Thesis Portfolio

Advocating for Compression Aware Algorithms

(Technical Report)

An Investigation of the Relationship between Moore's Law and the Data Congestion Problem

(STS Research Paper)

An Undergraduate Thesis

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Arman Lokhandwala
Spring, 2019

Department of Computer Science

Table of Contents

Sociotechnical Synthesis

This document contains the presentations of my technical and STS research projects. My technical research was about compression aware algorithms and my STS research project was an investigation about the relationship between Moore's Law and the data congestion problem. Compression aware algorithms is a worthwhile field researching further, as Moore's Law is slowing down because of physical limitations at the nanoscopic level, as well as numerous other reasons. Therefore, finding software based solutions to advance the efficiency and power of current transistors is a way to prolong Moore's Law, which has played a pivotal role in creating industrial and societal norms. This reasoning leads into my STS research which explores the relationship between Moore's Law (and it's slowing), and how the data congestion problem has been affected by Moore's Law, as well as future implications.

Speaking more of my technical project, by creating more efficient algorithms that operate on compressed data, we can target a very specific purpose of transistors which will help prolong Moore's Law. The algorithms that were used for this research were the bipartite assignment and the topological sorting algorithms, both of which were chosen to represent the broader algorithm categories of breadth first search (BFS) and depth first search (DFS), respectively. Various experiments were conducted to observe the speed and space tradeoff of graphs when varying aspects such as algorithms used (standard vs. compression aware), graph specification (uncompressed vs. compressed), graph size and density, as well as compression ratio. The report concludes claiming that since compression aware and standard algorithms are asymptotically the same, there is no difference in always using the compression aware versions. Compression aware algorithms operate much quicker on compressed data sets than the time needed to decompress data and run standard algorithms on it, therefore compression aware algorithms should be adopted as the new standard.

In my STS research, I explore the data congestion problem and investigate the effect of Moore’s Law on society and how its slowing will be perceived in the future. I claim that Moore’s Law has helped shape society into believing that technology should be discarded and upgraded about every two years. This notion implies that Moore’s Law has significantly influenced society’s current sociotechnical imaginary, an STS framework developed by Jasanoff and Kim. Also, I discuss the reoccurring data congestion problem and how Moore’s Law has always provided an answer. Data congestion becomes more of an issue when consider that Moore’s Law is inevitably slowing down. I conclude by arguing that the only possible way to counter upcoming data congestion problems, considering that Moore’s Law is inevitably slowing down, is to alter the sociotechnical imaginary into one that does not depend on Moore’s Law.

The value of my work is, in a sense, dependent on the significance and importance of my contributions to further supporting the research described in my reports. My experimental setup and initial test results should prove useful in supporting the claim that standard algorithm implementations should be replaced with compression aware implementations. Although my work is not sufficient enough to advocate the change itself, it serves as the first step in the right direction. For future steps, I would recommend continuing experimentation with different representations of data, for instance by using graphs of different structures (hub and spoke graphs).

As far as my STS work is evaluated, I believe that my claims from my investigation are valid and worth considering further. My STS project was fruitful in that I was able to learn about a new STS framework and apply that to our society in an effective manner. If this research was furthered, I would recommend analyzing the patterns and behaviors of society in relation to their treatment towards technology through other STS frameworks.

Advocating for Compression Aware Algorithms

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Arman Lokhandwala
Spring, 2019

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature ____ Date _____________
Arman Lokhandwala

Approved ____ Date _____________
Nathan Brunelle, Department of Computer Science

Advocating for Compression Aware Algorithms

Abstract

By creating more efficient algorithms that operate on compressed data, we can attempt to prolong Moore's Law. The algorithms that were used for this research were the bipartite assignment and the topological sorting algorithms, both of which were chosen to represent the broader algorithm categories of breadth first search (BFS) and depth first search (DFS), respectively. Various experiments were conducted to observe the speed and space tradeoff of graphs when varying aspects such as algorithms used (standard vs. compression aware), graph specification (uncompressed vs. compressed), graph size and density, as well as compression ratio. The report concludes by discussing the benefits of operating on compressed data, which ultimately leads to the claim that since compression aware and standard algorithms are asymptotically the same, there is no difference in always using the compression aware variants. (The content from this paragraph also appeared in the sociotechnical synthesis section).

Introduction

Microelectronics as a field is undergoing rapid advances. Surely, the speed at which the field advances must plateau eventually, especially when considering that microelectronics is inherently limited by the physical size of transistors that make up the field. There is no guarantee that transistors will continue to exponentially decrease in size, as Moore's Law suggests, because of the limitations of our own discoveries and scientific understanding. Furthermore, Andreas Stiller, a tech journalist who has covered processor hardware and technology for more than 30 years, commented that, "silicon-based lattice structures simply won't support electron flow"

(Feldman, 2018). This ‘leveling’ of improvement and growth is unavoidable yet incredibly worrisome. Moore’s Law is the observation of the rapidly increasing number of transistors in the world, a segment used to build high(er) performing hardware components. This ‘law’ is nothing more than an observation made in 1965 by Intel’s co-founder, Gordon E. Moore, pointing out that the number of transistors placed in a chip had doubled every year since its invention. 10 years later though, he noted that the rate had slowed down by doubling not annually, but rather approximately every two years. (Tardi, 2019). Therefore, Moore’s Law predicts that the rate of transistor production and growth will continue at the same rate, doubling in quantity approximately every two years.

Due to the competitive nature of manufacturers, programmers, and investors alike, Moore’s Law quickly became the standard without people ever realizing how. For example, large tech companies rely on better hardware through more advanced and denser chips in order to progress and develop new technologies. Nonetheless, with Moore’s Law setting the expectations, the general population has experienced this trend of doubling in power about every two years whether they realize it or not. Additionally, this observation becomes more and more of a ‘law’ as more industries adopt and deploy it in different ways. This competitive nature of companies helped spark the construction of Moore’s Law into the standard it is today. (The content that appeared in the last two paragraph also appears in the later section: An Investigation of the Relationship between Moore’s Law and the Data Congestion Problem).

Motivation

The motivation to research compression aware algorithms stems from the fact that Moore’s Law is slowing down. Moore’s Law is the observation that the number of transistors that can fit on a microchip doubles approximately every two years. This exponential growth prediction can only continue for so long, as there are physical limitations and barriers that are proving challenging to overcome. Due to the scientific progress we have made in the field of nanotechnology, some transistors “are smaller than a virus”. These chips are perfectly made in order to help move electricity along the circuits faster. Unfortunately though, the transistors reach extreme temperatures which make it impossible to create smaller circuits, because cooling the transistors takes more energy than what passes through the transistors (Tardi, 2019). This implies that Moore’s Law is coming to an end, as producing smaller and denser chips is becoming physically impossible. Therefore, in order to extend Moore’s Law, we must approach Moore’s Law by focusing on its added software benefits. By creating more efficient algorithms that operate on compressed data, we can target a very specific purpose of transistors which will help prolong Moore’s Law.

Analogously, one can intuitively appreciate the benefits of operating on compressed data, rather than raw data, by considering the benefits of shipping uncooked rice, rather than cooked rice. Transporting uncooked rice is much more efficient than transporting cooked rice, as one cup of uncooked rice is equivalent to about 5 cups of cooked rice. There are two main advantages of this approach: (1) by shipping uncooked rice, we could use a smaller truck to do so, and (2) it takes less time to load and unload the truck because we can fit more grains in one package (Brunelle, 2017).

Similarly, operating on compressed data provides the same benefits. Since the data is in compressed form, the memory requirements of our algorithms would be reduced, hence a smaller truck. And, our algorithms would have a higher effect on each part of the data it actually observes or manipulates, hence, more grains in each package and takes less time to move them all around (Brunelle, 2017). Approaching algorithm design with this analogy in mind can help counteract the slowing down of Moore's Law.

