithms to handle out the partition/bin-packing problems. (Yang Lu)
Added by a1782685
Consider what the content and structure of the requirements document should be. (Yang Lu)
Added by a1782685
Based on the new data set, I will do more research and fix the compression algorithm.
Added by a1784375
Prepare the problems for the next client meeting.
Added by a1804817
8 In progress
Preparing the sprint review meeting. (Yang Lu)
Added by a1782685
Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Po-Yi Lee, Yuanpeng Liu)
Added by a1806297
Understand the current code structure and functions. (Yang Lu)
Added by a1782685
Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)
Added by a1784184
Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
Added by a1806297
25 Done
document meeting minutes (Liuyang Yun, Sprint 2)
Added by a1811518
try to use the clients' viewer, Three.js, to have a basic understanding of block viewing (Sprint 2)
Added by a1811518
Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Yang Lu)
Added by a1782685
Prepare the coming sprint Retrospective 1 (include individual part)
Added by a1797683
Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)
Added by a1804817

8 In progress

- Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)

  Added by a1784184

- Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)

  Added by a1806297

- Try to improve the data structure(Hechen Wang).

  Added by a1786785

- Organize all of these team meetings. (Yang Lu)

  Added by a1782685

- Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer)

  The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.

  Added by a1162576

25 Done

- Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)

  Added by a1782685

- Collect the questions we gonna ask during the client meeting. (Yang Lu)

  Added by a1782685

- Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)

  Added by a1784375

- Determine the members' task in the initial phase (coding and documentation)

  Added by a1806297

- Determine and set up communication channels

  Added by a1162576

- Determine the frequency and time of meetings for member each week

  Added by a1784375

- Summarize the major content of the twice

25 **Done**

Determine the members' task in the initial phase (coding and documentation)

Added by a1806297

Determine and set up communication channels

Added by a1162576

Determine the frequency and time of meetings for member each week

Added by a1784375

Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)

Added by a1782685

Research papers/techniques on block model compression

Added by a1784375

25 **Done**

Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)

Added by a1782685

Research papers/techniques on block model compression

Added by a1784375

Research on the format and meaning of both input data and output data. (Po-Yi Lee)

Added by a1806297

Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ .(Yuanpeng Liu)

Added by a1806297

Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)

Added by a1811518

25 **Done**

Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea)，which might influence the use of algorithm. (Xiaoman Li)

Added by a1804817

document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)

Added by a1811518

Sketch out initial software architecture. (Karl Asenstorfer)

Added by a1162576

Develop Set of Coding Standards. (Karl Asenstorfer)

Added by a1162576

Set up Static Code Analysis. (Karl Asenstorfer)

Added by a1162576

Configure static analysis tools (Karl Asenstorfer)

Added by a1162576

25 **Done**

Develop Set of Coding Standards. (Karl Asenstorfer)

Added by a1162576

Set up Static Code Analysis. (Karl Asenstorfer)

Added by a1162576

Configure static analysis tools (Karl Asenstorfer)

Added by a1162576

Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)

Added by a1804817

Implement the base framework described in System Architecture of initial report

Added by a1162576

Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)

Added by a1162576

Sprint Backlog and User Stories

There are no major modifications to the Sprint Backlog and User Stories, and we are mainly trying to implement the software architecture mentioned in the initial report. The abstraction of the model was to be initialized implementation.

Also, after the client meeting, we realized that we still needed to collect information on various compression algorithms and put them into a wiki page. And to do that, we need a better division of labour

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned.

- relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.
- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a class method to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compression engine is responsible for delegating the compression to the various algorithms. The parent blocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.
- We try to use the clients' viewer, Three.js, to have a basic understanding of block viewing.
- We create a wiki page to collect ideas about compression algorithms.
- We implement the base framework described in the System Architecture of the initial report.
- We extend the modular architecture (merged last week) to use multiprocessing.
- We add more methods for accessing the blocks within a parent block better.
- We add some descriptions of Algorithm ideas on wiki page.
- We clarify the aims for the next sprint after the client meeting.

