

A Scalability Metric for Parallel Computations on Large, Growing Datasets (like the Web)

Jesse Weaver

Tetherless World Constellation, Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract. One of the greatest challenges facing computations on data crawled from the Web is the (in)ability to scale to such large quantities of data. While some computations are less challenged by this than others, inference on the Semantic Web is certainly limited in this regard. Parallelism has been employed to scale inference to larger datasets, but evaluations of recent works have fallen back on common parallel computing metrics that do not apply to this specific scalability challenge. In this position paper, the name *data scaling* is given to this scalability challenge, and the metric *growth efficiency* is defined.

1 Introduction

One of the greatest challenges facing computations on data crawled from the Web is the (in)ability to scale to such large quantities of data. Parallelism is often employed to scale computation to larger datasets while keeping execution time reasonable. However, traditional parallel computing metrics focus on one of two forms of scaling: *strong scaling* or *weak scaling*. The goal of strong scaling is to reduce execution time for a fixed problem by adding processors. The goal of weak scaling is to keep execution time constant by adding processors to accommodate additional workload (not necessarily proportional to amount of data). In contrast to these notions of scaling, a new notion of scalability – *data scaling* – is introduced herein, and it is concerned with keeping execution time constant by adding processors to accommodate more *data*.

The ideas presented in this position paper are preliminary in nature, still requiring formal development. The specific intent of this paper, though, is to instigate a paradigm shift in the way scalability evaluations of parallel inference (and possibly other problems) on large, RDF datasets are performed.

In section 2, the need for data scaling is motivated. In section 3, common scalability metrics are shown to be unfit for measuring data scalability, and a new data scaling metric – *growth efficiency* – is defined. Discussion about, and a retroactive example of, evaluating a system with growth efficiency is given in section 4, and conclusion is given in section 5.

2 Motivation: data scaling

This work subscribes to the following statement by Hitzler and van Harmelen: “Concerning scalability, reasoning systems have made major leaps in the recent

past However, it remains an open question when (and if¹) these approaches will scale to the size of the web, . . .” [5]. From this statement, two assumptions are inferred which are used to motivate the work presented herein.

1. It is important for reasoning (including inference) systems to scale toward the size of the Web.
2. The Web is continuously growing.

Perhaps these assumptions are debatable, but for the intents and purposes of this paper, they are considered axiomatic. Additionally, although inference motivates this work, the definitions and their application are not specific to any particular class of problems.

To support definitions throughout this paper, a simplistic, intentionally non-specific computational model is assumed. A (terminating) parallel computation is performed on some dataset, in some amount of time, utilizing some number of processors. This is sufficient for the following discussion.

Indeed, progress has been made for specific forms of reasoning on large datasets. At the very least, varying degrees of RDFS and OWL inference have been achieved on real-world datasets containing around a billion RDF triples [6, 8, 11, 14]. However, in 2010, linked RDF data on the Web consisted of over 24.7 billion RDF triples, and the amount of such data appears to be rapidly growing [1]. Therefore, even if effective inference could be demonstrated on the entire body of RDF data on the (current) Web, it would likely need to scale to even larger datasets in the future.

Evaluations of recently studied, parallel inference approaches [2, 6–13] give no direct indication of how well the approaches will scale to dataset sizes beyond those used in the evaluations. This seems to be due to reliance on scalability metrics traditionally used in parallel computing that do not apply to the challenge of large and growing datasets. Therefore, there is a need to explicitly name and define this scalability issue, and to provide relevant metrics for it.

Data scaling is concerned with the change in execution time of a parallel computation as processors are added to accommodate larger datasets. Data scaling is arguably the central scalability issue for parallel computations on the Web, and it is distinct from strong scaling and weak scaling as illustrated through metrics in the following section.

3 Common scalability metrics and growth efficiency

This section contains a brief review of fundamental metrics often used for measuring scalability (in some sense) of parallel systems. These are defined herein in an atypical way in order to relate the metrics to dataset size, thus highlighting their insufficiency for measuring data scalability. Then a new metric for data scaling is introduced. To aide this discussion, the following definition is needed.

¹ “Since the web keeps growing, they may never scale, even if they become much more efficient.” Footnote in quote from [5]. Footnote number appearing in the quote herein differs from the number used in [5].

Definition 1. A **growing dataset** is effectively a function D that maps positive integers to datasets such that for any positive integer n , $|D(n)| = n$ and $D(n) \subset D(n + 1)$.

Relative speedup and metrics based on it are the scalability metrics reported by nearly every recent work [2, 8–12]. Others report only (variants of) execution time [6, 7, 13].

Definition 2. Let D be a growing dataset, and fix k to some positive integer. Let T_1 be the execution time for one processor with dataset $D(k)$, and let T_P be the execution time for P processors with dataset $D(k)$. Then the **relative speedup** is defined as $S_P = T_1/T_P$.

Clearly, relative speedup (the most common strong scaling metric) gives no direct indication of data scaling since the dataset is fixed. As an alternative, one may resort to (empirical) scaled speedup [4] (the most common weak scaling metric).

