Retrospective Sprint 4
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

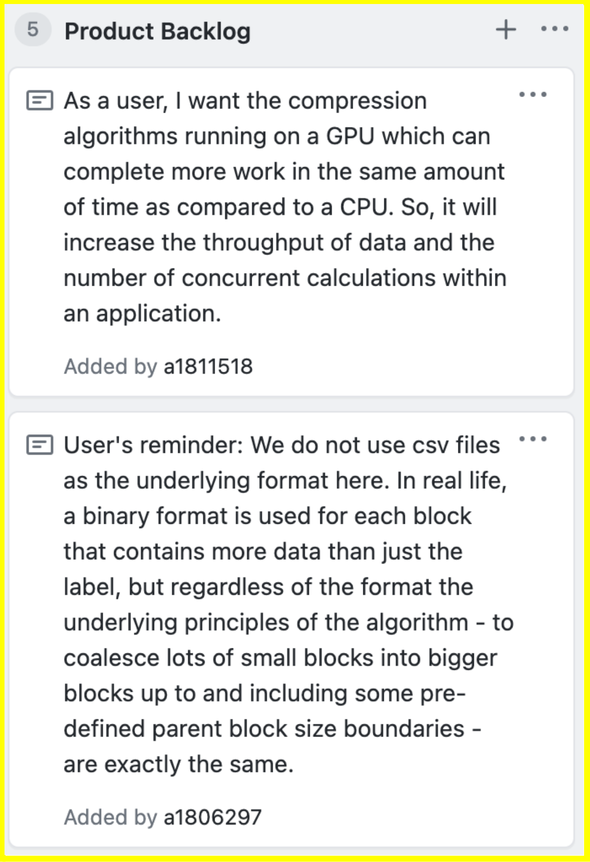
a1784184 Kaiyang Xue

a1811518 Liuyang Yun

Snapshot Week 9 of Group BLOCKS7PG

Product Backlog and Task Board

Product Backlog:



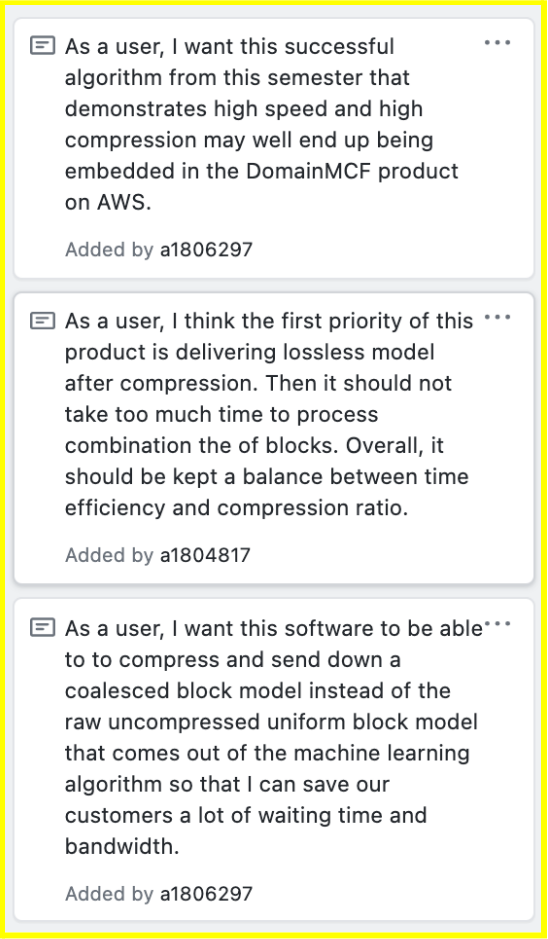
Product Backlog

As a user, I want the compression algorithms running on a GPU which can complete more work in the same amount of time as compared to a CPU. So, it will increase the throughput of data and the number of concurrent calculations within an application.

Added by a1811518

User's reminder: We do not use csv files as the underlying format here. In real life, a binary format is used for each block that contains more data than just the label, but regardless of the format the underlying principles of the algorithm - to coalesce lots of small blocks into bigger blocks up to and including some pre-defined parent block size boundaries - are exactly the same.

Added by a1806297



As a user, I want this successful algorithm from this semester that demonstrates high speed and high compression may well end up being embedded in the DomainMCF product on AWS.

Added by a1806297

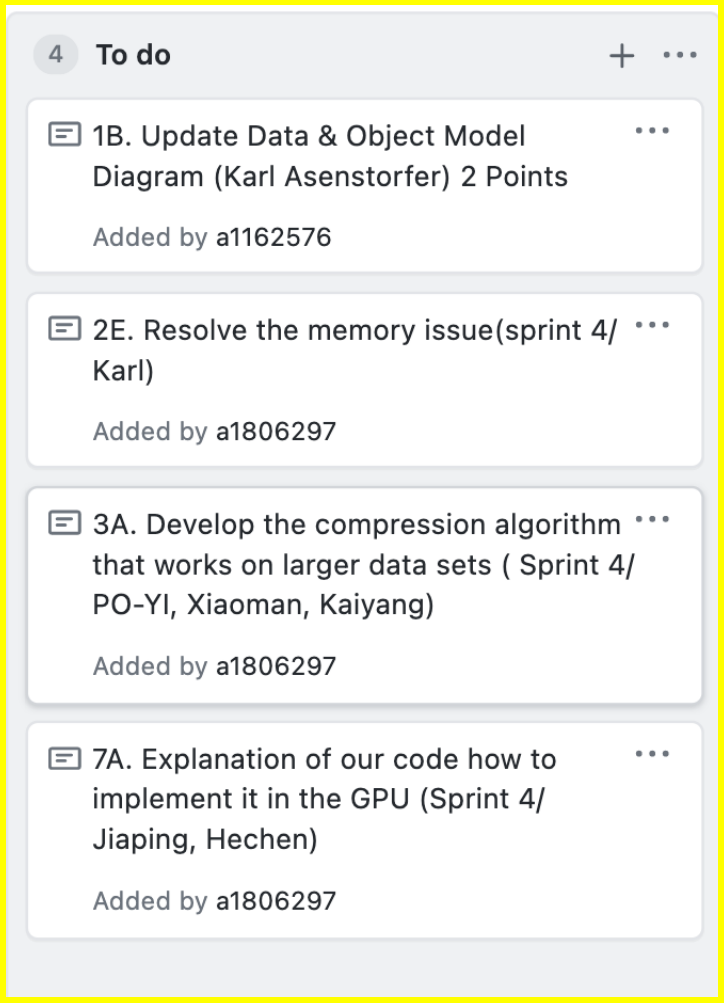
As a user, I think the first priority of this product is delivering lossless model after compression. Then it should not take too much time to process combination the of blocks. Overall, it should be kept a balance between time efficiency and compression ratio.

Added by a1804817

As a user, I want this software to be able to to compress and send down a coalesced block model instead of the raw uncompressed uniform block model that comes out of the machine learning algorithm so that I can save our customers a lot of waiting time and bandwidth.

Added by a1806297

Task Board:



4 To do

1B. Update Data & Object Model Diagram (Karl Asenstorfer) 2 Points

Added by a1162576

2E. Resolve the memory issue(sprint 4/ Karl)