Summary of Changes:

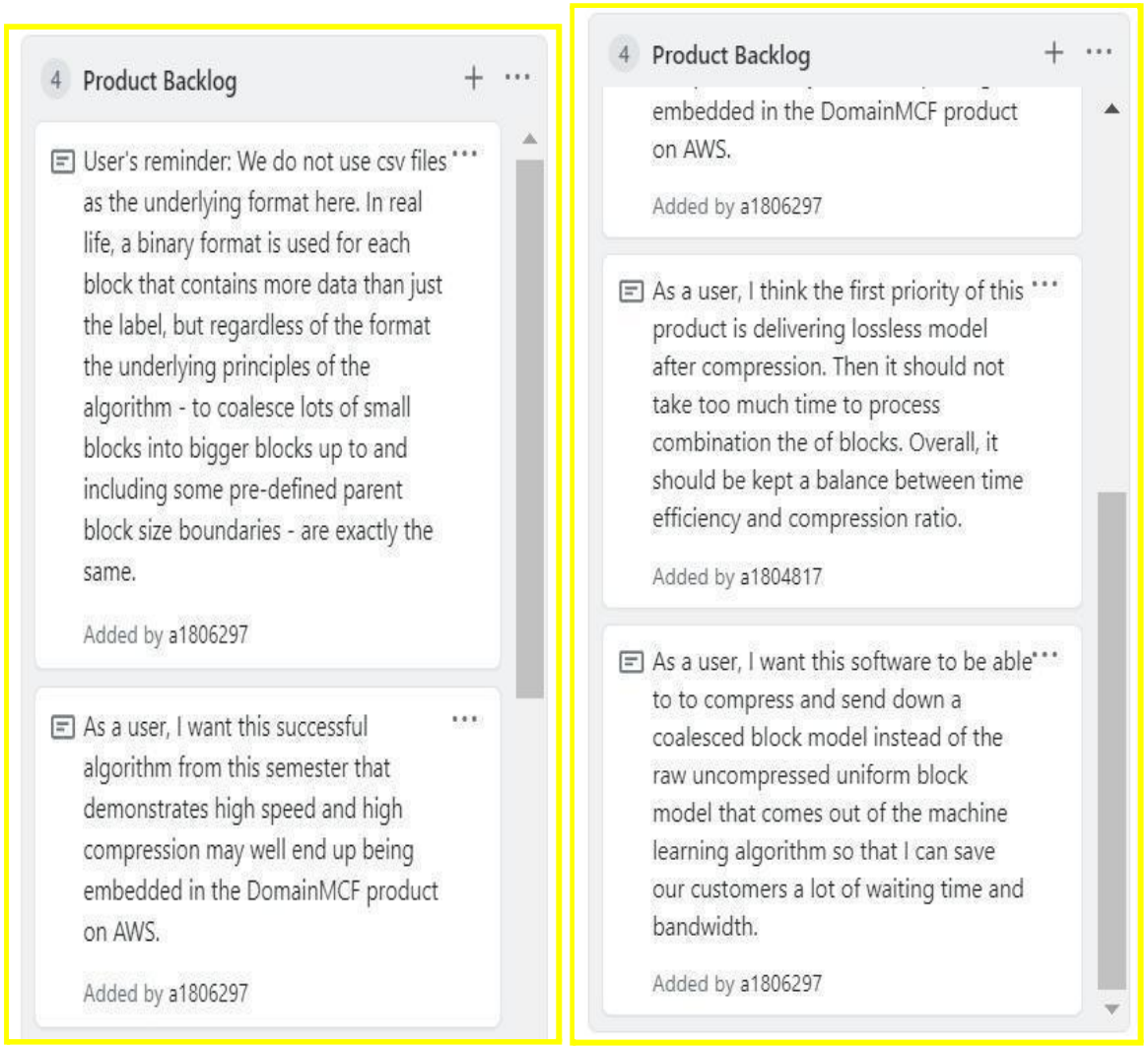
In this week, we extended the modular architecture written by Karl last week, using multiprocessing, and Karl also added some better methods to better access the blocks in the parent block. And we add the information we collected to the wiki page to improve the algorithm, including Recursive Cubes which merges 2x2x2 blocks in the same domain until no cube can be generated; Combine Along 1 Dimension, which starting at the (0,0,0) block and continuing along with one of the dimensions x, y, or z and combining all blocks in the same domain into 1 block; Expanding Blocks, which Starting from a block at (x, y, z), if this block can be merged, it will continue to judge from its adjacent position such as (x+1, y, z) until no more blocks can be added.