Background on Algorithms Used

The algorithms used in this research were the bipartite assignment and the topological sorting algorithm. These two algorithms were chosen to be used in experiments conducted because they represent the broader graph traversal algorithms which are: breadth first search (BFS) and depth first search (DFS), respectively. The graph data structure is a common and useful programming method to store data. Unlike other data structures like a queue or a stack, trees are non-linear and a hierarchical based structure. Examples of data stored in trees or tree-like data structures include any hierarchical based data, such as a family tree (TK, 2017). The component that holds the data, whether it be a number, name, or another data structure, is a node. Nodes are connected via links called edges (TK, 2017).

Graph traversal algorithms visit every node stored in a graph. Considering a variety of data can be stored using the graph data structure, it makes sense that BFS and DFS are an interesting classification of algorithms to research. Essentially, BFS algorithms start at the top of a graph and explore all of its neighbors before going down any specific path. DFS algorithms start at the top node (or any node specified in the algorithm) and go down an entire path before exploring other neighboring paths. Slightly altering the standard BFS and DFS algorithms allows the implementation of various different algorithms, such as sorting, searching, calculating the shortest path, etc. For these reasons, the bipartite assignment and topological sort were chosen to be studied, as finding evidence that apply to these algorithms can help scale upwards to similar algorithms of the same categorization.

Background on Compression Scheme

The algorithms needed a consistent and intuitive compression scheme in order to run efficiently on compressed. The chosen scheme limited each compressed node to no more than 2 adjacent nodes, and hence 2 outgoing edges. Also, the compression scheme introduced 'dictionary nodes' to represent multiple other nodes. For instance, in an uncompressed graph, node A may have adjacent nodes *B, C,* and *D*. However, when compressed, a dictionary node called *dict_BCD* may be created to represent and have nodes *B, C,* and *D* be adjacent to it. This new dictionary node would then be added to node A's adjacency list while removing nodes *B, C,* and *D* from the list. It's important to note that a dictionary node may have other dictionary nodes in its adjacency list. A scheme like this one reduces the number of items in the nodes' adjacency list, thus lowering the number of edges needing to be traversed. This compression scheme allows for a consistent and fairly intuitive traversal when altering standard algorithms to perform on compressed graphs. Additionally, this compression scheme proves useful because once compressed, the data set is still represented as a graph. Each dictionary node appears as a normal node in the graph with no new data and zero weight, and instead has edges to each of the nodes it represents. This implies that "many classical algorithms require only minor adjustment to operate on the compressed data" (Brunelle, 2017, p 34).

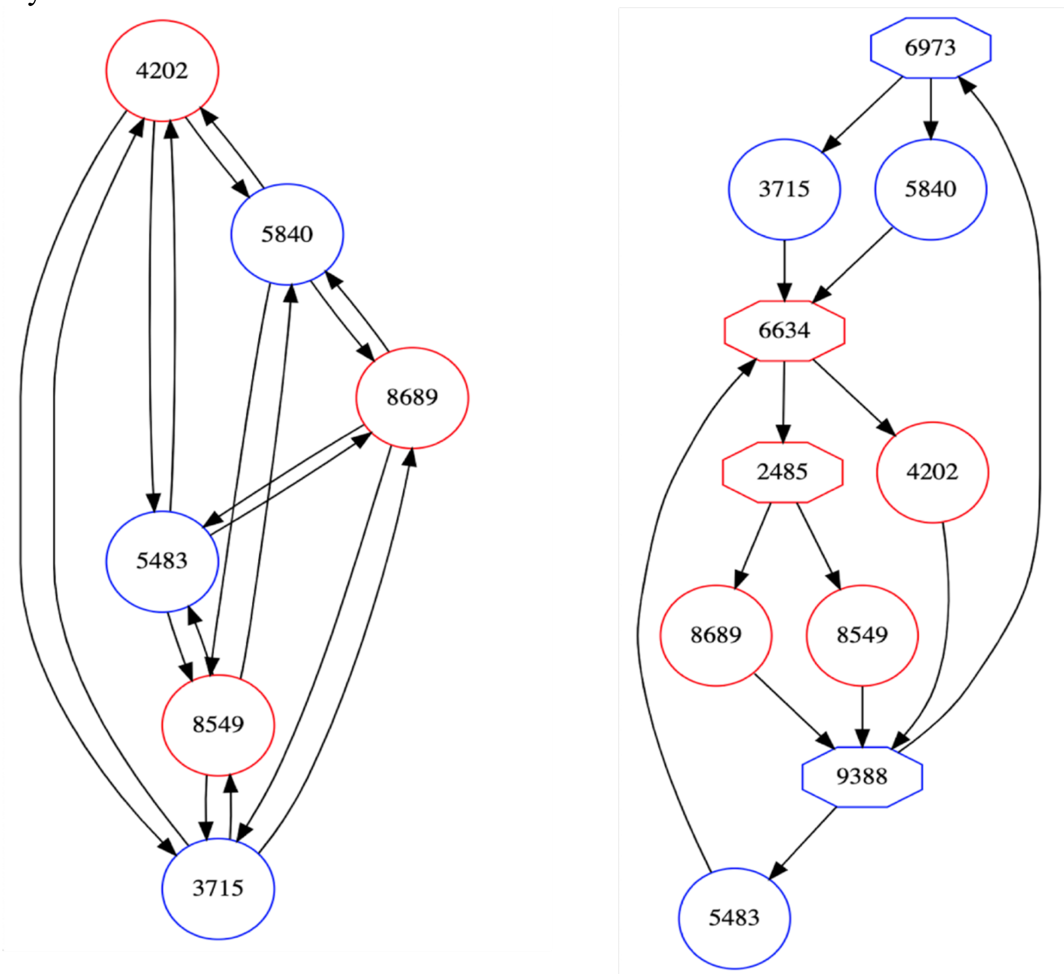
Explanation of Compression Aware Algorithms

Bipartite Matching Algorithm

Given the context of how the graphs are compressed, the compression aware algorithms are more understandable. The normal bipartite algorithm works by assigning opposite colors to neighboring nodes. Meaning, if node A is colored red, then all of A's adjacent nodes must be a different color, say blue. At the end of the bipartite matching, if any two adjacent nodes are the same color, the graph is not a bipartite graph. The compression aware algorithm behaves similarly. However, it is designed to identify dictionary nodes as ‘dummy nodes’ whose assigned color should be the same as the dictionary node’s adjacent nodes. This is because the dictionary node acts as a replacement node in order to reduce the number of total traversals. In reference to the provided example above, if node A was colored red, normally neighboring nodes *B, C,* and *D* would be colored oppositely as blue. Following the compression scheme would leave node *dict_BCD* as a filler node, which would then appropriately be colored blue, as would the actual nodes *B, C,* and *D*. Below are images of actual graphs, with the figure 1.a being an uncompressed perfect bipartite graph with 6 nodes total, and figure 1.b the compressed version of the graph seen in 1.a. The numbers in the nodes represent a unique ID of each node, which corresponds to the data it holds.

When the compression aware bipartite matching algorithm is run on a graph of size *C*, it finishes in time *O(C)* (Brunelle, 2017). This is due to the fact that although this algorithm is a modification of breadth first search, “there is no asymptotic time penalty for running this algorithm over breadth first search, giving the time complexity to be $O(|V_c| + |E_c|) = O(C)$.”, where V_c is the number of vertices or nodes, and E_c is the number of edges traversed. (Brunelle, 2017, p. 47).

Figure 1:A sample graph (a) and its representation as a compressed graph (b). The octagonal nodes are dictionary keys and their outgoing lines are the edges which they replace in the adjacency lists.



(a) **(b)**

The compressed graph in figure 1.b visibly has fewer edges, intuitively implying that its traversal may be more efficient. This is more apparent when scaling up to a larger number of nodes. The dictionary nodes are identified as the nodes that are octagonal in shape.