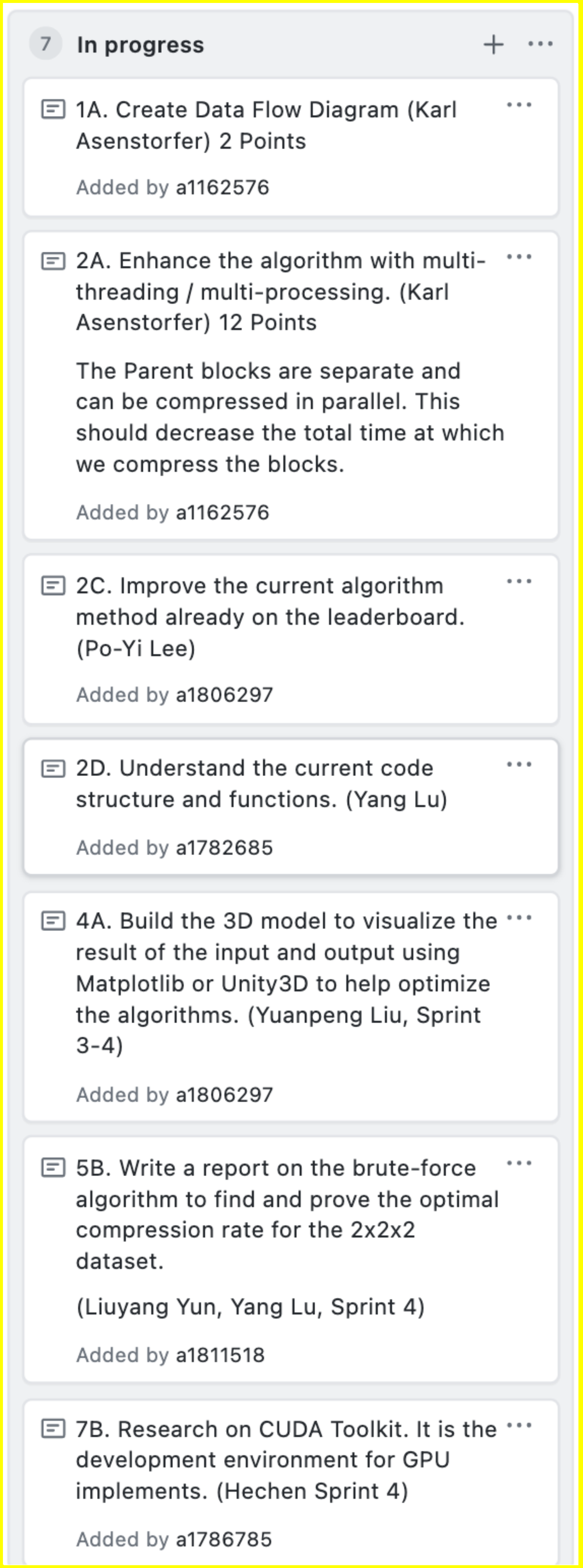
Added by a1806297

3A. Develop the compression algorithm that works on larger data sets (Sprint 4/ PO-YI, Xiaoman, Kaiyang)

Added by a1806297

7A. Explanation of our code how to implement it in the GPU (Sprint 4/ Jiaping, Hechen)

Added by a1806297



In progress

1A. Create Data Flow Diagram (Karl Asenstorfer) 2 Points

Added by a1162576

2A. Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer) 12 Points

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

2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)

Added by a1806297

2D. Understand the current code structure and functions. (Yang Lu)

Added by a1782685

4A. Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Yuanpeng Liu, Sprint 3-4)

Added by a1806297

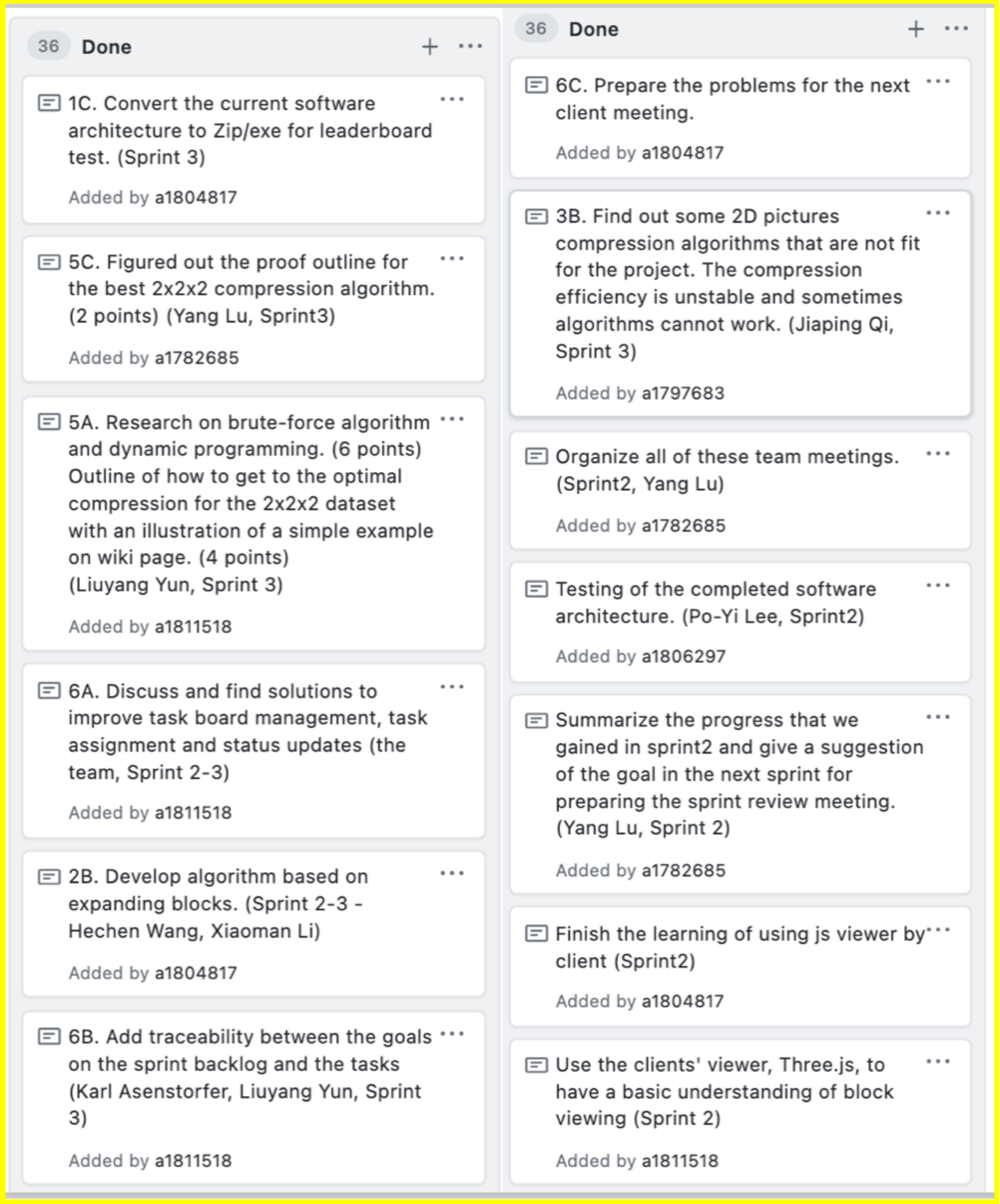
5B. Write a report on the brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset.

(Liuyang Yun, Yang Lu, Sprint 4)

Added by a1811518

7B. Research on CUDA Toolkit. It is the development environment for GPU implements. (Hechen Sprint 4)

Added by a1786785



36 Done

1C. Convert the current software architecture to Zip/exe for leaderboard test. (Sprint 3)

Added by a1804817

5C. Figured out the proof outline for the best 2x2x2 compression algorithm. (2 points) (Yang Lu, Sprint3)

Added by a1782685

5A. Research on brute-force algorithm and dynamic programming. (6 points) Outline of how to get to the optimal compression for the 2x2x2 dataset with an illustration of a simple example on wiki page. (4 points)
(Liuyang Yun, Sprint 3)

Added by a1811518

6A. Discuss and find solutions to improve task board management, task assignment and status updates (the team, Sprint 2-3)

Added by a1811518

2B. Develop algorithm based on expanding blocks. (Sprint 2-3 - Hechen Wang, Xiaoman Li)

Added by a1804817

6B. Add traceability between the goals on the sprint backlog and the tasks (Karl Asenstorfer, Liuyang Yun, Sprint 3)

Added by a1811518

36 Done

6C. Prepare the problems for the next client meeting.