In this client meeting, we reported our project progress to the client and clarified the goals for the next sprint, including proving what the optimal compression is for the 2*2*2 data set and submitting an algorithm to compress the new datasets by the end of the next week (submitting and iterating to get improvements to the system).

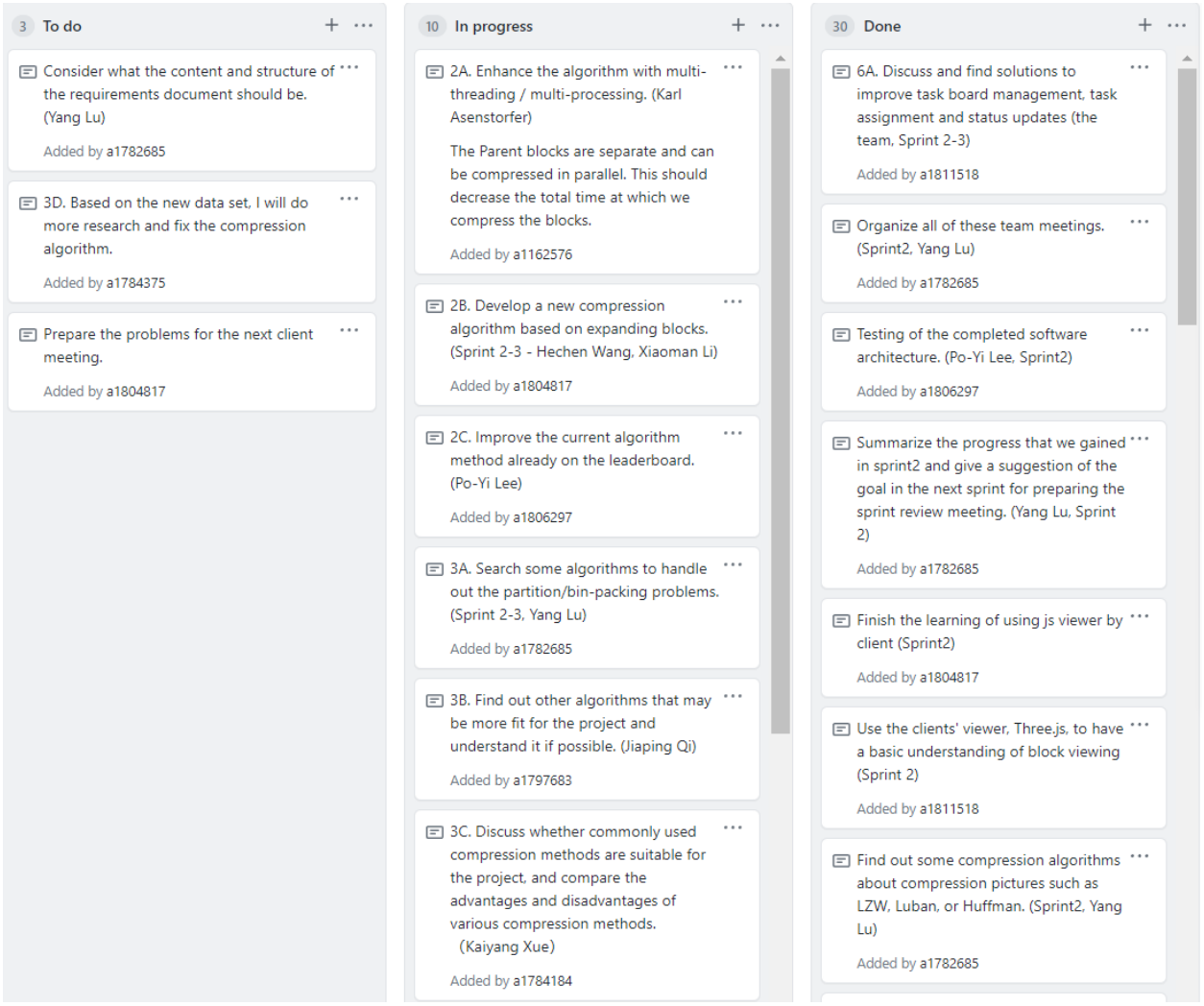
Snapshot Week 8 of Group BLOCKS7PG

Product Backlog and Task Board

Product Backlog:



Task Board:


3 To do
Consider what the content and structure of the requirements document should be. (Yang Lu)
Added by a1782685
3D. Based on the new data set, I will do more research and fix the compression algorithm.
Added by a1784375
Prepare the problems for the next client meeting.
Added by a1804817
10 In progress
2A. Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer)
The Parent blocks are separate and can be compressed in parallel. This should decrease the total time at which we compress the blocks.
Added by a1162576
2B. Develop a new compression algorithm based on expanding blocks. (Sprint 2-3 - Hechen Wang, Xiaoman Li)
Added by a1804817
2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)
Added by a1806297
3A. Search some algorithms to handle out the partition/bin-packing problems. (Sprint 2-3, Yang Lu)
Added by a1782685
3B. Find out other algorithms that may be more fit for the project and understand it if possible. (Jiaping Qi)
Added by a1797683
3C. Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)
Added by a1784184
30 Done
6A. Discuss and find solutions to improve task board management, task assignment and status updates (the team, Sprint 2-3)
Added by a1811518
Organize all of these team meetings. (Sprint2, Yang Lu)
Added by a1782685
Testing of the completed software architecture. (Po-Yi Lee, Sprint2)
Added by a1806297
Summarize the progress that we gained in sprint2 and give a suggestion of the goal in the next sprint for preparing the sprint review meeting. (Yang Lu, Sprint 2)
Added by a1782685