Topological Sort Algorithm

The topological sort algorithm operates on compressed graphs similarly to how it would uncompressed graphs. Topological sort is classified as a depth first search algorithm and works by visiting every node in the graph. It starts with a given vertex, typically the root, then visits all of the neighbors of that node. This pattern continues until all nodes are visited. The order in which the nodes are visited are returned in a stack and that is considered the sorted graph. The compression aware algorithm invokes the standard one, however, before returning the final sorted order, the algorithm filters out the dictionary nodes.

When the compression aware topological sort is run on a graph of size C, it finishes in time $O(C)$ (Brunelle, 2017). The compressed graph size, C, can be represented via the simple equation: $|V_c| + |E_c| = C$, where $|V_c|$ is the number of nodes and $|E_c|$ is the number of edges in the graph. A normal topological sort requires time $O(|VC| + |EC|) = O(C)$. Filtering out the dictionary nodes takes only $O(|V_c|)$ time. This results in the final time complexity of the compression aware algorithm to be $O(C)$ (Brunelle, 2017).

Software Developed

Throughout the course of the project, much of the research was done in implementing software to test out conceptual hypotheses. One of the larger endeavors I took on was implementing a working compression aware bipartite matching function. Though the project had already had a version of the function, it produced incorrect results that were inconsistent with our expectations. Debugging the algorithm proved more difficult than starting over. This new algorithm works similar to how it was intended to, as described in *Super Scalable Algorithms*, the dissertation that this research is based on (Brunelle, 2017). Essentially, it labels the first node as one color, red, and goes on to do a breadth first search, labeling the nodes after checking for certain conditions, such as the color of the previous node, and whether that it was a dictionary node or not. After the nodes are all labeled, a check is done to assure the bipartite matching was completed correctly.

Also, for this algorithm to work, I had to alter the way the bipartite node class was structured. Originally, the bipartite node had a list of attributes, which I simply changed into various field that would be useful in different algorithms, such as node color, vertex, predecessor, and an adjacency list. This change made the bipartite matching algorithms much easier to implement and read.

Lastly, both the normal bipartite and compression aware bipartite matching algorithms were altered to perform a full breadth first search on the entire graph, so that the algorithm always runs in the worst case scenario. This change was made because exiting the function too quickly when a graph is not a bipartite would skew the time it would take to perform a BFS on the entire graph. So, for the purpose of supporting the research, the algorithms were changed to perform a full BFS on all graphs.

Additional software developed include error checking in various methods such as topological sort. A test running file was also made which essentially ran multiple instances of tests and stored data in files to be parsed and analyzed. A data parser was also created to quickly parse and analyze a large number of data files, stored in spreadsheets. The rest of the software implemented throughout this research project are test files, which will be discussed in a later section.

Experimental Setup

Description of Experiments

The experimental setup was generally the same for many different trials. Essentially, we wanted to test the runtime of the two algorithms on various classifications of graphs. As shown in the table below, all of the cells with a check mark (✔), describe varied independent variables where the runtime was the dependent variable.

	Standard Algorithms	Compression Aware Algorithm
Uncompressed	✔	✔
Compressed	❌	✔
Decompressed	✔	✔

Figure 2: A table showing the independent variables that were changed to test for runtime and space complexity of the two algorithms (bipartite matching and topological sort). Each cell with a check mark (✔) indicates the conditions in which the algorithms were tested.

In addition to runtime, space complexity was also an important factor to observe throughout the experiments. Another dependent variable was the size of the graph, measured by the total number of nodes and edges. As mentioned briefly above, the expectations were that compressed graphs have a higher number of nodes and a lower number of edges than the uncompressed version of the same graph. Another set of experiments were based on comparing the compression ratio of the graphs to the appropriate speedup in terms of algorithmic run time. Randomly generated compressed graphs were produced with varying sizes (number of nodes), then decompressed. Both the bipartite matching and topological sort algorithms were run on both graphs, where the compression aware variants were run on the compressed graphs and the standard algorithms were run on the decompressed graphs.

Test Software Developed

A large part of the test software was setting up graphs. Since different density graphs were being tested, various methods were needed to initialize said graphs. Thus, I created functions to randomly generate highly compressible graphs, poorly compressible graphs, perfect bipartite graphs, and compressed graphs. Each graph can be ‘Re-paired’, a function which enables them to be compressed or decompressed. The compression scheme’s name is prefixed with “Re” because it stands for the word “recursive”, which explains that the compression works by recursively pairing nodes (Larsson & Moffat, 1999).

Additionally, many tests functions, which ran bipartite matching and topological sort, were made to run on the different classifications of graphs described above. At first, these test functions timed how long it took for the algorithms to run, then printed out the results. But later, the functions were optimized by appending the times (and test descriptions) to a string and returning the string. This optimization allowed another file, the test runner, to run multiple instances of the tests and properly store the results for analysis.

Results and Analysis

Since there were numerous experiments conducted, many observations and test results were recorded. Firstly, as hypothesized, the compression aware algorithms were shown to operate on uncompressed graphs at a slightly slower rate than the normal implementation. This is likely due to the compression aware algorithms being a modification of the standard implementations, except with more checks to handle dictionary nodes. This observation existed regardless of the algorithm and density of the graphs. For instance, figures 3.1 and 3.2 show this tendency in poorly compressible graphs when topological sort is run on it, as well as on highly compressible graphs which ran the bipartite matching algorithm, respectively.

Figure 3.1: A graph showing the results of topological sort run on poorly compressible graphs. The blue line is the normal algorithm and the orange line is the compression aware version.

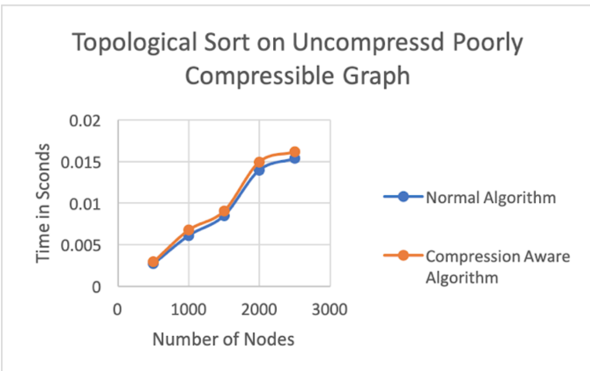
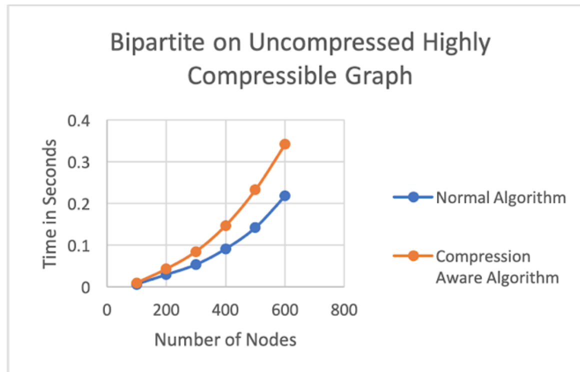
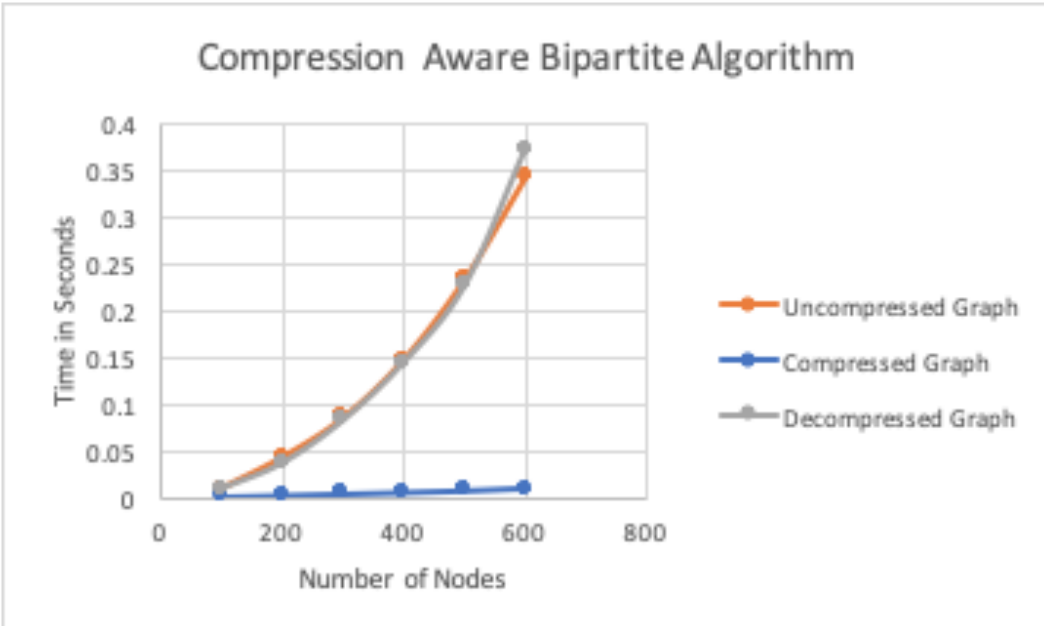