Definition 3. Let D be a growing dataset, and fix k to some positive integer. Let t_1 be the execution time for one processor with dataset $D(k)$. Let $i > k$ be a positive integer such that the execution time for P processors with dataset $D(i)$ is t_1 . Then let t_P be the execution time for one processor with dataset $D(i)$. Then the **scaled speedup** is defined as $S_P = t_P/t_1$.

Unlike with relative speedup, in scaled speedup, the dataset size changes with the number of processors. However, processors are added not to accommodate more data but rather to keep execution time constant.

Therefore, these metrics (and those metrics derived from them) are unsuited for measuring data scalability, and a new metric is needed.

Definition 4. Let D be a growing dataset, and fix k to some positive integer. Let \mathcal{T}_1 be the execution time for one processor with dataset $D(k)$, and let \mathcal{T}_P be the execution time for P processors with dataset $D(P \cdot k)$. Then **growth efficiency** is defined as $\mathcal{G}_P = \mathcal{T}_1/\mathcal{T}_P$.

Growth efficiency directly aligns with the notion of data scaling. The idea is that the size of the input dataset should grow linearly with the number of processors, as captured in the definition. Thus, processors are added to accommodate more data. Growth efficiency is more comparable to efficiency $E_P = S_P/P$ or scaled efficiency $\mathcal{E}_P = S_P/P$ in that it is a value between zero and one² (inclusively).

4 Evaluations using growth efficiency

Performance evaluations using growth efficiency are fairly intuitive, but there are some important details, particularly when the evaluation is meant to compare systems.

² That is, in theory. Although undetermined at present, there may exist some conditions in practice that allow for growth efficiency to be greater than one. This would be akin to superlinear speedup in which efficiency can be greater than one.

Points $\langle x, y \rangle$ in scatter plots should be such that x is the dataset size and y is the growth efficiency. Using notation from definition 4, the points are $\langle P \cdot k, \mathcal{G}_P \rangle$ for some k and various P . This brings up the issue of what k should be. k is the amount of data for a single processor, and as such, it is referred to herein as the *processor capacity*. Processor capacity can be defined in numerous ways. One possibility is availability of space (e.g., RAM, disk); another possibility is the maximum amount of data a single processor can handle without exceeding a specific upper bound on execution time. Regardless, an evaluation should include discussion and justification of how processor capacity is determined.

It is often the case that evaluations of different systems are performed independently of each other, and some meaningful comparison is retroactively sought. Thus, consideration must be given to potential differences in choice of growing dataset and notion of processor capacity. That is, evaluations are comparable only in as much as the parameters of the evaluations are comparable.

Growing datasets should be similar, if not the same. Meeting this requirement is straightforward with synthetic datasets, but more difficult with real-world datasets. Unless it is obvious, an evaluation should make clear the method by which the dataset was linearly “grown” with number of processors. It is conceivable that the order of adding data can vary the change of execution time, for example, in the context of inference with negation as failure.

Notions of processor capacity should be similar, although it is not necessary that they be exactly the same. For example, two evaluations using different notions of space-bounded, processor capacity are likely to still be comparable strictly in the context of data scaling.

Although more thorough discussion is warranted, an example, retroactive evaluation illustrating the differences of strong scaling and data scaling would likely be a better use of the remaining space, given the limitation on paper length. Some of the results from parallel, RDFS inference in [13] are reorganized in this section to address data scaling. The growing dataset is LUBM [3] generated using a seed of zero. A notion of space-bounded capacity is used, specifically RAM-bounded capacity. In this case, the processor capacity is 2,699,360 triples. This is not necessarily the maximum processor capacity, but since this evaluation is retroactive, it is sufficient for the purposes of demonstration.

This example is intended to illustrate how growth efficiency differs from efficiency $E_P = S_P/P$ in the strong scaling sense. Therefore, the two are plotted below over number of processors. (Recall that it was stated that the x-axis should be dataset size for growth efficiency, but such an x-axis is nonsensical for efficiency.)

Table 1a shows metrics for the overall computation, which includes I/O from/to a parallel file system. Efficiency and growth efficiency are plotted for number of processors in figure 1a. Growth efficiency for 256 processors is 0.66. That is, 256 times as much data was processed in about $1/0.66 \approx 1.5$ times as much time as with a single processor. Efficiency, which is 0.36 for 256 processors, does not make this evident at all.

Table 1b and figure 1b show the same metrics for only the inference portion of the computation. The inference portion of the computation is embarrassingly parallel, so there is no interprocess communication or contention. Clearly, the inference portion of the computation is very scalable – at least for LUBM data – in both the strong and data scaling senses. This indicates that the inference portion of the computation will likely scale to very large datasets without significantly impacting execution time, although the same cannot be said for the overall computation.