Finish the learning of using js viewer by client (Sprint2)
Added by a1804817
Use the clients' viewer, Three.js, to have a basic understanding of block viewing (Sprint 2)
Added by a1811518
Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, Yang Lu)
Added by a1782685

10 In progress

- 3A. Search some algorithms to handle out the partition/bin-packing problems. (Sprint 2-3, Yang Lu)
  Added by a1782685
- 3B. Find out other algorithms that may be more fit for the project and understand it if possible. (Jiaping Qi)
  Added by a1797683
- 3C. Discuss whether commonly used compression methods are suitable for the project, and compare the advantages and disadvantages of various compression methods. (Kaiyang Xue)
  Added by a1784184
- 4A. Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Po-Yi Lee, Yuanpeng Liu)
  Added by a1806297
- 5A. Write a software that use brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset. (Sprint 3)
  Added by a1811518
- 6B. Add traceability between the goals on the sprint backlog and the tasks (Sprint 3)
  Added by a1811518
- Understand the current code structure and functions. (Yang Lu)
  Added by a1782685

30 Done

- Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, Yang Lu)
  Added by a1782685
- Testing new architecture(Sprint2 - Hechen Wang).
  Added by a1786785
- Prepare the coming sprint Retrospective 1 (include individual part)
  Added by a1797683
- Finish the research of helpful algorithm for data compression such as run-length encoding (RLE) , and analyse the possibility and efficiency of using it. (Xiaoman Li)
  Added by a1804817
- Summarize the content of the client meeting and Upload the recording for the sprint review meeting. (Yang Lu)
  Added by a1782685
- Collect the questions we gonna ask during the client meeting. (Yang Lu)
  Added by a1782685
- Finish the draft program of the compression file of the intro one and the fast one data.(Yuanpeng Liu)
  Added by a1784375
- Determine the members' task in the initial phase (coding and documentation)
  Added by a1806297