Figure 3.2: A graph showing the results of bipartite matching on highly compressible graphs. The blue line is the normal algorithm and the orange line is the compression aware version.



The results demonstrated from the first graph indicate that our algorithm is equally efficient to the traditional algorithm. The second graph implies that using the standard algorithm provides no asymptotic advantage in runtime when compared to the compression aware version. Interestingly, the results support the idea that compression aware algorithms are more efficient to run on compressed data than operating on uncompressed data. These results are depicted in figure 4 below.

Figure 4: Graph showing the results of the compression aware bipartite matching algorithm run on different classifications of graphs. The orange line is the uncompressed graph. The blue line is the compressed graph. The gray line is the decompressed graph.



From this graph, we discover that the compression aware algorithm sees dramatic speedups when compressing the graph when compared to the original uncompressed version. We also demonstrate that the decompressed graph is similar to the uncompressed graph to further prove that any gain in runtime is achieved by exploiting the compression aware algorithm on the compressed graph.

However, compressing the data is an expensive operation, taking much longer than the any algorithm would take individually. However, given the benefits of using compression aware algorithms, it would be more advantageous to standardize the use of compression aware algorithms. This choice is sensible, because although standard, i.e. non-compression aware, algorithms are slightly faster than compression aware algorithms, the difference is not asymptotically significant. Also, since more and more data sets are being transferred and stored compressed, the reliance of compression aware algorithms are anticipated to increase.

Repeatedly decompressing graphs and running standard algorithms is a longer process which can be skipped over by directly operating on compressed data.

Conclusion and Future Steps

In conclusion, since compression aware and standard algorithms are asymptotically the same, there is no difference in always using the compression aware versions. Compression aware algorithms operate much quicker on compressed data sets than the time needed to decompress data and run standard algorithms on it, therefore compression aware algorithms should be adopted as the new standard. Future works in the field should include further experimenting to support the findings of the research described in this report. Additionally, more research should be conducted to control for compression ratios in relation of algorithmic speedup. Using different representations of data, such as by using graphs of different structures is also worth considering. Not only verifying other types of graph structures, but adapting them to more applications of BFS and DFS algorithms would prove useful in further supporting the research.

Additionally, an area of ongoing development included trying to control for compression ratio to see how that would affect the speedup that results when compressing the graph. However, this specific research did not come to fruition as controlling the compression ratio is not as intuitive as expected. We built random graphs by assigning nodes to adjacency lists as sampled from a non-uniform distribution; however, this approach did not result in enough variance from the independent variable in order to draw conclusions about how different compression ratios cause different speedups from uncompressed graphs using standard algorithms. In the future, creating graphs where nodes have intentionally assigned edges will help discover which conditions contribute to controlling the compression ratio. This approach seems promising and could lead to valuable findings and discussions in the future.

References

Brunelle, N. (2017). *Super-Scalable Algorithms*(Unpublished doctoral dissertation). University of Virginia. doi:http://www.cs.virginia.edu/~njb2b/Brunelle_phdDissertation_UVACS_2017.pdf

Larsson, N. J., & Moffat, A. (1999). *Dictionary-Based Compression*(Data Compression Conference, pp. 296-305). IEEE Computer Society.

Tardi, C. (2019, April 20). Moore's Law Explained. Retrieved from http://www.investopedia.com/terms/m/mooreslaw.asp

TK. (2017, November 05). Everything you need to know about tree data structures. Retrieved from https://medium.freecodecamp.org/all-you-need-to-know-about-tree-data-structures-bceacb85490c

An Investigation of the Relationship between Moore's Law and the Data Congestion Problem

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Arman Lokhandwala
Spring, 2019

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature ____ Date _____________
Arman Lokhandwala

Approved ____ Date _____________
Tolu Odumosu, Department of Engineering and Society

An Investigation of the Relationship between Moore's Law and the Data Congestion Problem

Introduction

Microelectronics as a field is undergoing rapid advances. For example, in very few years we have progressed from individual transistors to integrated circuits. Surely, the speed at which the field advances must plateau eventually, especially when considering that microelectronics is inherently limited by the physical size of transistors that make up the field. There is no guarantee that transistors will continue to exponentially decrease in size, as Moore's Law suggests, because of the limitations of our own discoveries and scientific understanding. Furthermore, Andreas Stiller, a tech journalist who has covered processor hardware and technology for more than 30 years, commented that, "silicon-based lattice structures simply won't support electron flow" (Feldman, 2018). This 'leveling' of improvement and growth is unavoidable yet incredibly worrisome. Moore's Law is the observation of the rapidly increasing number of transistors in the world, a segment used to build high(er) performing hardware components. This 'law' is nothing more than an observation made in 1965 by Intel's co-founder, Gordon E. Moore, pointing out that the number of transistors placed in a chip had doubled every year since its invention. 10 years later though, he noted that the rate had slowed down by doubling not annually, but rather approximately every two years. (Tardi, 2019). Therefore, Moore's Law predicts that the rate of transistor production and growth will continue at the same rate, doubling in quantity approximately every two years.

Due to the aforementioned limitations of current research and progress made using silicon-based transistors, Moore's Law inevitably slows down. However, it is imperative that our techniques for data processing and computing improves in order to keep up with the prediction that processing power doubles every 18 months (Simonite, 2016). This is the only alternative to postpone the end of Moore's Law, as increasing the computational power in the transistors being produced today reduces the need of having a higher quantity of transistors in chips. With this approach, higher quality chips trump higher quantity. Researchers can achieve this higher quality by improving the method in which large amounts of data is processed. Furthermore, the most necessary area of development related to efficient data processing is compression-aware algorithms, which refers to the calculations or analysis of data that is compressed. For example, when sending large spreadsheets online, they are often compressed into a .zip file in order to save space. Researching ways to operate on the compressed file rather than having to decompress the file first would save time, space, and potentially computing power. Optimized compression-aware algorithms would be capable of efficiently using computing resources while scaling upwards, which would otherwise be used decompressing the same data. These computing resources would be best optimized because the new techniques mentioned above would operate directly on compressed data sets whereas traditionally, algorithms are run on uncompressed data - a method which takes too much time and too much space. Implementing algorithms to run on compressed data directly allows for a comparison with the aim of seeing a speed up in runtime when compared to running similar algorithms after decompressing the data first. For my technical project I will be researching compression aware algorithms.

Although data processing is often times overlooked and not considered when evaluating societal norms, efficient data processing plays an enormous role on present day societal and political norms. For example, a popular topic currently is the idea of autonomous vehicles and whether self-driving cars will ever become a reality. The data processing required for autonomous driving to occur from a technical standpoint, as well as be endorsed into society is not limited to live data processing, but also the analysis of human behavior, trends, and history.

