

MP6 Report

Xu Zihan 3220110781 zihan1.22@intl.zju.edu.cn

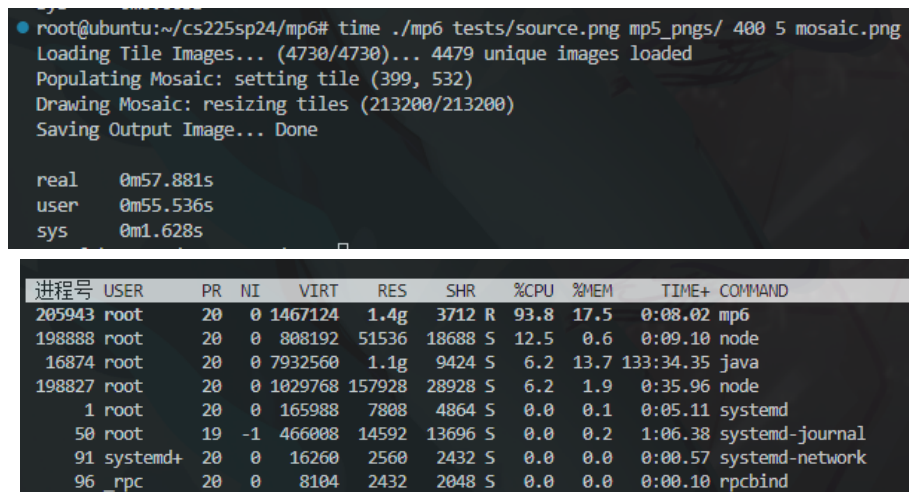


Figure 1 Performance of original MP5

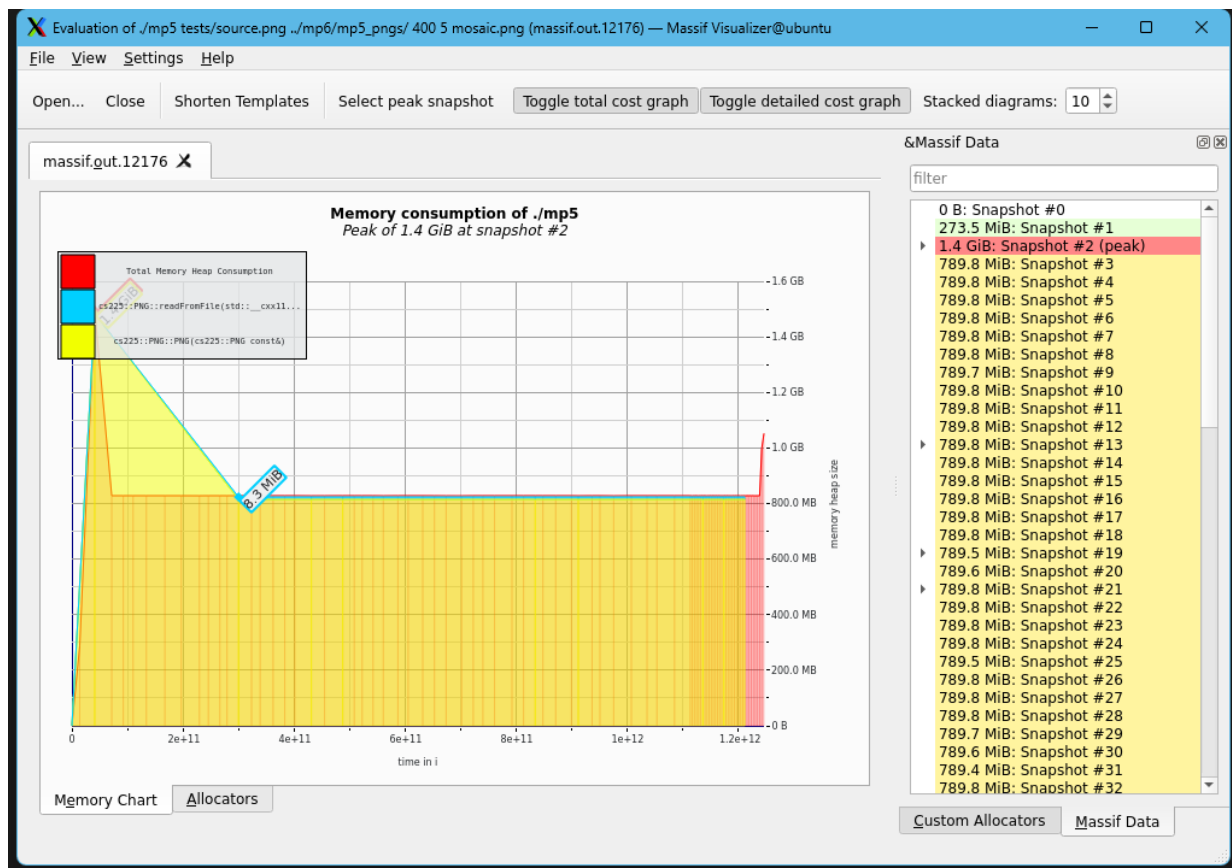


Figure 2 Memory consumption of original MP5

Reduce Space Complexity:

Analyzing memory usage using Valgrind's Massif tool, it can be seen that all png files are loaded into memory.