30 Done

- Determine the members' task in the initial phase (coding and documentation)
  Added by a1806297
- Determine and set up communication channels
  Added by a1162576
- Determine the frequency and time of meetings for member each week
  Added by a1784375
- Summarize the major content of the twice team meeting and ask for other ideas for the rest of the team members. (Yang Lu)
  Added by a1782685
- Research papers/techniques on block model compression
  Added by a1784375
- Research on the format and meaning of both input data and output data. (Po-Yi Lee)
  Added by a1806297
- Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)
  Added by a1806297

30 Done

Upload the valid code on the competition website, shake out your environment and let us test a first solution at https://titan.maptek.net/ . (Yuanpeng Liu)

Added by a1806297

Collected information on the contents of the snapshot and the initial report from the Q&As on discussion board and use the tips to check whether our drafts meet the requirements. (Po-Yi Lee, Liuyang Yun, Sprint 1)

Added by a1811518

Identify the user needs as well as the structure of the model (the colours of blocks are distributed randomly or more integrated like land and the sea), which might influence the use of algorithm. (Xiaoman Li)

Added by a1804817

document meeting minutes (Yang Lu, Liuyang Yun, Sprint 1)

Added by a1811518

Sketch out initial software architecture. (Karl Asenstorfer)

Added by a1162576

Develop Set of Coding Standards. (Karl Asenstorfer)

Added by a1162576

Set up Static Code Analysis. (Karl Asenstorfer)

Added by a1162576

30 Done

Added by a1811518

Sketch out initial software architecture. (Karl Asenstorfer)

Added by a1162576

Develop Set of Coding Standards. (Karl Asenstorfer)

Added by a1162576

Set up Static Code Analysis. (Karl Asenstorfer)

Added by a1162576

Configure static analysis tools (Karl Asenstorfer)

Added by a1162576

Develop initial algorithm that might contribute to the project (traditional algorithm like brute force or greedy might be low effciency)

Added by a1804817

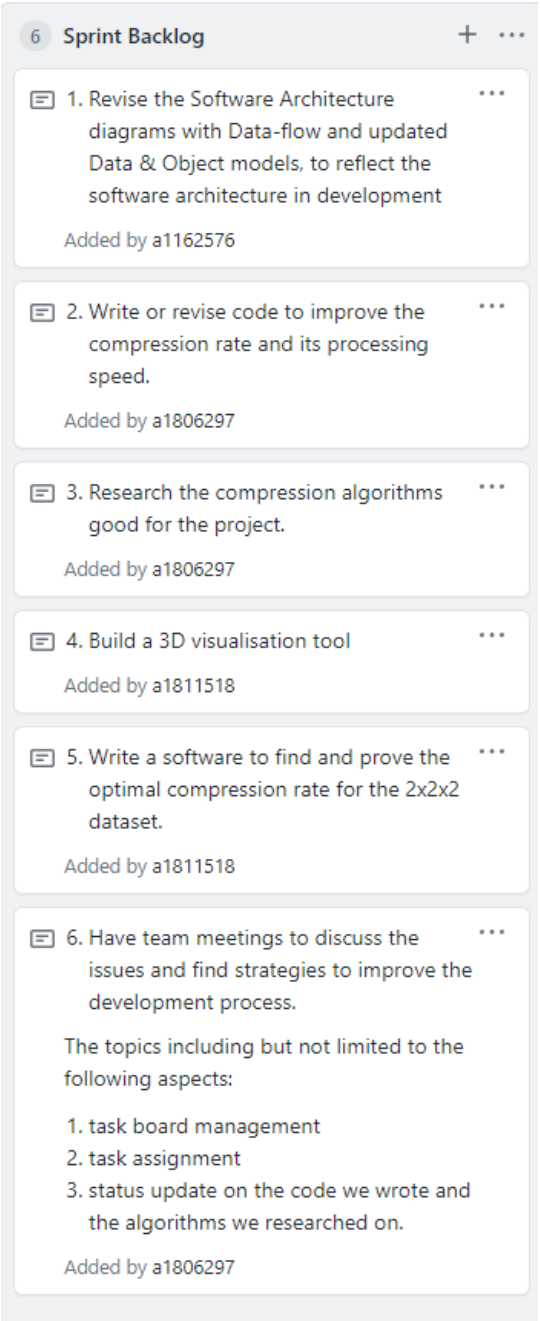
Implement the base framework described in System Architecture of initial report