This example directly relates to the idea of congestion, too. The idea of congestion is not new, as it exists all around us, manifesting in areas such as the streets through automobiles on the roads, or people flooding the sidewalks, and of course, the ever-increasing bandwidth and server traffic. Many people don't consider data congestion a problem, let alone know it exists! Data congestion proves to be an area of concern for those in the fields of technology, business, and medicine.

Naturally, as more and more information is transferred digitally, regardless of the means, data traffic will continue to increase, resulting in the lack of uniformity and consistency. Such traffic can cause serious losses in the above-mentioned fields. The slowing of Moore's Law introduces a larger problem, as advanced hardware components typically keep data congestion under control. As Moore's Law slows, superior methods of data processing, for instance with compression-aware algorithms, would help reduce the data congestion problem by improving consistency and uniformity in digitally transferred data. For my STS research project, I will explore the data congestion problem and investigate the effect of Moore's Law on society and how its slowing will be perceived in the future. I will study Moore's Law through the sociotechnical framework, developed by Sheila Jasanoff, in order to better understand how society has behaved and developed in correspondence to it. The concept of sociotechnical imaginaries offers a framework to analyze events by examining the relationship between scientific and technological policy and development to [societal] culture. Studying Moore's Law with this perspective in mind can help uncover the relationship between Moore's Law and society – an aspect that undoubtedly has affected the data congestion problem and will continue to do so in the future.

Finally, by gathering data regarding the relationship between Moore's Law and data congestion, I can better understand society's role on the data congestion problem. If we can establish a pattern and make predictions about when, where, and how congested data transmission will be, scientists and industries can anticipate and plan a solution to the issue before it becomes unmanageable.

More about Moore

As mentioned above, Gordon Moore discovered his prediction by simply observing the trend of transistors doubling in quantity in the timeframe of about every two years. Moore's Law is worth analyzing considering the results Moore's Law has proven thus far, and the implications yet to come. Due to the competitive nature of manufacturers, programmers, and investors alike, Moore's Law quickly became the standard without people ever realizing how. For example, large tech companies rely on better hardware through more advanced and denser chips in order to progress and develop new technologies. A consumer example would be how customers expect new hardware - such as a new Apple iPhone every year. The public expects it to be newer, better, faster, and more capable. Nonetheless, with Moore's Law setting the expectations, the general population has experienced this trend of doubling in power about every two years whether they realize it or not. Additionally, this observation becomes more and more of a 'law' as more industries adopt and deploy it in different ways. For instance, in relation to the Apple example above, after Apple started deploying newer technologies on a regular basis, so did other competitors in the industry. This competitive nature of companies helped spark the construction of Moore's Law into the standard it is today.

The remarkable feat of setting not only an industrial standard but also a universal standard is incredible, especially when thinking about the inevitable end of the recently ongoing trend. Technologists have not yet thought about how to face the public about why technological advancement has slowed, but they should, as the public will soon demand answers and have a right to know the truth. An approach to informing the public on the matter is to try to prolong Moore's Law and in the meantime, educate people about the imminent future where slowed progress is a reality.

The uprise of Moore's Law as an industry standard has been discussed, but the downfall and slowdown of the law in itself is much more interesting. Moore's Law boils down to the idea that the number of transistors doubles nearly every two years, and respectively, so does computing power. However, as the number of transistors increases, so does the cost of making them, as the precision required for such high-density chips becomes increasingly difficult the smaller they get. Due to the manner in which industries think about Moore's Law, they create products with the intention of replacing them in a short time span. Consequently, industries competed with each other and created lower quality products and society has adopted the norm of replacing technology every few years. This dependency contributes to the severity of the downfall of Moore's Law. However, due to the scientific progress we have made in the field of nanotechnology, some transistors "are smaller than a virus". These chips are perfectly made in order to help move electricity along the circuits faster. Unfortunately though, the transistors reach extreme temperatures which make it impossible to create smaller circuits, because cooling the transistors takes more energy than what passes through the transistors (Tardi, 2019).

In addition to the physical challenges in the upkeep of Moore's Law, economic factors also play a large role in whether the trend continues or not. Producing smaller chips requires an extensive amount of research and development, not including the sheer manpower and resources required for experimentation with different technologies, actual factory/plant/development centers, machinery, tools, and of course the actual materials used (and conversely wasted). As the inventor of Moore's Law himself, Gordon Moore, comments to CNET via the Scientific American article, "Intel makes these investments, which are in the [billions of dollars], because they expect to reap way more [billions of dollars] in profits in following years. But if there is doubt that those profits will arrive, and possibly if they just doubt they can come up with the necessary silicon improvements, they may not want to make the investment at all." (End of Moore's Law: It's Not Just about Physics). This implies that Gordon Moore himself cannot guarantee the benefits of investing resources into further developing transistors, as more limitations are being discovered and profit margins are shrinking. Also, if Intel were to stop investing in creating smaller chips, then other competitors would undoubtedly follow suit and halt development. As the article summarizes it, the end of Moore's Law is near, as physics and economics are both presenting unavoidable and seemingly ever-lasting barriers to any further development.

Often times, a limitation such as the one presented by the conclusion of Moore's Law does not pose much of a threat to the common person in their daily life, but rather hinders their social and work lives as technology struggles to keep up with the ever-changing world. However, as mentioned above, nearly all industries and areas of life are modernizing and hence, are becoming reliant on smaller and more powerful transistors. This implies that society is either going to have live with what we have and handle the current technological limitations, or scientists will need to come up with a technological fix to take over or prolong Moore's Law. This becomes even more urgent when considering the data congestion problem - a commonly overlooked issue that is growing in significance. Its importance will be explained in greater detail in the next section, but in essence is the loss in quality and/or data that is sent over networks. This is a problem because of the idealistic, technologically advanced and technology-reliant future that society so eagerly wishes to obtain.

Data Congestion Problem

Data congestion, in reference to networks, refers to the state when "a node or link carries so much data that it may deteriorate network service quality, resulting in queuing delay, frame or data packet loss, and the blocking of new connections." (What Is Congestion?). It's clear that the more data that is sent through a specific medium, the lower the fidelity of the data. In simple terms, data congestion is like traffic on the highway; if multiple cars try to merge into the same lane at once, a traffic jam will occur, because more cars are trying to fit in than the lane allows (Rivenes, 2016). Analogously with data congestion, radio waves, internet, or cellular networks are the lanes and the actual data [being transmitted] are the cars trying to merge in.

This congestion happens for multiple reasons, with the main causes being low bandwidth or the border gateway protocol (BGP). Bandwidth can be simplified to be understood as "the size of the pipe" in which data can travel through (What Is Internet Congestion?, 2017). Surely then, the bandwidth affects how much data can be transferred at a time, for if the density and quantity of the data exceeds the capacity that can be transported, there will be delays. In references to the other main cause, essentially, border gateway protocol is an algorithm which controls internet traffic and how it is handled. BGP works by sending data over the 'shortest logical path'. Though this sounds like an efficient approach to handling traffic, it isn't the best approach for every situation, as this approach sacrifices many other factors that prove to be significant when calculating the optimal handling strategy. For example, the greedy choice of shortest logical path does not consider the fastest path, the lowest path cost, the path with the most bandwidth available, or even how much data is being sent. The simplest solution to the reduced performance of network latency would be to choose the most optimal path, however all network transactions are unique and of different sizes. Computing the optimal path isn't always feasible as the increased (and variable) computation on each request causes more congestion and wait time on other aspects of the system. Hence, the greedy choice of shortest logical distance is efficient when amortized over a long period of time (Rivenes, 2016 June). Again though, this choice in handling internet traffic sacrifices other optimal solutions that may reduce congestion. Since the BGP sends all network traffic over the shortest path, it does not consider how busy the path is. If the shortest path is already overloaded with data, it will result in slower speeds when sending data over network, which is by definition data congestion.