Added by a1804817

3B. Find out some 2D pictures compression algorithms that are not fit for the project. The compression efficiency is unstable and sometimes algorithms cannot work. (Jiaping Qi, Sprint 3)

Added by a1797683

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)

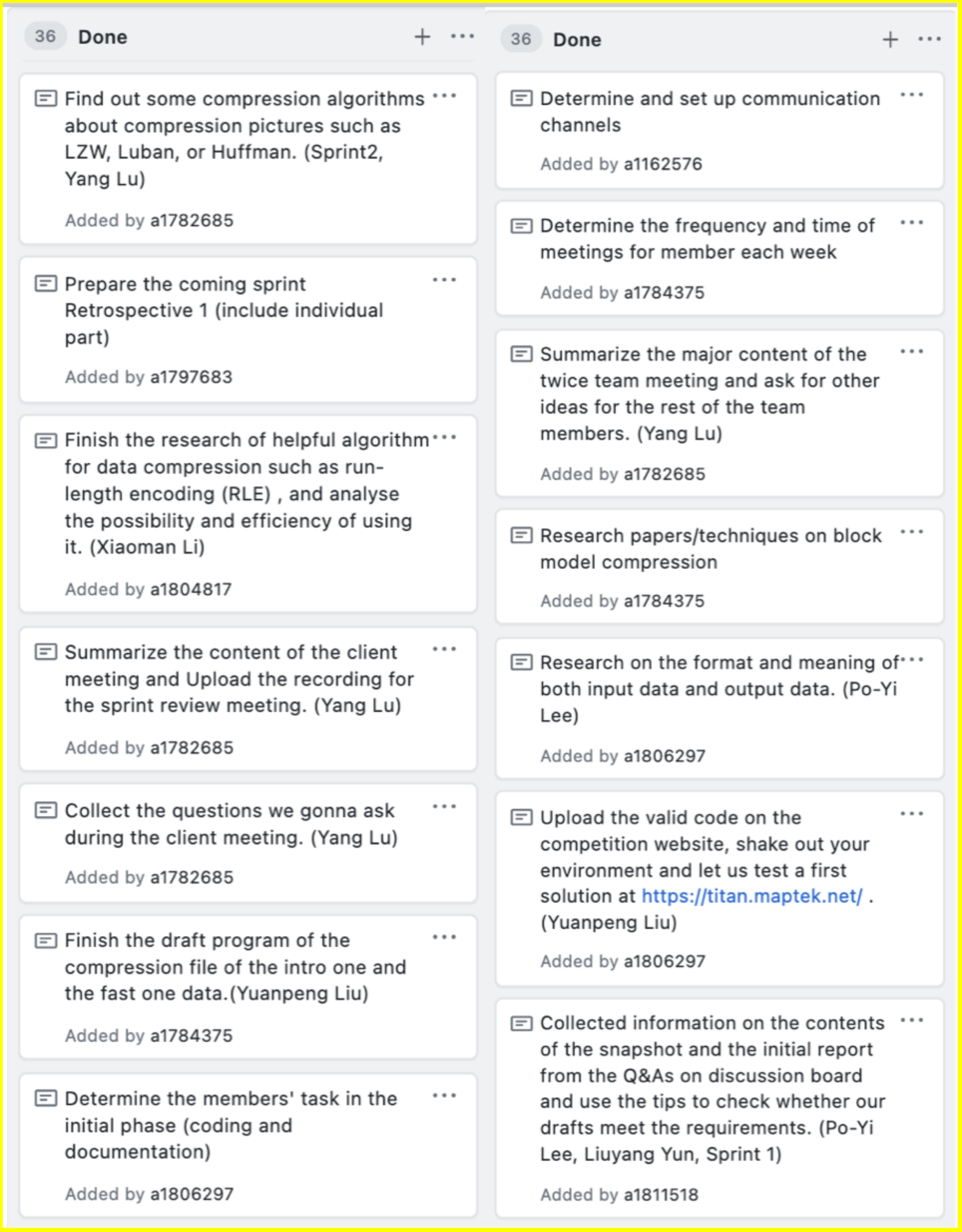
Added by a1782685

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



36 Done

Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, 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

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

36 Done

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)

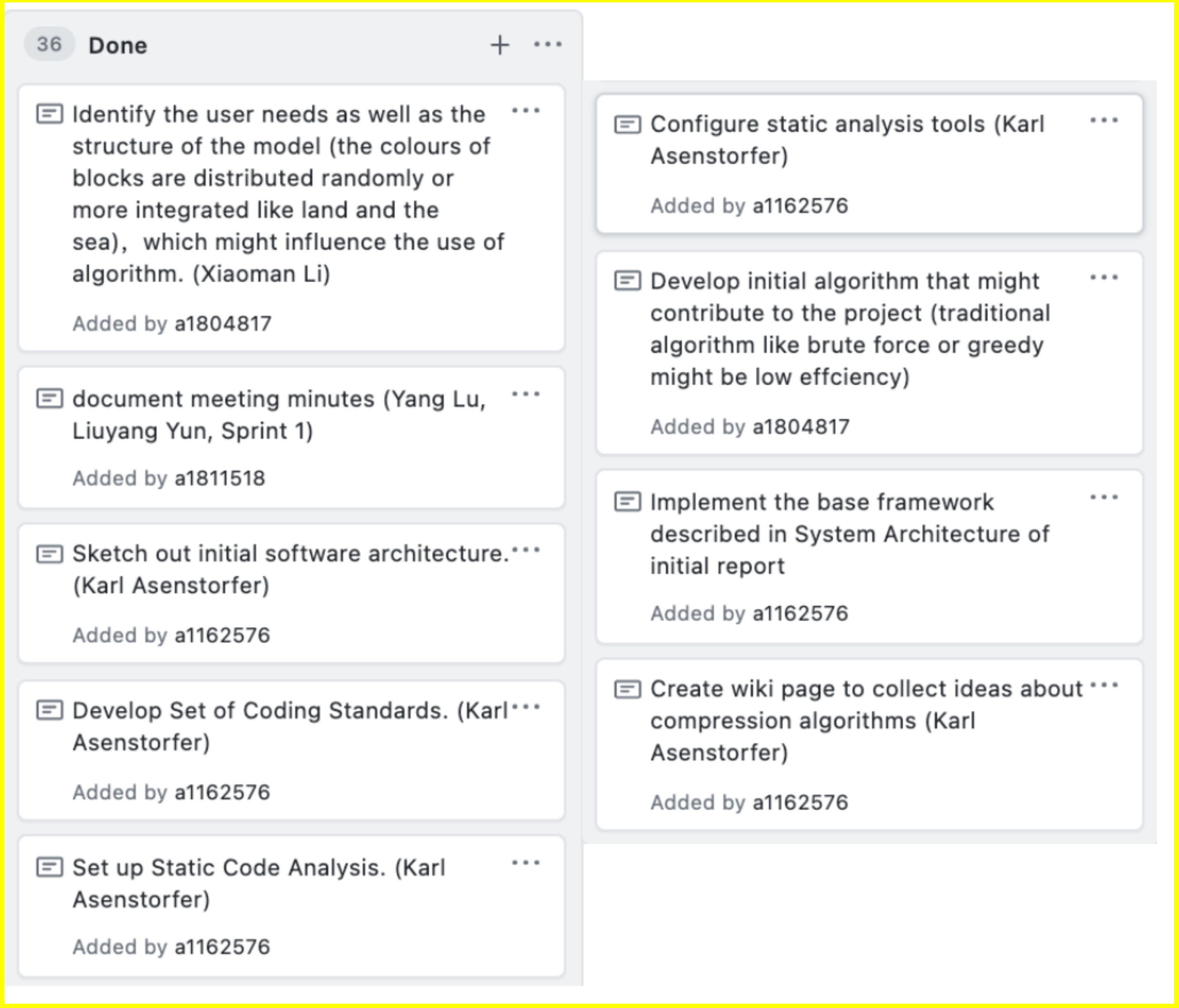
Added by a1806297

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

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



36 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

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

Added by a1804817

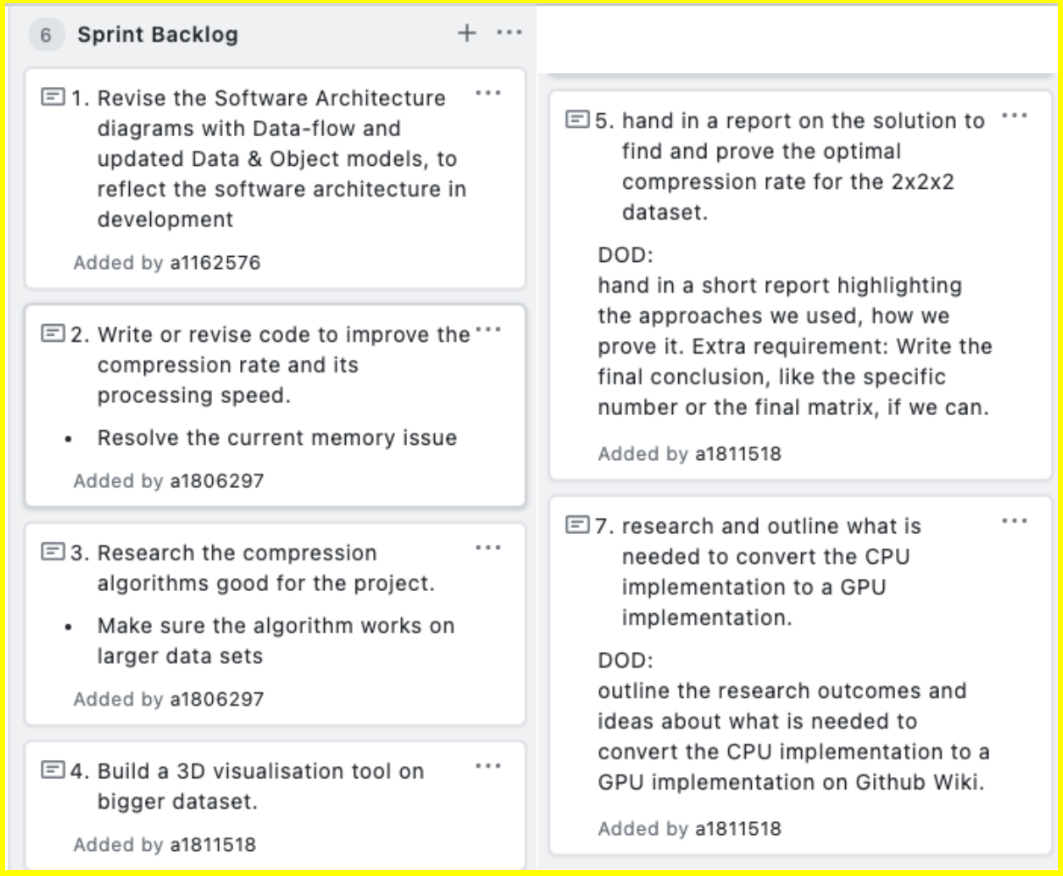
Implement the base framework described in System Architecture of initial report