In the original version, `main.cpp::vector<TileImage>getTiles (string tileDir)` initialized each image as a `TileImage` and returned a vector of `TileImage`, which is the sum of all tiles, greatly wasting memory space.

1. My operation changes its return value to the `map<Point<3>, string>` of the file name and corresponding color value. As is well known, the size of the string is much smaller than that of the `tileImage` containing PNG.
2. Rewrite the `TileImage` class simultaneously and move the function and value that calculates `AverageColor` to

main.cpp. This can also reduce the complexity of mapTiles (Source Image const&TheSource, map<Point<3>, string>&TheTiles).

3. Once FNN finds nearby color blocks, using the TheTiles index, it can directly return the file name at the corresponding position on the canvas.
4. To this end, modify the function maptiles{cpp,h}::string* get_match_at_idx(const KDTree<3>&tree, map<Point<3>, string>&TheTiles, const Source Image&TheSource, int row, int col) and mosaiccanvas {cpp, h}:: const string&MosaicCanvas::getTile (int row, int column).
5. Finally, when rendering the canvas, load all tiles one by one and call the modified tiles. Theoretically, tiles will only occupy a maximum of $1 * w * h$ of memory space.

Memory: 1.4G -> 233.1M

Time: 57.881s -> 2m16s

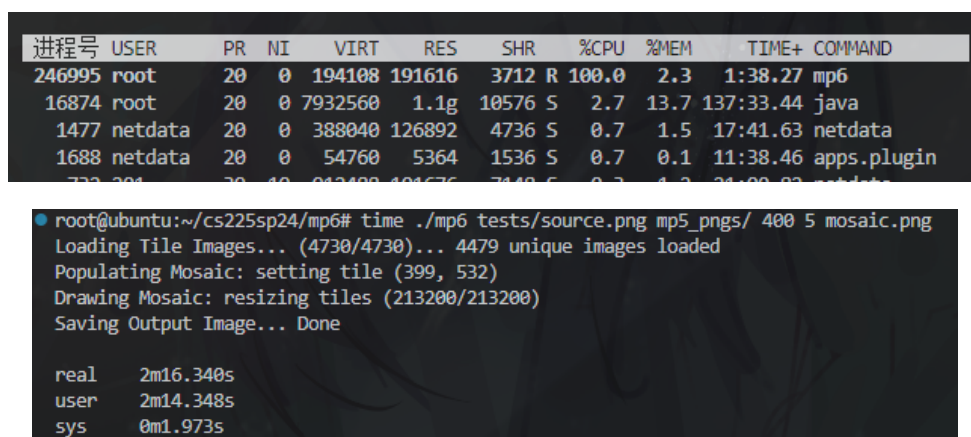


Figure 3 Performance of space-optimized MP5

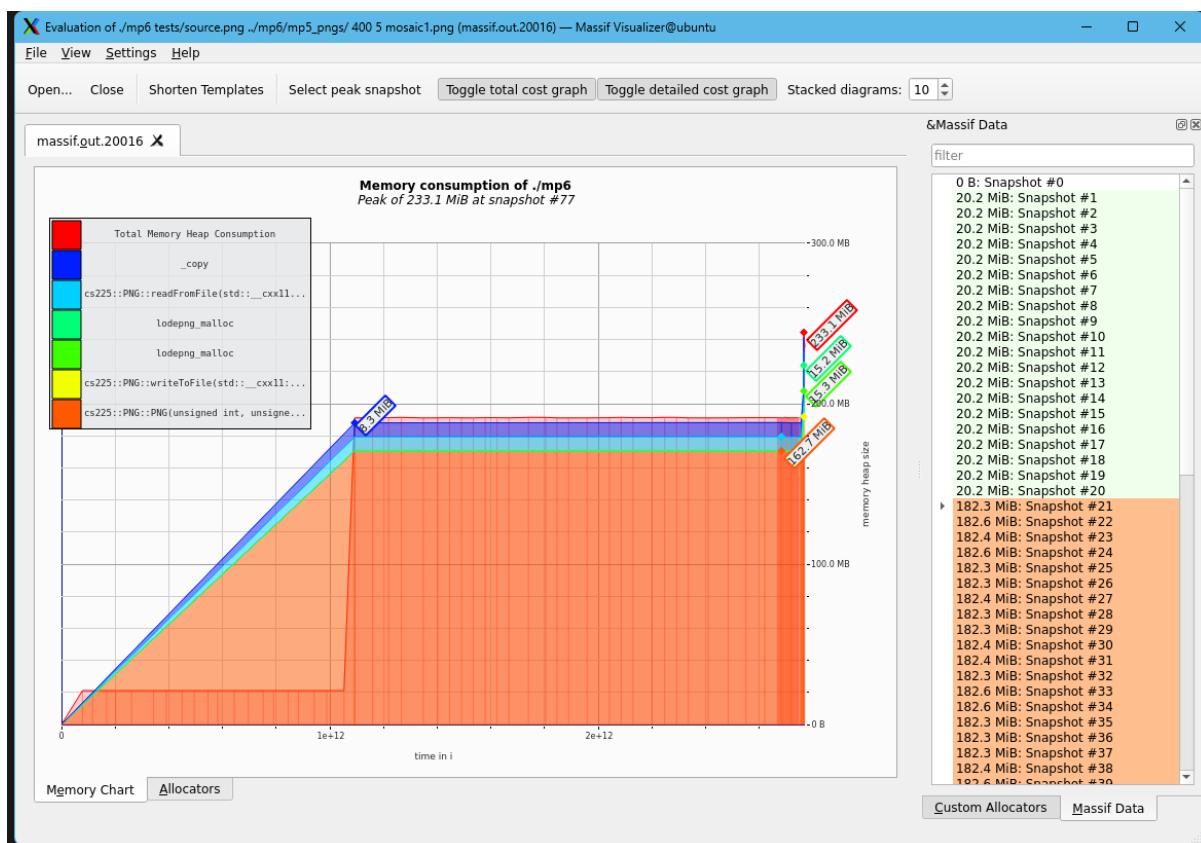


Figure 4 Memory consumption of space-optimized MP5

7



1



1

significantly faster than the original MP5, and there is no additional cost in memory.

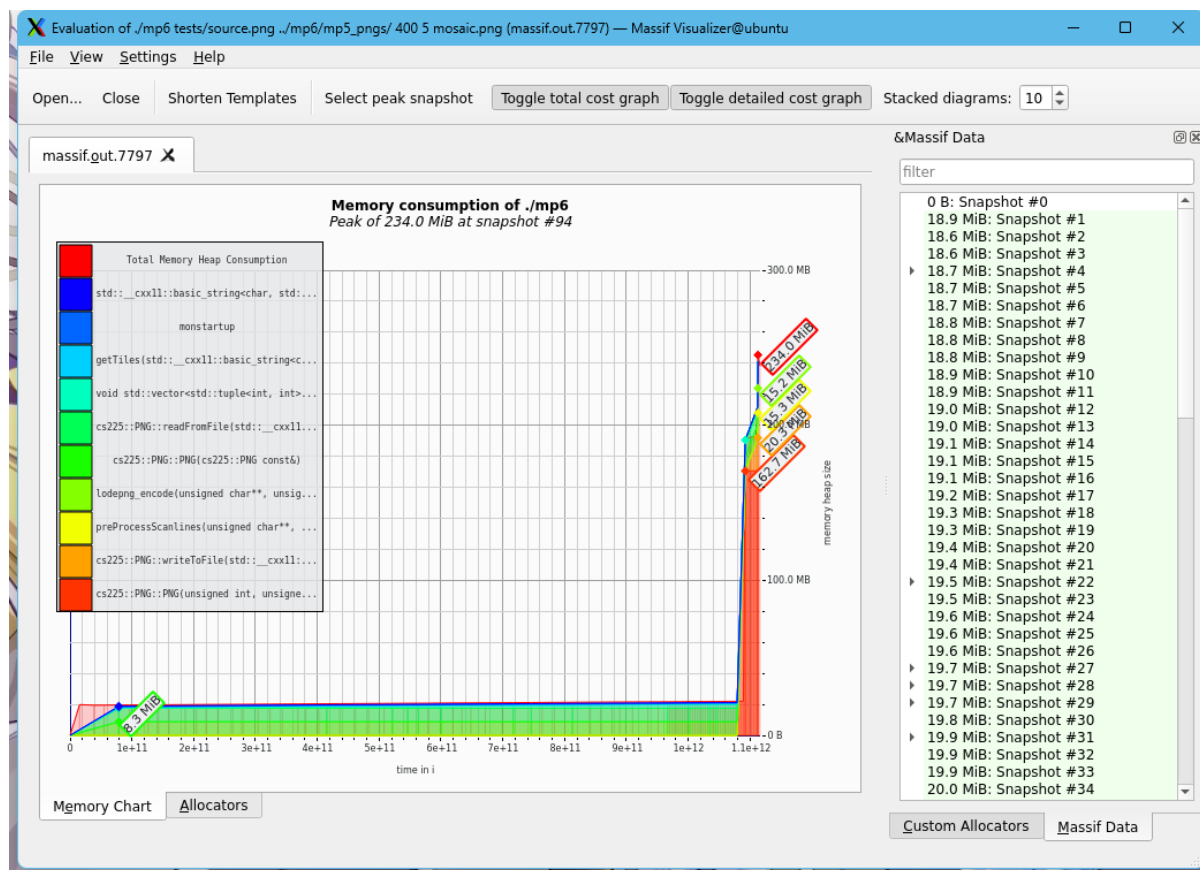


Figure 7 Memory consumption of final MP6

```
real    0m36.956s
user    0m34.339s
sys     0m2.469s
```

Final Performance:

Memory: 1.4G -> 233.1M -> 234.0M

Time: 57.881s -> 2m16s -> 36.956s