Slower connections and computations are not only a nuisance, but can even be dangerous in certain situations. With the world becoming increasingly dependent on technology and internet connectivity, the consequences of slow connections become more dire. For example, many people believe that the reason for developing newer cell signals (moving from 1G to 4G LTE) was the make them faster. The real reason for introducing 2G, the successor of 1G, was because originally the cell spectrum was divided into different channels that allowed people to make phone calls. In order to avoid interference of calls being placed at the same time, the phone network required large gaps in the spectrum, as each channel could only support one user at a time, limiting capacity and scalability of the network. Hence, "mobile communication demands soon outgrew the capabilities of 1G leading to the improvement of technology and existence of 2G" (Mountstephens, 2017). Furthermore, Glenn Fleishman, a senior contributor and tech journalist, states in his article *The state of 4G: it's all about congestion, not speed*, "speed isn't actually the main reason why every carrier in the developed world is trying to extend the life of

3G networks by speeding them up…. It's all about congestion and capacity, not speed" (Fleishman, 2010). In his article Fleishman details the reasoning behind the need to expand cellular networks as he argues that increased capacity and decreased congestion is the main driving force being developments from 1G to 2G and onwards. This brief example shows some of the efforts that have already been made in order to combat the data congestion problem. The problem becomes even more troublesome when taking into account the different mediums and opportunities in which the problem can present itself. Data congestion occurs in many forms such as packet loss, interference in data transferred via radio waves, poor throughput leading to data loss, lower bandwidth, and prolonged process queuing. Nonetheless, as Moore's Law slows down, researchers must come up with an even more intricate solution to deal with the inevitable data congestion problem.

Reimagining the Future

Sociotechnical imaginaries, a concept first introduced in Sheila Jasanoff and Sang-Hyun Kim's work in 2009, was used [in their work] to explain science and technology developments in relation to political institutions. This framework applies to Moore's Law and data congestion in that both developments coincide and intertwine with society. The basis of sociotechnical imaginaries is that they "encode not only visions of what is attainable through science and technology", but also how life should, or should not, be lived; with this understanding, sociotechnical imaginaries "express society's shared understandings of good and evil" (Jasanoff and Kim, 2015, p. 4). Thus, sociotechnical imaginaries are "collectively imagined forms of social life and social order reflected in the design and fulfillment of nation-specific scientific and/or technological projects" (Jasanoff and Kim, 2015, p. 4). Additionally, sociotechnical imaginaries are "collective, durable, and can be performed, yet they can also be temporally situated and culturally particular" (Jasanoff and Kim, 2015, p. 19). The imaginary that forms through the narrative that is the recent history of the evolution of technology observed through Moore's Law is worth analyzing with respect to how social, political, and ethical commitments are formed into national, if not at times universal, trajectories of technoscientific developments and how said commitments contribute to innovations in various other fields, such as in relation to the data congestion problem.

Borrowing from the loose definition provided above, it is only just that the idea of Moore's Law and all of the social and political factors revolving around it forms a sociotechnical imaginary. Moore's Law influenced society by establishing an industry standard for all technology chip manufacturers. Gordon Moore helped form the industry's sociotechnical imaginary by pointing out, and thus setting the expectation of the number of transistors found on a chip doubles approximately every two years. The imaginary was more so supported because other chip manufacturing industries also started referring to the prediction as a production goal, originally as competition. Thus, Moore's Law is deployed by industries and investors alike as an excuse to develop more powerful hardware, pushing the limits of researchers, producers, as well as consumers. All of these aspects combine to create a rich and intertwined sociotechnical imaginary, where each respective party (researchers, investors, producers, consumers) have a reliance on one another. Since sociotechnical imaginaries are forms of social life and order reflected in the design and fulfillment of scientific and technical projects, the acceptance and deployment of Moore's Law in industries provides an interesting narrative for the societies that benefit from the observation. It is important to note though, in its early stages, Moore's Law originally predicted the time frame to be yearly instead of every two years. The fact that Moore had to correct himself implies that he knew that his prediction would significantly impact the industry and thus the social life and social order of American societies.

This standard helped develop the sociotechnical imaginary of many Americans, including the consumers, producers, as well as various industries. The prediction of doubling computational power approximately every 2 years presents many implications for consumers. For instance, people assumed and expected technology to double in speed, since the hardware was twice as capable. The public equates faster computation to mean better quality and thus more desirability. The public has shifted focus on faster processing rather than storage capacity, battery life, camera quality, or other features. A large part of the sociotechnical imaginary that has formed over the years is credited to advertising from various technology manufacturers. For example, Apple boasts about their newer phones being superior in processing power, which in turn enables their newer products to be packed with more features and capabilities. Many other corporations advertise in similar fashions for all technologies. This perception of technological advancement that has been forced on to the public has caused people to believe that faster processors equals progress. This sociotechnical imaginary stuck to people's mindset and so the industry had no choice but to meet their expectations with more and more technical products. Over the years, the cost of developing faster chips has doubled for the producers, being invested in researchers and equipment, whereas consumers were able to enjoy decreasing prices for more powerful computational components. As Rock's Law, or Moore's Second Law outlines: the cost for semiconductor chip fabrication doubles every four years (Spacey, 2016). This trend also helped form the sociotechnical imaginary that Americans accepted to be true. Speaking to how different industries were affected and a part of the sociotechnical imaginary, industries behaved similarly to consumers, except at a much larger scale.

The trends of Moore's Law and Rock's Law did not establish themselves and become a standard without other influences; the existing sociotechnical imaginary helped consolidate these trends into standards, providing pressures on the relevant parties to produce products. Similarly, the societal expectations helped drive the technological innovations that demanded the law hold. For instance, hardware manufacturers, tech companies, and industries alike relied on anticipated hardware upgrades. These pressures only added on to the sociotechnical imaginary that coerced producers to keep meeting the expectations. Everyone has become overly dependent on Moore's Law, which is why its role in forming the imaginaries for the public is so critical to analyze. This lens reveals two hidden behaviors of the public, one focusing on the consumer and the other on researchers. Firstly, the public assumes that new technologies will be developed regularly - this assumption needs to stop. The dependency caused by such an assumption causes people to disrespect their technologies by discarding and mistreating it, assuming a replacement will soon be released. Secondly, researchers need to find a way to deal with the data congestion problem without new technologies being produced because of Moore's Law. This directly implies that researchers will have to focus more on methodology or algorithm changes rather than the traditional component and hardware upgrades.

Sociotechnical imaginaries also serve as a method of warning against risks that may surface if innovation is pushed too hard or fast. As discussed before, any exponential growth must eventually slow due to physical and economical impediments. The sociotechnical imaginaries that have long been established are being challenged by the slowing of Moore's Law. A new sociotechnical imaginary must be formed in order to accurately convey the severity of the issue, as older social life and order have simultaneously driven innovations for solving the recurring data congestion problem. Now, scientists must find a way to not only to alter the preconceived notion of exponential computational growth, but also to deal with data congestion. The problem of data congestion has been suppressed due to the sociotechnical imaginary that has been accepted by society in the past. For instance, briefly mentioned above, the development of new (Apple) cell phones were brought about based on the assumption that Moore's Law would hold. Currently, even Apple has slowed its hardware announcements, focusing more on software updates rather than physical hardware updates. One of the implications of Moore's Law is that when more dense and powerful chips are made, the price for consumers also decreases, hence making older technologies (that were considered the benchmark 2 years prior) even more affordable. Thus, with more powerful technologies becoming more and more affordable, it follows suit that technology usage and reliance (by the common person) would go up drastically.

The rigorous increase in technology usage inherently leads to decreased bandwidth for all mediums involved, and consequently increased data congestion. Therefore, the new sociotechnical imaginary needs to be preventative rather than a hopeful future, since Moore's Law is ending and data congestion is undoubtedly on the rise. Whatever design and goals of new technological projects are, they should imply to society that the age of rapid hardware developments is halting. Only deliberate design choices and innovative technical products can help establish a new sociotechnical imaginary where society does not rely on expected hardware updates. Consumers have benefitted from new technologies being developed while Moore's Law was still being followed in the past. However, technologists must now anticipate new problems related to the growing issue of limited technological power, since the researchers are accustomed to Moore's Law providing more powerful technology.