Added by a1162576

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

Added by a1162576

Sprint Backlog and User Stories



6 Sprint Backlog

1. Revise the Software Architecture diagrams with Data-flow and updated Data & Object models, to reflect the software architecture in development

Added by a1162576

2. Write or revise code to improve the compression rate and its processing speed.

- Resolve the current memory issue

Added by a1806297

3. Research the compression algorithms good for the project.

- Make sure the algorithm works on larger data sets

Added by a1806297

4. Build a 3D visualisation tool on bigger dataset.

Added by a1811518

5. hand in a report on the solution to find and prove the optimal compression rate for the 2x2x2 dataset.

DOD:
hand in a short report highlighting the approaches we used, how we prove it. Extra requirement: Write the final conclusion, like the specific number or the final matrix, if we can.

Added by a1811518

7. research and outline what is needed to convert the CPU implementation to a GPU implementation.

DOD:
outline the research outcomes and ideas about what is needed to convert the CPU implementation to a GPU implementation on Github Wiki.

Added by a1811518

This week, there are both user stories rolled from the previous sprint and new user stories on the sprint backlog. We have four major goals in the next sprint: 1. Write or revise code to improve the compression rate and its processing speed, and **resolve the memory issue on the current algorithm**; 2. Develop a compression algorithm that can handle the big_one dataset; 3. Hand in a report on the solution to find and prove the optimal compression rate for the 2x2x2 dataset; 4. **research and outline what is needed to convert the CPU implementation to a GPU implementation**.

Since there are four main tasks, we should have a clear division of personnel and improve our communication in the next sprint.

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 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.
- For the task to improve the task board management: 1) add traceability between the sprint backlog and product backlog; 2) add the estimate of time and record the actual time on task board for comparison. 3) We add the deadline to the tasks.
- For the task to find the optimal compression algorithm: hand in a short report highlighting the approaches we used and how we prove it.
- For the GPU task: outline the research outcomes and ideas about what is needed to convert the CPU implementation to a GPU implementation on Github Wiki.

Summary of Changes:

Following changes happened since the last snapshot:

What we have done in this week

- We attended the sprint review and planning meeting and had a group meeting to discuss the division of personnel to fulfill the four goals for sprint 4;
- We submitted the new algorithm to Titan, and it passed the first three datasets;
- We updated the outline of how to get to the optimal compression algorithm for the 2x2x2 dataset on the GitHub wiki page.
- Yuanpeng has built a 3D visualisation tool that will work on small datasets;

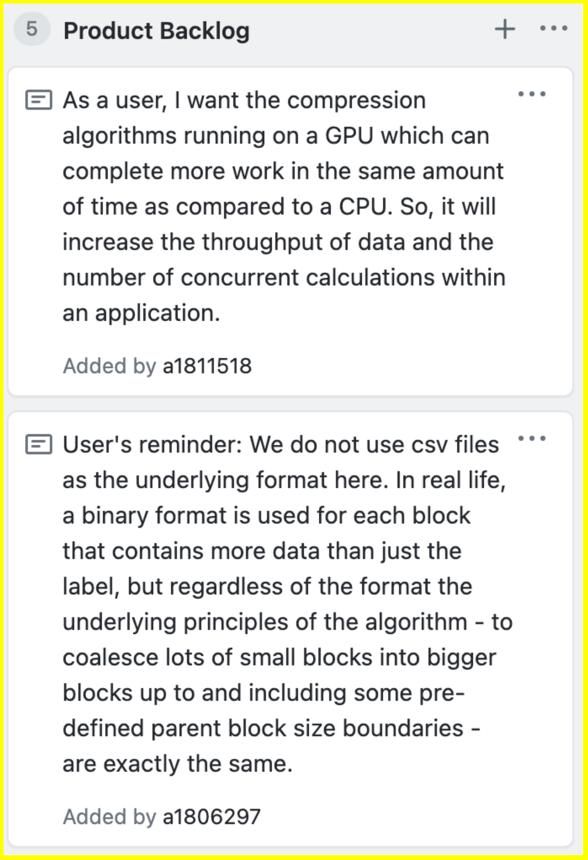
changes on client requirements:

- The ultimate solution for the project is running on a GPU. We need to research and develop some ideas on how to convert our current algorithm from a CPU implementation to a GPU implementation.

Snapshot Week 10 of Group BLOCKS7PG

Product Backlog and Task Board

Product Backlog:



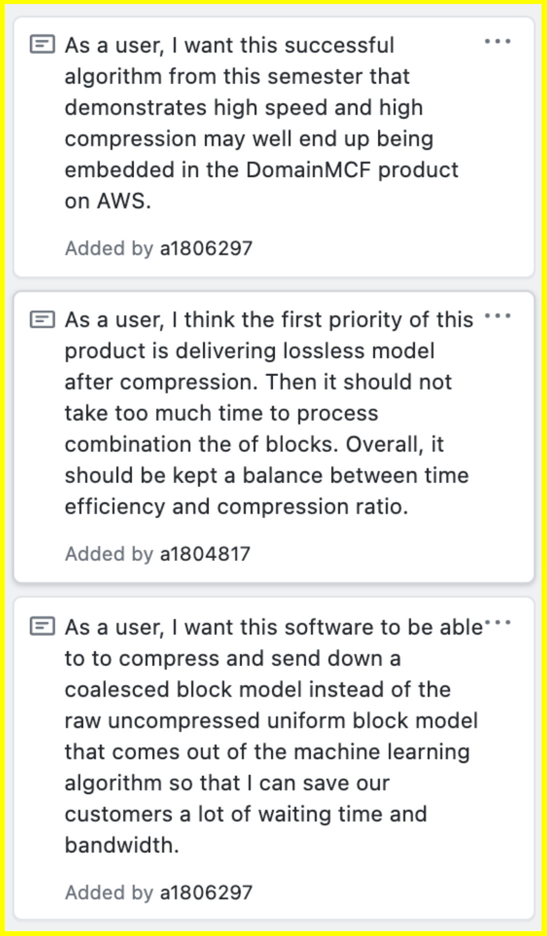
5 Product Backlog

As a user, I want the compression algorithms running on a GPU which can complete more work in the same amount of time as compared to a CPU. So, it will increase the throughput of data and the number of concurrent calculations within an application.

Added by a1811518

User's reminder: We do not use csv files as the underlying format here. In real life, a binary format is used for each block that contains more data than just the label, but regardless of the format the underlying principles of the algorithm - to coalesce lots of small blocks into bigger blocks up to and including some pre-defined parent block size boundaries - are exactly the same.

Added by a1806297



As a user, I want this successful algorithm from this semester that demonstrates high speed and high compression may well end up being embedded in the DomainMCF product on AWS.

