CS 194 **Fall 2015**

Baseline for Class Project

Harry He, Kada Situ, Brian SungYong Park, Conrad Yuan Shiao, Nikhil Unni

1 Project Description

The goal of this project is to optimize PIGZ, which is a parallel version of GZIP, a popular compression software in UNIX. PIGZ has already been parallelized using Pthreads in CPU and has achieved a linear speedup. However, with the rise of general purpose GPUs, we believe there is an opportunity to improve upon the PIGZ implementation. Our plan is to use CUDA technology to make use of the GPU to attempt to achieve further performance speedup during different stages of the compression algorithm, all the while trying to not significantly sacrifice the compression ratio. Because pigz uses zlib as its compression library, a large portion of our project will be making major changes in the zlib folder to use the GPU.

2 Code Description

Github Repository: `https://github.com/CS194-project/CS194-project`
Testing Data Website: `http://sun.aei.polsl.pl/~sdeor/index.php?page=silesia`

The "pigz" and "zlib" directories make up functional part of the project. The corpus directory stores the benchmark testing data. Other directories contain mainly reference code. Run "make" under root directory to build pigz and zlib. Inside pigz directory, "pigz" is parallelized version of pigz and "pignz" is the single threaded version of pigz.

3 Testing Data Description

We use Silesia Corpus as our testing data set. The corpus includes files with different formats in various sizes (ranging from 6MB to 51MB), which is great for testing compression efficiency. The actual data is not included in the repository. Run make corpus to download the testing data. Inside corpus folder, ".orig" files are the files downloaded. ".big" files are big data which are the combination of 10 copies of corresponding .orig files. "combined" file is the combination of all ".orig" files. Our benchmarking is based on all ".big" files and the "combined" file.

4 Steps to Run Benchmark Code

1. Clone the repository
   $ git clone https://github.com/CS194-project/CS194-project.git
   If you want a more verbose benchmarking output, change to "benchmark" branch. ($ git checkout benchmark)

2. Download Silesia corpus benchmark files
   $ make corpus

3. Build pigz (both serial/parallel) and zlib
   $ make

4. Run benchmark
   $ make benchmark

5 Testing Machine

- Machine IP: hive1.cs.berkeley.edu
- Memory size: 32GB
- CPU: Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz
- Number of cores: 4
- Number of hardware threads: 8
- L1 instruction cache: 32KB, 8-way, 64B cache line
- L1 data cache: 32KB, 8-way, 64B cache line
- L2 cache: 256KB, 8-way, 64B cache line
- L3 cache: 8MB, 16-way, 64B cache line
- Max memory bandwidth: 25.6 GB/s
- Supported SIMD instruction: SSE 4.1/4.2
- GPU: Geforce GT 740
- GPU Memory Size: 1GB
- Base clock: 993 MHz
- Number of CUDA cores: 384
- GPU Memory Bandwidth: 80 GB/s
- CUDA Compute Capability: 3.0
- CUDA library version: 7.5

6 Benchmark

Running the serial version on our corpus we get:

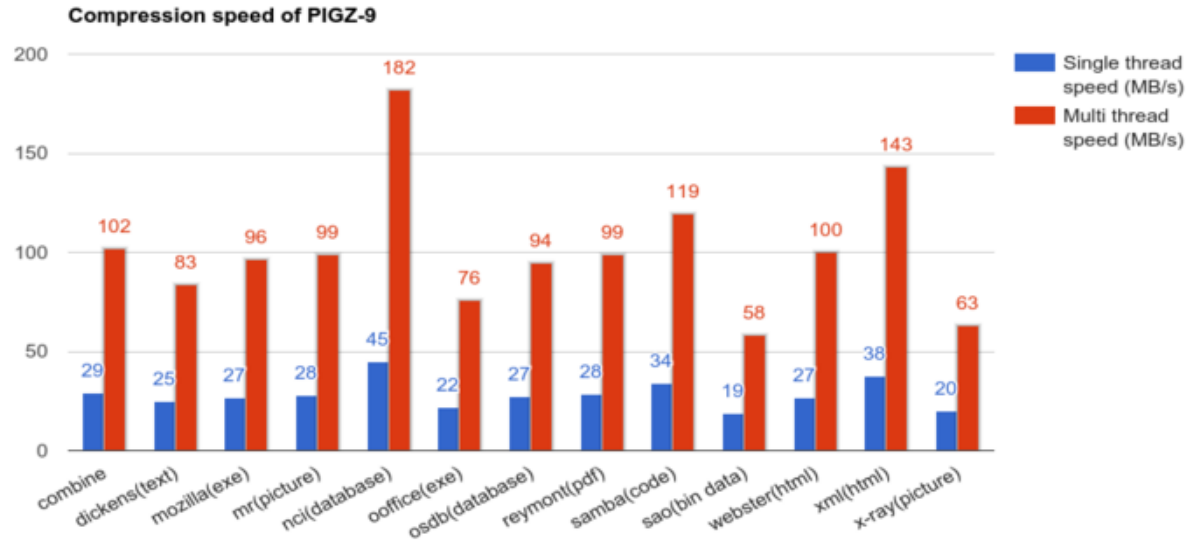
Corpus	Total Time (s)	IO Time (s)	Speed (MB/s)	Input Size (byte)	Output Size (byte)	Compression Ratio
combine	7.09	0.048	28.701	211938890	78351541	2.705
dickens	0.393	0.003	24.881	10192446	4725440	2.157
mozilla	1.823	0.012	26.967	51220480	20783147	2.465
mr	0.343	0.002	27.942	9970564	3886965	2.565
nci	0.707	0.006	45.627	33553445	4500144	7.456
ooffice	0.271	0.002	21.806	6152192	3343141	1.84
osdb	0.362	0.002	26.711	10085684	4138273	2.437
reymont	0.226	0.001	28.175	6627202	2386691	2.777
samba	0.623	0.004	33.326	21606400	6216311	3.476
sao	0.374	0.002	18.604	7251944	5670905	1.279
webster	1.434	0.009	27.739	41458703	15334578	2.704
xml	0.145	0.001	35.511	5345280	1148727	4.653
x-ray	0.417	0.003	19.5	8474240	6205547	1.366