Implications and Future Steps

The relationship that has been established between the two phenomena, Moore's Law and the data congestion problem, supports that both have played a pivotal role in the history of technological progress. This is most clearly noted when observing the creation of new technologies, in terms of higher computational power and higher bandwidth. This relationship thus implies that the two are correlated, if not causative, where Moore's Law causes the postponing of data congestion. Due to the slowing of Moore's Law, data congestion will inversely begin to rise in how much of a problem it poses to the public.

The implications to be learned from this association is two-fold: one is that Moore's Law needs to be extended so that newer technologies can be produced as a result of more powerful hardware. The other option is to alter the sociotechnical imaginary that has been deemed true for so long so that people change their mindset on how the technology field progresses, taking into account the slowing of Moore's Law. Despite research being done for the former option, there is no guarantee that any permanent fix will be found. Even still, Moore's Law cannot be extended any longer, as any exponential growth curve must eventually reach a barrier, thus flattening and slowing. Regardless, the fix would most likely be temporary. The only other option then is to construct a new sociotechnical imaginary in which Moore's Law is not a dependency, nor a standard. As mentioned briefly before, this new sociotechnical imaginary serves as a preventative one, not a hopeful future vision.

Data congestion is inevitable, especially as more and more people around the world become dependent on technology in their day to day lives. Similarly to how new drivers are learning to have patience on the road in order to handle traffic, internet users should learn to accept and understand that data congestion is a rising problem. As a brief aside, another option to consider is the thought of removing this dependency altogether. This implies that instead of a technical solution, which is what has been followed for all prior cases of data congestion, a 'social engineering' approach is taken. This idea presents itself in Alvin Weinberg's piece, *Can technology replace social engineering*, in which he eventually concludes that social engineering cannot be replaced by technological fixes (Weinberg, 1967). This directly means that, no matter how long we extend Moore's Law, the only way to sustain a peaceful public is to change their expectations and understanding of what technology in the society should look like.

Conclusion

Congestion exists in all aspects of life: traffic on highways, too many people in a busy store, and of course too many people using up the cell lines at the same time. With the increasing dependency on technology, congestion of data trying to be transmitted is bound to occur, and at alarmingly increasing rates. Not only can data congestion be cumbersome when trying to browse the internet or check emails, but it can be dangerous in a variety of industries! Thanks to Moore's Law, which is the prediction that processing power doubles every 18 months, the data congestion problem has been kept at bay (Simonite, 2016).

The more prevalent data congestion became, more superior technology was invented in order to better combat the problem. However, Moore's Law cannot be sustained forever and is slowing down. A long running trend such as Moore's Law introduces a sociotechnical imaginary, an idea invented by Sheila Jasanoff and Sang-Hyun Kim. This sociotechnical imaginary, or social life, has been adopted by society and established due to Moore's Law providing regular technological advancements.

Society is evolving, and consequently so are available technologies, resources available [to develop said technologies], as well as the societal norms. Despite a seemingly abundant number of directions to take as society, moving forward without Moore's Law consistently providing more powerful technology at a more affordable cost provides obvious challenges. The most prominent and difficult to overcome is the fact that, aside from a minor inconvenience due the changes in newer technologies available on the market (such as new phones at scheduled intervals), society hasn't the slightest idea of the setback that researchers and industries alike must face. In order to prevent largely negative reactions from the public, the only plausible option is to change the sociotechnical imaginary moving forward. Surely, communities and eventually societies will understand the importance of Moore's Law and all of the advancements and inventions that have come about due to the prediction that became the standard.

References

End of Moore's Law: It's not just about physics. (n.d.). Retrieved from http://www.scientificamerican.com/article/end-of-moores-law-its-not-just-about-physics/.

Feldman, M. (2018, June 6). As Moore's Law Winds Down, Chipmakers Consider the Path Forward. Retrieved from http://www.top500.org/news/as-moores-law-winds-down-chipmakers-consider-the-path-forward/.

Fleishman, G. (2010, March 29). The state of 4G: It's all about congestion, not speed. Retrieved from https://arstechnica.com/tech-policy/2010/03/faster-mobile-broadband-driven-by-congestion-not-speed/

Jasanoff, S., & Kim, S. (2009). Containing the Atom: Sociotechnical Imaginaries and Nuclear Power in the United States and South Korea. *Minerva,47*(2), 119-146. doi:10.1007/s11024-009-9124-4

Jasanoff, S., & Kim, S. (2015). *Dreamscapes of modernity: Sociotechnical imaginaries and the fabrication of power*. Chicago: The University of Chicago Press.

Jasanoff, S., Kim, S. H., & Sperling, S. (2007). Sociotechnical imaginaries and science and technology policy: a cross-national comparison. *NSF Research Project, Harvard University*

Mountstephens, S. (2017, October 27). The Evolution of Mobile Internet. How 1G Became 4G LTE. Retrieved from https://blog.oneringnetworks.com/the-evolution-of-mobile-internet

Reuters, E. A. (2015, May 21). 'Data traffic jams' are now a potential problems. Retrieved from http://www.businessinsider.com/r-not-so-fast-connected-cars-could-cause-data-traffic-jams-2015-5

Rivenes, L. (2016, June 2). 10 causes of Network congestion you should know about. Retrieved from https://datapath.io/resources/blog/10-causes-of-network-congestion-you-should-know-about/

Rivenes, L. (2016, July 19). What is internet congestion? Learn about overloaded routes. Retrieved from https://datapath.io/resources/blog/what-is-network-congestion/

Simonite, T. (2019). *Intel is putting the brakes on Moore's Law*. MIT Technology Review. Retrieved from https://www.technologyreview.com/s/601102/intel-puts-the-brakes-on-moores-law/.

Spacey, J. (2016, May 17). Moore's Law vs Rock's Law. Retrieved from https://simplicable.com/new/moores-law-vs-rocks-law

Tardi, C. (2019, April 20). Moore's Law Explained. Retrieved from http://www.investopedia.com/terms/m/mooreslaw.asp

Weinberg, A. M. (1967). Can Technology Replace Social Engineering? *American Behavioral Scientist,10*(9), 7-7. doi:10.1177/0002764201000903

What is Congestion? - Definition from Techopedia. (n.d.). Retrieved from http://www.techopedia.com/definition/18506/congestion-networks

Advocating for Compression Aware Algorithms
(Technical Paper)

An Investigation of the Relationship between Moore's Law and the Data Congestion Problem
(STS Paper)

A Thesis Prospectus Submitted to the

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

Arman Lokhandwala
Fall, 2018

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature _______________________________________ Date ____________
Arman Lokhandwala

Approved _______________________________________ Date ____________
Nathan Brunelle, Department of Computer Science

Approved _______________________________________ Date ____________
Tolu Odumosu, Department of Engineering and Society

Introduction

As Moore's law, a phenomenon describing the rapidly increasing number of transistors in the world, slows down, it is imperative that our techniques for data processing improves (Simonite). The best way to approach development on efficient data processing is to create optimized compression-aware algorithmic techniques capable of efficiently using computing resources while scaling upwards. These computing resources would be best optimized because the new techniques mentioned above would operate directly on compressed data sets whereas traditionally, algorithms are run on uncompressed data, a method which takes too much time and too much space. Implementing algorithms to run on compressed data directly allows for a comparison with the aim of seeing a speed up in runtime when compared to running similar algorithms after decompressing the data first. For my technical project, my group and I will be researching compression aware algorithms.

Although data processing is often overlooked and not considered when evaluating societal norms, efficient data processing plays an enormous role on present day societal and political norms. The idea of congestion is not new, as it exists all around us: automobiles on the roads, people on sidewalks, and of course, the steadily increasing bandwidth and server traffic. Most people do not consider data congestion a problem, as they are totally unaware of it! For my STS research project, I will explore the data congestion problem and investigate how users and non-users effect the problem in relation to the slowing of Moore's law.