Added by a1806297

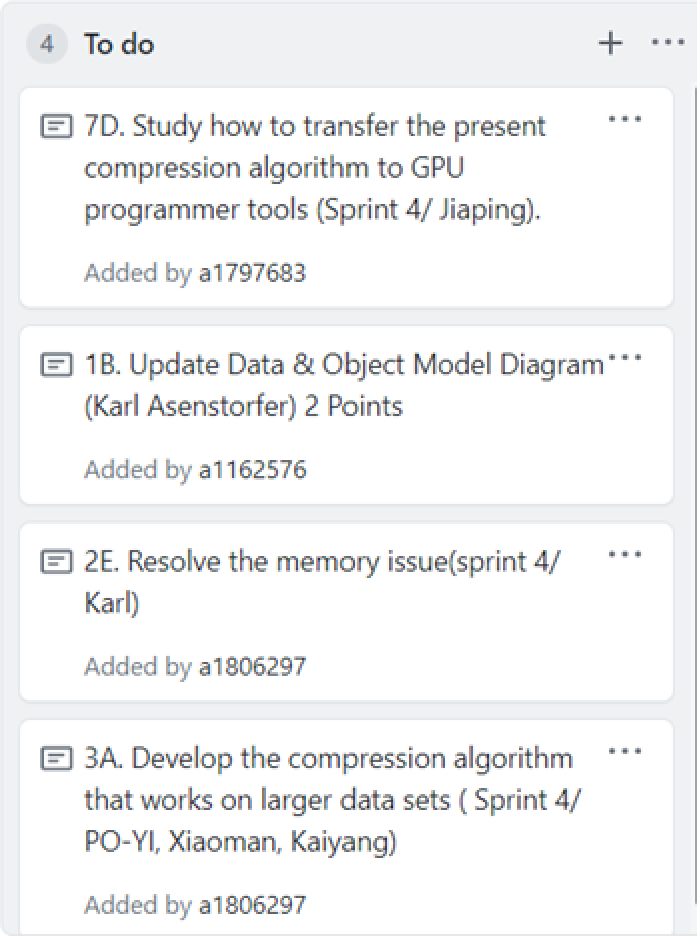
As a user, I think the first priority of this product is delivering lossless model after compression. Then it should not take too much time to process combination the of blocks. Overall, it should be kept a balance between time efficiency and compression ratio.

Added by a1804817

As a user, I want this software to be able to to compress and send down a coalesced block model instead of the raw uncompressed uniform block model that comes out of the machine learning algorithm so that I can save our customers a lot of waiting time and bandwidth.

Added by a1806297

Task Board:



4 To do

7D. Study how to transfer the present compression algorithm to GPU programmer tools (Sprint 4/ Jiaping).

Added by a1797683

1B. Update Data & Object Model Diagram (Karl Asenstorfer) 2 Points

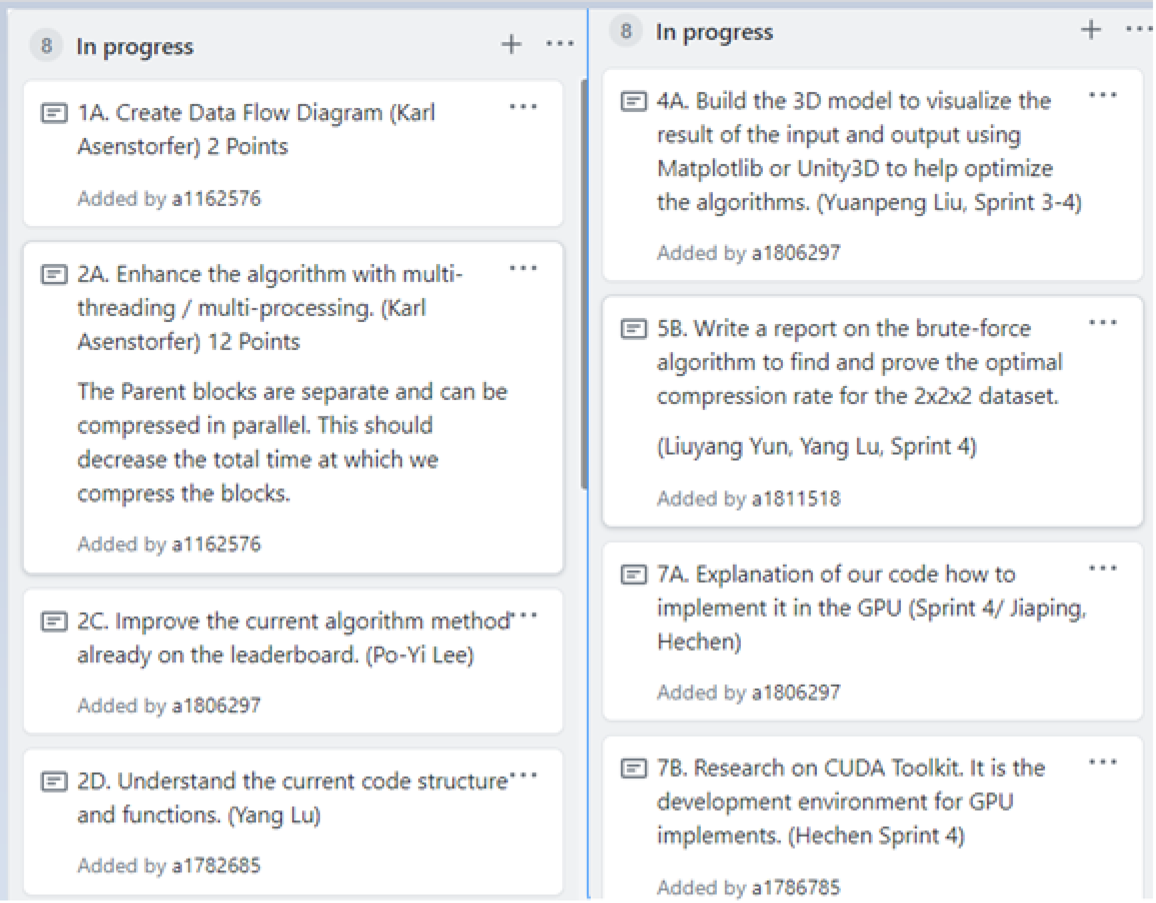
Added by a1162576

2E. Resolve the memory issue(sprint 4/ Karl)

Added by a1806297

3A. Develop the compression algorithm that works on larger data sets (Sprint 4/ PO-YI, Xiaoman, Kaiyang)

Added by a1806297



8 In progress

1A. Create Data Flow Diagram (Karl Asenstorfer) 2 Points

Added by a1162576

2A. Enhance the algorithm with multi-threading / multi-processing. (Karl Asenstorfer) 12 Points

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

2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)

Added by a1806297

2D. Understand the current code structure and functions. (Yang Lu)

Added by a1782685

8 In progress

4A. Build the 3D model to visualize the result of the input and output using Matplotlib or Unity3D to help optimize the algorithms. (Yuanpeng Liu, Sprint 3-4)

Added by a1806297

5B. Write a report on the brute-force algorithm to find and prove the optimal compression rate for the 2x2x2 dataset.

(Liuyang Yun, Yang Lu, Sprint 4)