Added by a1162576

Create wiki page to collect ideas about compression algorithms (Karl Asenstorfer)

Added by a1162576

Sprint Backlog and User Stories



For this week, there are no major changes to the sprint backlog. The main task currently is to optimise the current algorithm based on the existing software architecture. According to the discussion in this week's sprint review meeting, we came up with several new points for the standard of the task board and the plan of deliverables for the coming sprint. Moreover, each member of the group should be familiar with the use of the latest Modular Compressor Architecture.

Definition of Done

In the current phase:

- The code we develop is required to take standard input (strings of the form "x, y, z, x_size, y_size, z_size, 'domain'") and produce the result on standard output described in the project documentation.
- Either a .exe file or a Python script must be submitted to a verification service: MAPTEK TITAN.
- According to the user stories, we can submit our code once it improves the compression rate and processing speed, no matter how good they are.
- The datasets of input block models we implement must be comma-separated values (CSV) where each line encodes a block as a string of the form "x, y, z, x_size, y_size, z_size, 'domain'" and the code we develop is required to output a stream of the same format.
- The algorithm we develop must process a block model in slices of no more than parent block thickness at a time, rather than loading the entire input stream into memory first.
- All pull requests of code must be submitted on our GitHub repository, tested, and reviewed by two other team members (or one in the case of documentation/admin).
- The branches must pass the static code analysis before being merged. The Static Analysis ensures that the codes meet the PEP8 style standards (what all python code should aim for) and other issues such as unused variables and cyclomatic complexity. PEP8 is especially important for the developers because it mandates a set of conventions for things such as class/function names, spacing, and comment style. It allows all the developers to be able to read every developer's code in the same way.
- In the initialization implementation of the abstract classes of the model, we build the Model class. The Model is the internal representation of the block data model. The Model holds the current set of ParentBlocks which constitute the current slice of the model. The model also holds the mappings between domains and domain tags, which the Blocks use.
- In the initialization implementation of the abstract classes of the model, we build a ParentBlocks class. The ParentBlocks subdivide the model exactly with no remainder. Each Parent Block contains a collection of Blocks. Each Parent Block has a size and position. All of the Blocks that the Parent Block contains are positioned.

- relative to the ParentBlock position. The ParentBlock implements the iterator protocol, so can be iterated over natively in for loops.
- We also created the Block class in the abstraction part. This class describes a block, which has a size, a position relative to the start of its ParentBlock, and a domain tag.
- There is also a class method to combine all of those blocks and return them into a new block.
- We initialized the implementation of the modular compression system. All the compression algorithms are separated into modules. Each one takes as input a ParentBlock. The compression engine is responsible for delegating the compression to the various algorithms. The parent blocks are then written to the output.
- The Modular Architecture has been implemented, but it still needs to be tested with the runner.py in the Windows environment and then try to merge it.
- We also clarified some information in README, created requirements.txt, and configured flake8.
- We try to use the clients' viewer, Three.js, to have a basic understanding of block viewing.
- We create wiki page to collect ideas about compression algorithms.
- We implement the base framework described in the System Architecture of the initial report.
- We extend the modular architecture (merged last week) to use multiprocessing.
- We add more methods for accessing the blocks within a parent block better.
- We add some descriptions of Algorithm ideas on wiki page.
- We clarify the aims for the next sprint after the client meeting.
- We implement the Multiprocessing as well as the Greedy-Expander Algorithm.
- We add traceability between the sprint backlog and product backlog.
- We add the estimate of time and record the actual time on task board for comparison.
- We develop the algorithm to find and prove the best compression rate.
- We add the deadline to the tasks