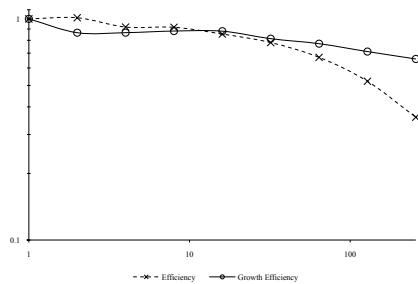
P	T_P	E_P	\mathcal{T}_P	\mathcal{G}_P
1	360	1.00	360	1.00
2	178	1.01	415	0.87
4	98.0	0.92	415	0.87
8	49.1	0.92	408	0.88
16	26.3	0.85	409	0.88
32	14.4	0.78	442	0.81
64	8.39	0.67	466	0.77
128	3.91	0.52	506	0.71
256	3.91	0.36	546	0.66

(a) Overall

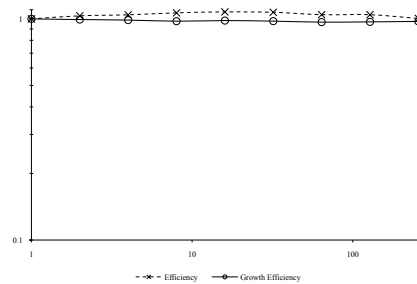
P	T_P	E_P	\mathcal{T}_P	\mathcal{G}_P
1	283	1.00	283	1.00
2	137	1.03	285	0.99
4	67.7	1.04	286	0.99
8	33.2	1.06	289	0.98
16	16.4	1.08	288	0.98
32	8.24	1.07	289	0.98
64	4.23	1.04	292	0.97
128	2.11	1.04	291	0.97
256	1.10	1.00	290	0.97

(b) Inference Only

Table 1: Execution times, efficiency, and growth efficiency up to 256 processors



(a) Overall



(b) Inference Only

Fig. 1: Efficiency and growth efficiency (log/log) up to 256 processors

5 Conclusion

Traditional scalability metrics from parallel computing fail to address the specific scalability challenge faced by parallel computations on data crawled from the Web, that is, the ability to handle large, growing datasets. A notion of data

scaling has been defined that is concerned with how execution time is affected as data grows, increasing the number of processors linearly with dataset size. A new metric, growth efficiency, has been introduced for evaluating data scalability of parallel computations. Focus has been on inference over RDF data crawled from the Semantic Web.

Acknowledgements. Much thanks to Jacopo Urbani and David Mizell for their constructive feedback in revising this paper.

References

1. Bizer, C.: Pay-as-you-go Data Integration on the public Web of Linked Data. Keynote Presentation at the 3rd Future Internet Symposium (September 2010)
2. Goodman, E.L., Mizell, D.: Scalable In-memory RDFS Closure on Billions of Triples. In: Proceedings of the 6th International Workshop on Scalable Semantic Web Knowledge Base Systems (2010)
3. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web* 3(2-3) (2005)
4. Gustafson, J.L.: Reevaluating Amdahl's Law. *Communications of the ACM* 31(5), 532–533 (May 1988)
5. Hitzler, P., van Harmelen, F.: A Reasonable Semantic Web. *Semantic Web Journal* 1(1), 39–44 (2010)
6. Hogan, A., Pan, J.Z., Polleres, A., Decker, S.: SAOR: Template Rule Optimisations for Distributed Reasoning over 1 Billion Linked Data Triples. In: Proceedings of the 9th International Semantic Web Conference. (2010)
7. Kaoudi, Z., Miliaraki, I., Koubarakis, M.: RDFS Reasoning and Query Answering on Top of DHTs. In: Proceedings of the 8th International Semantic Web Conference. pp. 499–516 (2008)
8. Kotoulas, S., Oren, E., van Harmelen, F.: Mind the Data Skew: Distributed Inferencing by Speeddating in Elastic Regions. In: Proceedings of the 19th International World Wide Web Conference (2010)
9. Oren, E., Kotoulas, S., Anadiotis, G., Siebes, R., ten Teije, A., van Harmelen, F.: Marvin: Distributed reasoning over large-scale Semantic Web data. *Web Semantics: Science, Services and Agents on the World Wide Web* 7(4), 305–316 (2009)
10. Soma, R., Prasanna, V.K.: Parallel Inferencing for OWL Knowledge Bases. In: Proceedings of the 37th International Conference on Parallel Processing. pp. 75–82 (2008)
11. Urbani, J., Kotoulas, S., Maassen, J., van Harmelen, F., Bal, H.: OWL reasoning with WebPIE: calculating the closure of 100 billion triples. In: Proceedings of the 7th Extended Semantic Web Conference (2010)
12. Urbani, J., Kotoulas, S., Oren, E., van Harmelen, F.: Scalable Distributed Reasoning using MapReduce. In: Proceedings of the 8th International Semantic Web Conference (2009)
13. Weaver, J., Hendler, J.A.: Parallel Materialization of the Finite RDFS Closure for Hundreds of Millions of Triples. In: Proceedings of the 8th International Semantic Web Conference. pp. 682–697 (2009)
14. Williams, G.T., Weaver, J., Atre, M., Hendler, J.A.: Scalable Reduction of Large Datasets to Interesting Subsets. *Web Semantics: Science, Services and Agents on the World Wide Web* 8 (2010)