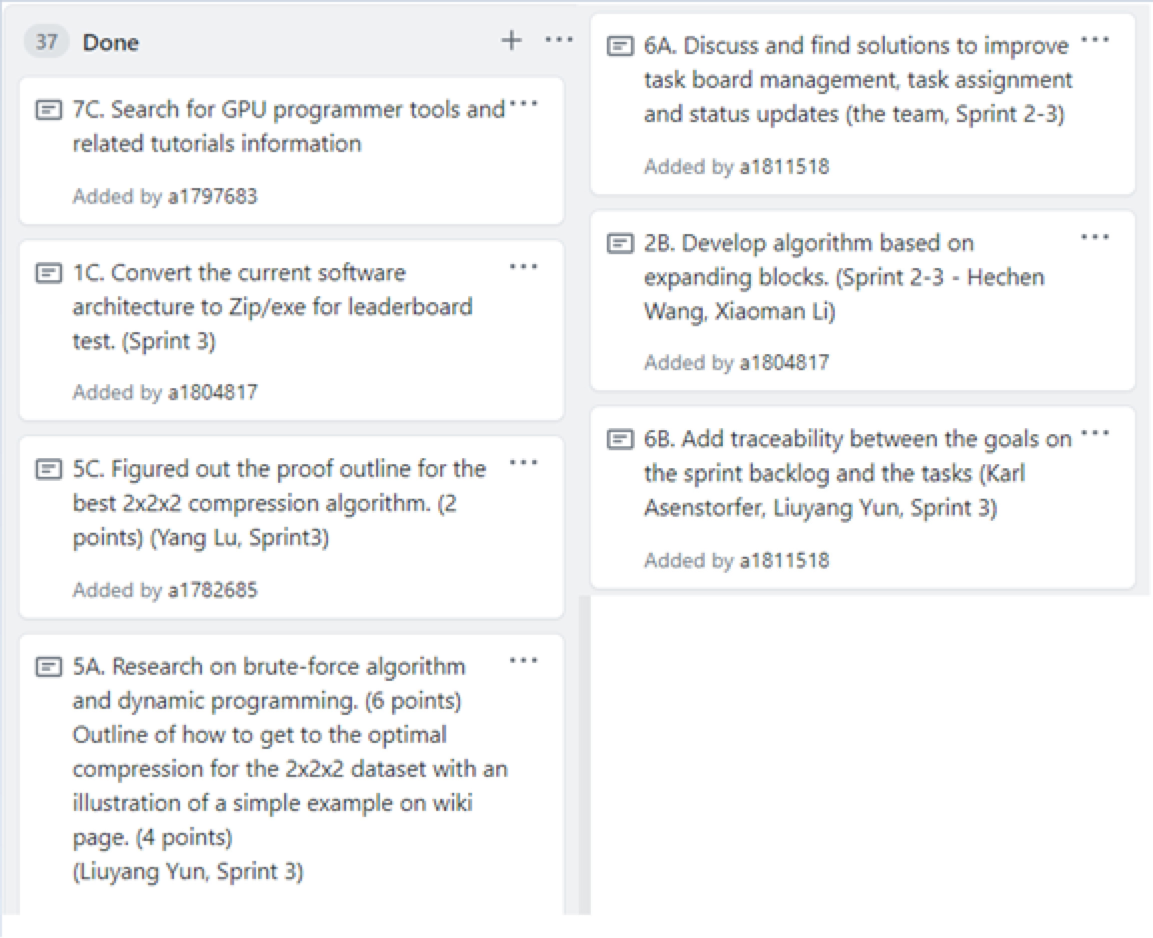
Added by a1811518

7A. Explanation of our code how to implement it in the GPU (Sprint 4/ Jiaping, Hechen)

Added by a1806297

7B. Research on CUDA Toolkit. It is the development environment for GPU implements. (Hechen Sprint 4)

Added by a1786785



37 Done

7C. Search for GPU programmer tools and related tutorials information

Added by a1797683

1C. Convert the current software architecture to Zip/exe for leaderboard test. (Sprint 3)

Added by a1804817

5C. Figured out the proof outline for the best 2x2x2 compression algorithm. (2 points) (Yang Lu, Sprint3)

Added by a1782685

5A. Research on brute-force algorithm and dynamic programming. (6 points)
Outline of how to get to the optimal compression for the 2x2x2 dataset with an illustration of a simple example on wiki page. (4 points)
(Liuyang Yun, Sprint 3)

6A. Discuss and find solutions to improve task board management, task assignment and status updates (the team, Sprint 2-3)

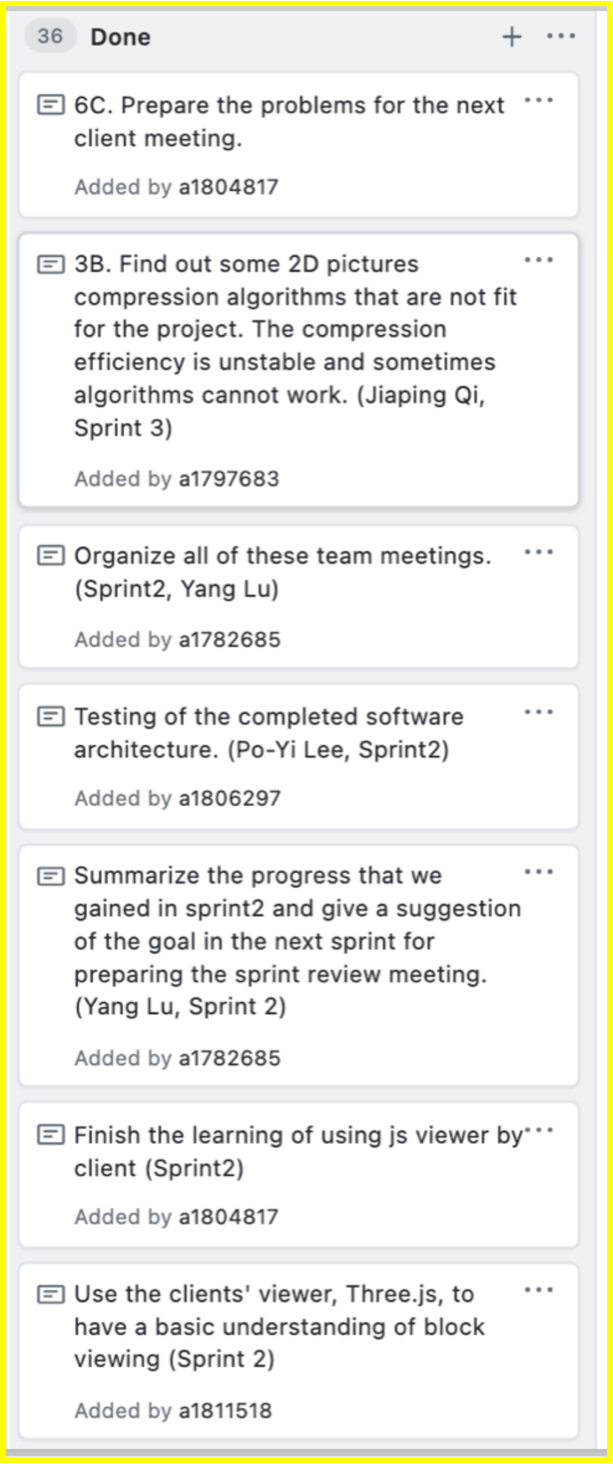
Added by a1811518

2B. Develop algorithm based on expanding blocks. (Sprint 2-3 - Hechen Wang, Xiaoman Li)

Added by a1804817

6B. Add traceability between the goals on the sprint backlog and the tasks (Karl Asenstorfer, Liuyang Yun, Sprint 3)

Added by a1811518



36 Done

6C. Prepare the problems for the next client meeting.

Added by a1804817

3B. Find out some 2D pictures compression algorithms that are not fit for the project. The compression efficiency is unstable and sometimes algorithms cannot work. (Jiaping Qi, Sprint 3)

Added by a1797683

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)