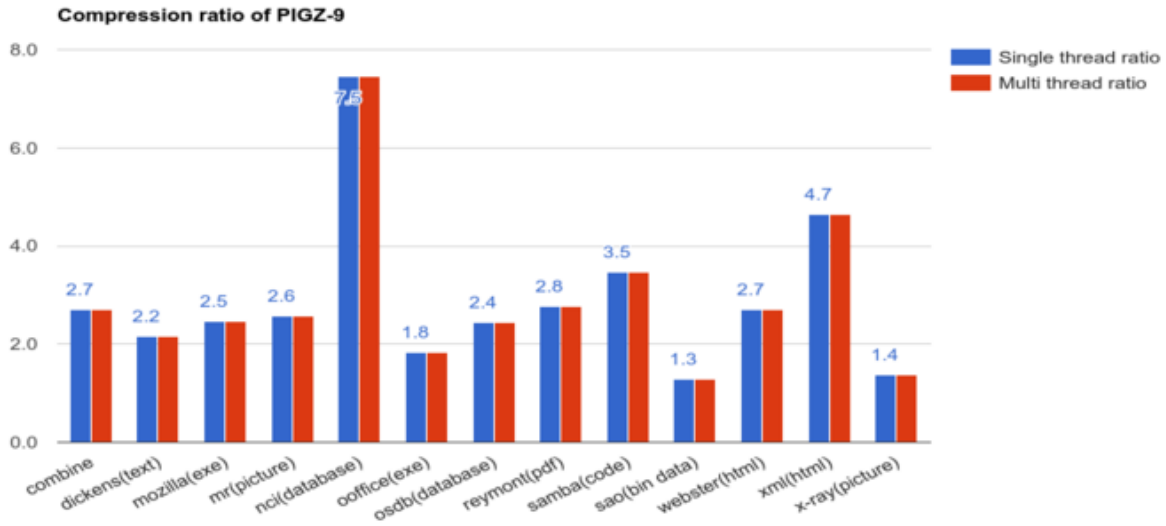
1. Serial Baseline

And running the full parallel version of pigz on the corpus we get:

Corpus	Total Time (s)	IO Time (s)	Speed (MB/s)	Input Size (byte)	Output Size (byte)	Compression Ratio
combine	5.287	4.568	281.313	211938890	78351541	2.705
dickens	0.249	0.196	180.796	10192446	4725440	2.157
mozilla	1.380	1.162	223.733	51220480	20783147	2.465
mr	0.227	0.186	231.616	9970564	3886965	2.565
nci	0.818	0.768	632.326	33553445	4500144	7.456
oofice	0.159	0.121	156.694	6152192	3343141	1.840
osdb	0.173	0.113	161.243	10085684	4138273	2.437
reymont	0.166	0.130	179.688	6627202	2386691	2.777
samba	0.449	0.367	249.627	21606400	6216311	3.476
sao	0.198	0.137	112.528	7251944	5670905	1.279
webster	1.121	0.967	257.036	41458703	15334578	2.704
xml	0.098	0.073	197.823	5345280	1148727	4.653
x-ray	0.174	0.101	110.974	8474240	6205547	1.366

2. Full Parallel Baseline





7 Appendix : PIGZ Profiling

We used the Linux perf tool to determine the time consumption of each functions.

Command under pigz folder:

$ sudo perf record ./pigz -k -9 corpus/combined

$ sudo perf report