Technical Project Details

Data compression is the representation of data in a compressed format by modifying, encoding, or converting how said 'data' (i.e., picture, text file, pdfs, etc.) is structured, stored, and transmitted. The largest and most obvious benefit of data compression is in regard to storage. For example, a string of characters making up a sentence would usually take up a decent amount of memory to store. After compressing the same string using a known algorithm, such as Huffman encoding which represents characters in unique bits, the same string would require far less space to be stored. Thus, compressing data provides benefits that normal uncompressed data could not offer, by saving a significant amount of disc space. Additionally, it allows for quicker and more efficient data transfer, as compressed data is, by definition, data represented using fewer bits and less information than a standard data (file) would.

Jumping back to Moore's Law, it boils down to the idea that the number of transistors doubles nearly every two years, and respectively, so does computing power. However, as the number of transistors increases, so does the cost of making them, as the precision required for such high-density chips becomes increasingly difficult the smaller they get. This trend calls for other measures to come into picture in order to keep up with the demand in computing resources. Hence, creating compression-aware algorithms introduces a way to keep up with the ever-progressing performance observation of Moore's Law, thus providing a life-extension strategy for Moore's Law.

As touched on before, there are a few compression strategies that have been developed and are well known. The relevant one for this technical project revolves around the idea of graph compression. Essentially, graphs will be searched to see if there are any recurring patterns of connected nodes by searching through each node's adjacency list. If long patterns of nodes exist for multiple nodes' adjacency list, a new dummy node will be introduced representing the pattern to replace the longer pattern. This will reduce the overall number of nodes and edges traversed by our algorithms, hence significantly compressing the data.

The aim of my independent research is to implement and test various compression aware algorithms. I will be conducting this research by validating the correctness of various compression aware algorithms and that the predicted outcome is actually possible on real hardware. The need to validate that this approach works on real hardware is because the theorization of compression-aware algorithms and its speedups leaves out constants, since normal algorithms' runtimes are not asymptotically affected by constants. Also, the theory does not discuss the benefits of different levels of cache, which real hardware has, so it is expected that this research provides beneficial insight on compression-aware algorithms and is plausible on real hardware. However, in practice, the constants in our compression-aware algorithms are definitely an aspect that needs to be taken into consideration. Essentially though, I plan to take various data sets, compress them using various compression techniques, run compression-aware algorithms on the new data set, and compare the time and space complexity on the compression-aware algorithms against similar algorithms run on uncompressed data. Additionally, I will run various experiments and evaluations while focusing on three specific variables: size of the input, compressibility of the input, and compressed size of the input. I will focus on continuing my research of compression-aware algorithms based off of the findings of these experiments. The goal of my research for the Fall 2018 semester is to validate prior work and run experiments to support the theory in practice. By the end of the Spring 2019 semester, I plan on having completed the experiments mentioned above, along with a relevant analysis report.

STS Project Details

Many people often overlook the fact that Moore's Law is not actually *a law*, but rather just the observation that the number of transistors doubles about every two years. This observation was named after the co-founder of Intel, Gordon Moore. He first introduced the idea when he wrote a paper observing that the number of transistors on an integrated circuit doubles every year and predicted that the pattern would continue for at least another decade, but later updated his forecasts to the number of transistors doubling every two years (Moore). This observation held true for more than a decade and eventually became the accepted goal for the industry to reach – doubling in power and number every two years. However, transistors are becoming increasingly difficult to optimize further, meaning the goal of doubling computing power every two years is not being met, pressuring researchers and industries alike. This slowdown in computing power would not pose too large of an issue if it were not for the increasing data congestion problem. It is clear that the more data that is sent through a specific medium, for instance radio waves, the lower the fidelity of the data will be. The relationship between the slowing of Moore's Law and data congestion is interesting, especially when considering the contributions and consequences of user and non-users.

This is exactly the relationship that I aim to investigate through my STS research thesis. I plan on first deeper examining Moore's Law, exploring both its origins and its slowing. Specifically, I am looking for more information pertaining to how and why industries and the public decided to use Moore's Law as a goal in the first place; clearly, no exponential growth prediction, such as Moore's Law, can go on indefinitely and must eventually slow. I am curious to find out how users and non-users, specifically technologists and businesspeople, influenced societal expectations to meet the predictions as defined by Moore's Law.

Similarly, I plan on researching how users and non-users contribute and are affected by data congestion by studying past technologies such as the 3G wireless network, past radio frequencies, etc., and how and why their successors (such as 4G) came to be. Possible users of technology (both in the past and present) include working adults in the technology field and even young adults, while non-users might include younger children and more traditional adults who reject using technology (for example, an old-fashioned farmer). Finding the reasoning behind successive technological innovations while focusing on the problem of data congestion will better help orchestrate the relationship of how data congestion drives innovation.

Studying both Moore's Law and the data congestion problem in depth will allow for parallels to be drawn, defining a corresponding relationship between the two. Also, I plan to study Moore's Law extensively through the Social Construction of Technology (SCOT) framework with transistors being the artifact (Pinch 1987). Using this lens to study transistors and Moore's Law will allow me to identify relevant social groups and their contributions to the Law's origins and downfall. The SCOT theory suggests that technology is constructed by society and using it can shed light on how society influences and shapes a certain technology. Studying transistors with this perspective in mind can help uncover the relationship between Moore's Law and society – an aspect that undoubtedly has affected the data congestion problem.

Finally, by gathering data regarding the relationship between Moore's Law and data congestion, I can better understand society's role on the data congestion problem. This problem is worth studying with respect to society's role in that the use of technology and data transfer is increasing at an alarming rate. If we can establish a pattern and make predictions about when, where, and how congested data transmission will be, scientists and industries can anticipate and plan a solution to the issue before it becomes unmanageable.

Conclusion

The end goal of my technical project is to determine whether compression-aware algorithms can provide a speed up in algorithmic runtime. This goal will be approached by validating work of previous compression-aware algorithms as well as by experimenting with different independent variables. The final deliverable for my technical project is a report of results and analysis. Fortunately, my research proves useful in multiple disciplines. First and foremost, my research will validate prior work in the field of compression aware algorithms, strengthening the validity of previous experiments. Secondly, my work will be available as a 'library' or reference document that others can use in the future to delve deeper into the field of compression aware algorithms. Therefore, by validating others' work and providing a physical resource for future use, I will be making a soft contribution to science and the field of research, and a hard contribution for the computing science research community here at the University of Virginia.

At the end of my STS research, I hope to have gained a better understanding of the interconnectedness between Moore's Law, data congestion, and society. I believe presenting trends and patterns between them will allow for an accurate prediction for future use. Hopefully, scientists and industries alike will be better equipped with more data when predicting future production goals and anticipating the inevitably reoccurring data congestion problem.

Resources

Brunelle, Nathan. Super-Scalable Algorithms. 2017. University of Virginia, PhD dissertation.

Hemmendinger, David. “Data Compression.” *Encyclopædia Britannica*, Encyclopædia Britannica, Inc., 23 Apr. 2013, www.britannica.com/technology/data-compression.

Moore, Gordon E. “Cramming More Components onto Integrated Circuits.” *Cramming More Components onto Integrated Circuits*, drive.google.com/file/d/0By83v5TWkGjvQkpBcXJKT1I1TTA/view.

Pinch, Trevor J., and Wiebe E. Bijker. "The Social Construction of Facts and Artifacts: Or How the Sociology of." The Social Constructions of Technological Systems: New Directions in the Sociology and History of Technology (1987): 17.

Simonite, Tom. “Intel Is Putting the Brakes on Moore's Law.” *MIT Technology Review*, MIT Technology Review, 15 Aug. 2016, www.technologyreview.com/s/601102/intel-puts-the-brakes-on-moores-law/.