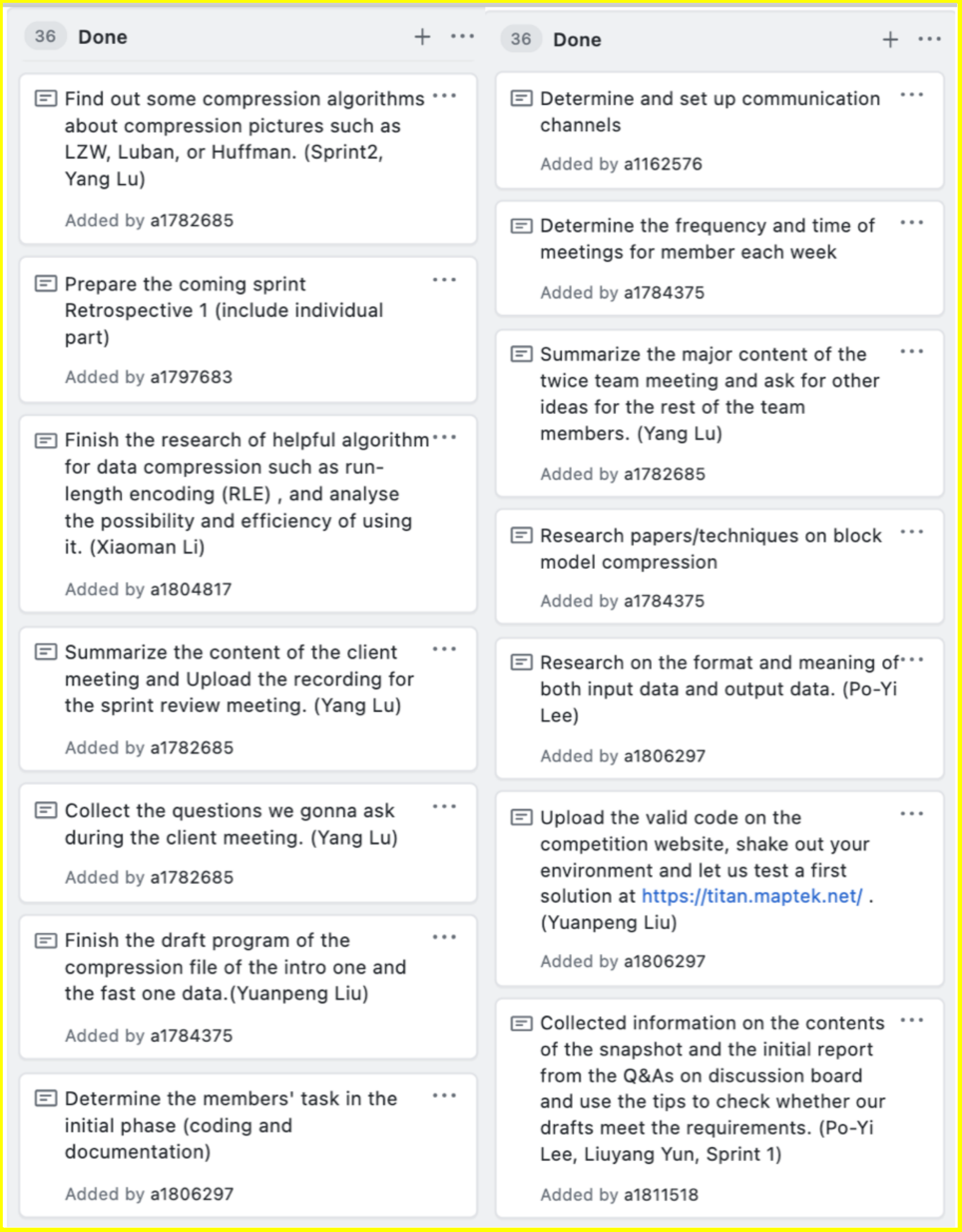
Added by a1782685

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



36 Done

Find out some compression algorithms about compression pictures such as LZW, Luban, or Huffman. (Sprint2, 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

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

36 Done

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)

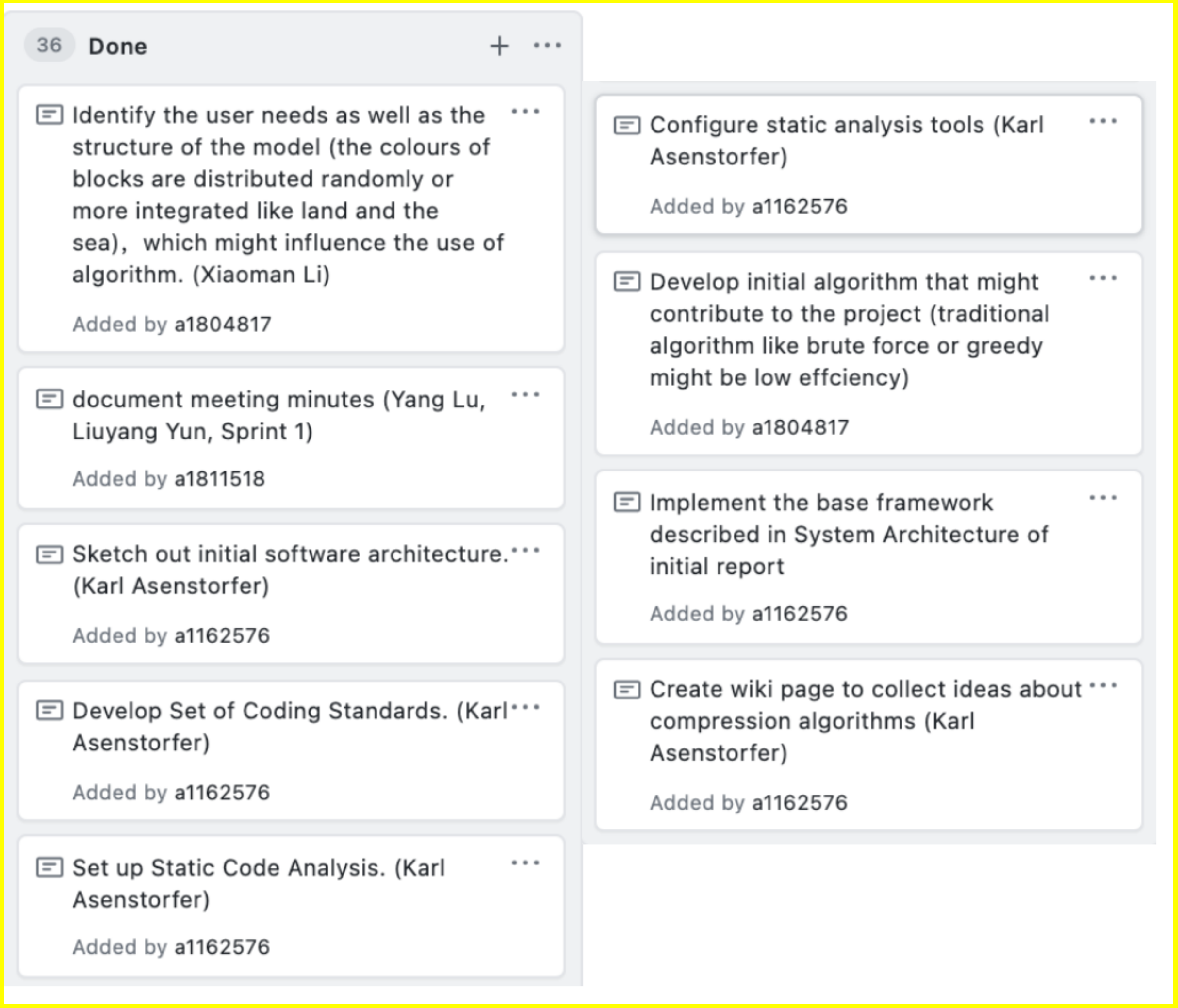
Added by a1806297

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

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



36 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

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

Added by a1804817

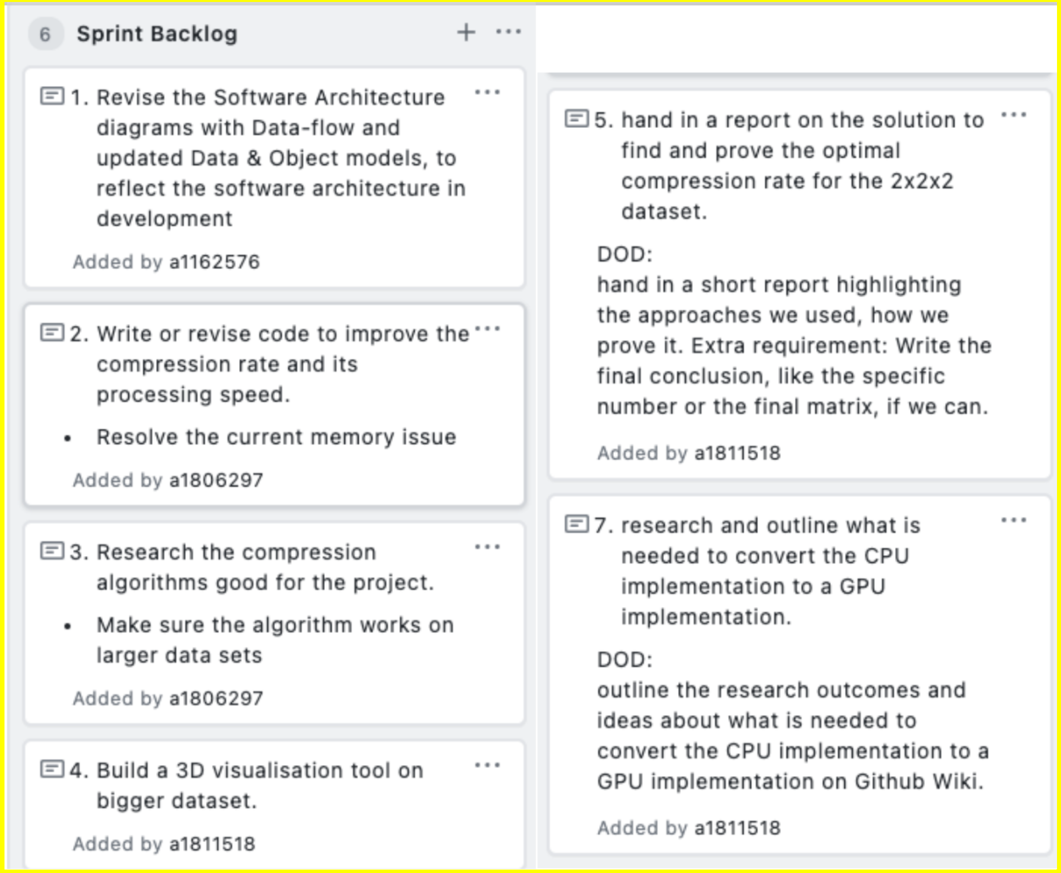
Implement the base framework described in System Architecture of initial report

Added by a1162576

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

Added by a1162576

Sprint Backlog and User Stories



6 Sprint Backlog

1. Revise the Software Architecture diagrams with Data-flow and updated Data & Object models, to reflect the software architecture in development

Added by a1162576

2. Write or revise code to improve the compression rate and its processing speed.

- Resolve the current memory issue

Added by a1806297

3. Research the compression algorithms good for the project.

- Make sure the algorithm works on larger data sets

Added by a1806297

4. Build a 3D visualisation tool on bigger dataset.

Added by a1811518

5. hand in a report on the solution to find and prove the optimal compression rate for the 2x2x2 dataset.