Summary of Changes:

In this week, we mainly focused on the refinement of the current solution based on existing software architecture developed by Karl, which aims to process the new dataset with larger size. The use of developed Greedy-Expander Algorithm and Multiprocessing have been implemented so that the efficiency of the current algorithm could be improved a lot. Besides, writing the code based on brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset was another task set for this sprint.

In the sprint review meeting this week, we have proposed some points that might contribute to team cooperation and communication, as well as the changes about agile management in the coming sprint. Firstly, for the communication, there would be a short update on the individual tasks every 3-4 days on slack. For the algorithm, we improved the performance of the algorithm on the first two datasets and tried to apply the improved algorithms on the new datasets. And there are more detailed changes considered like adding the deadline of the tasks in the task board to compare the estimated and actual spent hours. As the mention of feedback, we also add the traceability between your sprint backlog and product backlog so that to follow up the progress.

Declaration:

I attended:

- the sprint 3 review and sprint 4 planning meeting on 8th October, 2021 with the tutors, Aryaman Dhawan and Will Reid.
- the sprint 3 retrospective meeting on 9th October, 2021 with the other team members.

What went well in the sprint:

In the sprint 3, we successfully completed the multiprocessing compression system using a modular architecture and some layers of abstraction to simplify the development or compression algorithms. This system is able to decide which algorithm is applied to which parent blocks and apply these algorithms to the parent blocks in parallel using multiprocessing which can help the team to easily add a new compression algorithm in a simple way. With this multiprocessing system, we developed a new algorithm, figured out how to upload the executable files in a zipped folder and updated our competition dashboard with the higher compression rates on the dataset including the_intro_one(93.7378 %), the_fast_one(79.6388 %) and the_stratal_one(98.9630 %). In addition, we added the traceability between the sprint backlog and task boards by numbering the most of tasks in the

to-do and in-process lists which help the team understand the sprint information to each task more clearly. Lastly, the 3D model visualization has been accomplished with the smaller size of the dataset. Once it can be implemented with a bigger dataset, it could be used to see how the compression outcomes look after the blocks are compressed with the algorithms we developed and help the optimization of algorithms in the future.

What could be improved:

In the last sprint, the team proposed that every team member should update the team on what they have been doing and any issues they've had twice per week no matter what they actually contribute on. However, the fact was that there were still few random updates from a few members rather than the regular state updates. Furthermore, during the period of the holiday, we could have been tracing each member of the team because there was not too much progress made by the team in the two week's mid break. I think that the scrum master could have asked the team more frequently what progress we had and also we could have been more active with what the other members proposed the questions or new progress because there were not too many responses from the other members when the new contributions were published on Slack.

What will the group commit to improve in the next sprint:

According to the sprint 3 retrospective meeting and the customer's requirement from the sprint 4 planning meeting, we will:

1. revise the software architecture diagrams with data-flow and updated data and object models to reflect the software architecture in development and to understand the architecture better for the team and the final report and presentation.
2. have the extra meeting joined by the members charged in the same task to not only discuss revising the existing code but also research the new compression algorithms in order to improve the compression rate and its processing speed.
3. build the 3D model visualization which can be used on a bigger size of dataset by testing the visualization tool with different dataset.
4. do the research on how to implement our code from CPU to GPU to satisfy the customer's requirement.

Comment on your progress this sprint:

My in-progress tasks:

- Build the 3D model to visualize the result of the input and output using Matplotlib or Unity to help optimize the algorithms
- Manage to Improve the current algorithm method already on the leaderboard.

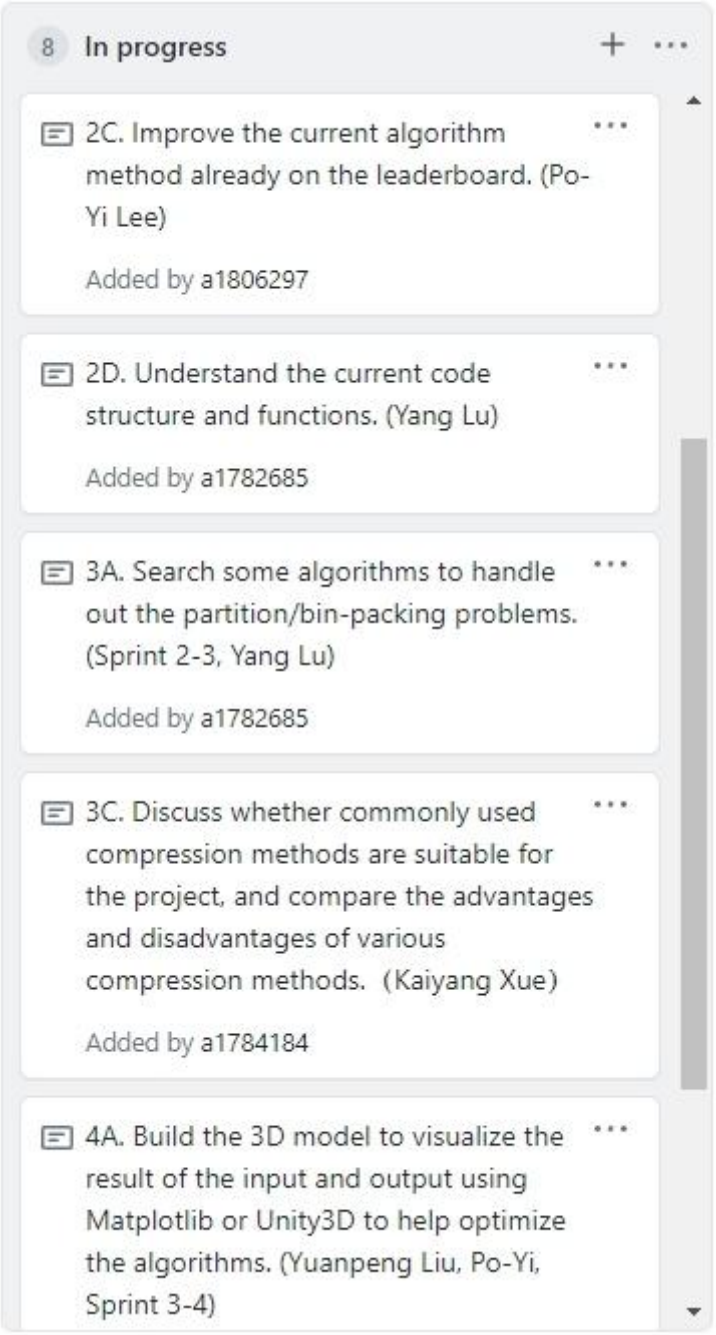


Figure 1: My in-progress tasks

In this sprint, I have been revising the original compression algorithms and finding the optimal one with the multiprocessing system to develop better compression methods. In addition, the visualization model of compressed block using Unity is still in progress because we are still trying to figure out how to implement the visualization model on a bigger size of dataset.

Requirements Changes:

According to the customer's requirement for the next sprint, we should convert the CPU implementation into the GPU implementation for our model compression system. Since our team has been using Python to run our compression algorithms and compress the given input datasets, we will use Numba—a Python compiler from Anaconda that can compile Python code for execution on CUDA-capable GPUs even though as an interpreted language, it's been considered too slow for high-performance computing. In the sprint 4, we should get familiar with how to use the library of Numba and CUDA Python and combine them with our existing or updated code for model compression. Firstly, to get started with Numba, we should install the Anaconda Python distribution and the required CUDA packages. And later we should write the outline and ideas for the combinations between the compression algorithm and CUDA Python, submit to the customer, get the feedback from them and eventually manage to accomplish the code.