DOD:
hand in a short report highlighting the approaches we used, how we prove it. Extra requirement: Write the final conclusion, like the specific number or the final matrix, if we can.

Added by a1811518

7. research and outline what is needed to convert the CPU implementation to a GPU implementation.

DOD:
outline the research outcomes and ideas about what is needed to convert the CPU implementation to a GPU implementation on Github Wiki.

Added by a1811518

For this week, there are no major changes to the Sprint Backlog and User Stories. The main tasks of our team at this stage include continuously improving our compression algorithm to enhance its compression rate and speed, handle the memory issue when compressing the huge dataset, and studying for the new issues our client raised about changing the compression implementation from CPU to GPU.

According to the discussion in the last meeting, we have managed to divide team members into four groups. All of the team members are working for their goals.

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 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.
- For the task to improve the task board management: 1) add traceability between the sprint backlog and product backlog; 2) add the estimate of time and record the actual time on task board for comparison. 3) We add the deadline to the tasks.
- For the task to find the optimal compression algorithm: hand in a short report highlighting the approaches we used and how we prove it.
- For the GPU task: outline the research outcomes and ideas about what is needed to convert the CPU implementation to a GPU implementation on Github Wiki.
- We clarify the team members’ division in the meeting.
- We add the visualization block to clarify the compression result.
- We add GPU programmer tools in the Github wiki.

Summary of Changes:

In this week, we mainly focused on the four goals we determined in the last meeting. The improving work continues. The compression rate for the 2*2*2 of the new architecture we upload can be 79.63% now and the 4*4*4 one can be 93.74%. The visualization function could help us with observing the compression result. According to that, the algorithm team could analyze the compression figure and do improving work. The proof of the optimal compression is still working. We have uploaded the brief outline of the proof to the Github wiki. The GPU team has finished the data preparation stage and is ready to work on implementing the algorithm on GPU programmer platforms.

Declaration:

I attended:

- the sprint 4 review and sprint 5 planning meeting on 22th October, 2021 with the tutors, Aryaman Dhawan and Will Reid.
- the sprint 4 retrospective meeting on 24th October, 2021 with the other team members.

What went well in the sprint:

In the sprint 4, we assigned four main tasks to each team member including writing or revising code to improve the compression rate and its processing speed, and resolve the memory issue on the current algorithm, developing a compression algorithm that can handle the big_one dataset, handing in a report on the solution to find and prove the optimal compression rate for the 2x2x2 dataset and researching what is needed to convert the CPU implementation to a GPU implementation in the sprint review and planning meeting and had a group meeting so that every sub-team have had their own meeting to discuss the progress and the methods to figure out their own task. In addition, we submitted the new algorithm to Titan which successfully passed the first three datasets. We also updated the outline of how to get to the optimal compression algorithm for the 2x2x2 dataset on the GitHub wiki page and waited for the tutor’s feedback. Yuanpeng has been working on a 3D visualisation tool that will work on bigger datasets.

What could be improved:

In the last sprint, the sub-team had their own meeting to discuss the progress and the methods to figure out their own task. However, each sub-team did not regularly update the team on what they have been discussing and any issues they've had twice per week no matter what they actually contribute to and the scrum master did not actually trace each team's progress. In the next spring, we could have the meeting more frequently for the team and have our own workshop twice a week to be able to trace the team's progress and achieve our goal more efficiently.

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

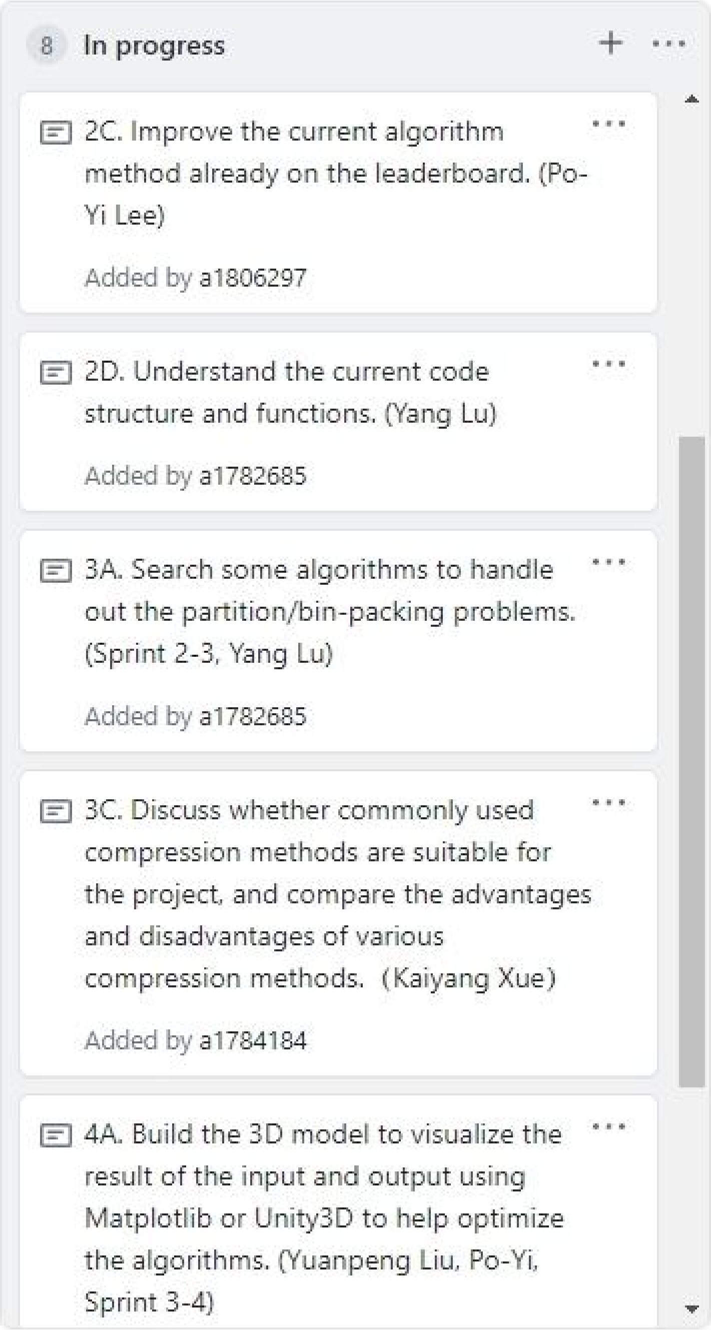
According to the sprint 4 retrospective meeting and the customer's requirement from the sprint 5 planning meeting, we will:

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.
5. Complete the final report and final presentation in the next sprint.

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.



8 In progress

2C. Improve the current algorithm method already on the leaderboard. (Po-Yi Lee)

Added by a1806297

2D. Understand the current code structure and functions. (Yang Lu)

Added by a1782685

3A. Search some algorithms to handle out the partition/bin-packing problems. (Sprint 2-3, Yang Lu)

Added by a1782685

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. (Yuanpeng Liu, Po-Yi, Sprint 3-4)

Figure 1: My in-progress tasks

In this sprint, I have been also 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, a new competition has been released and it includes the new dataset the_worldly_one_16777216_256x256x256, a ~16 million block model of the entire earth. Since the size of the blocks is the same as the size of the entire model(x_count, y_count, z_count, parent_x, parent_y and parent_z are 256), we can consider the entire model as a single parent block. Therefore we should revise the code and change the way of reading the parent block because there was a value error called "chunksize must be >= 1" when I initially implemented our algorithm with our multiprocessing method.