Empirical Analysis of Space Filling Curves for Scientific Computing Applications

Dary DeFord[1] Ananth Kalyanaraman[2]
Washington State University

Abstract

Space Filling Curves are frequently used in parallel processing applications to order and distribute inputs while preserving proximity. Several different metrics have been proposed for analyzing and comparing the efficiency of different space filling curves, particularly in database settings. Here, we introduce a general new metric, called **Average Communicated Distance,** that models the average pairwise communication cost expected to be incurred by an algorithm that makes use of an arbitrary space filling curve. For the purpose of empirical evaluation of this metric, we modeled the communications structure of the Fast Multipole Method for n body problems.

Using this model, we empirically address a number of interesting questions pertaining to the effectiveness of space filling curves in reducing communication, under different combinations of network topology and input distribution settings. We consider these problems from the perspective of ordering the input data, as well as using space filling curves to assign ranks to the processors. Our results for these varied scenarios point towards a list of recommendations based on specific knowledge about the input data. In addition, we present some new empirical results, relating to proximity preservation under the average nearest neighbor stretch metric, that are application independent.

Introduction

Many applications of parallel computing rely on distributing codependent portions of a given problem onto multiple processors. This communication behavior often limits the performance of algorithms in practice, as each processor's computations cannot be performed without the data, but generally all of the processors are trying to communicate at the same time over the same network.

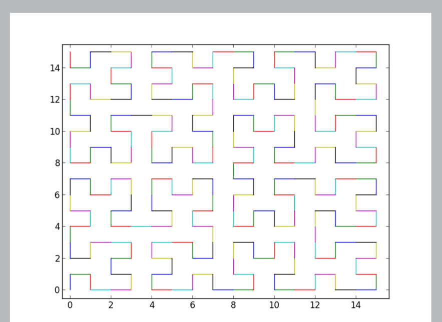
In the context of these parallel computing applications, space filling curves may be implemented in two distinct fashions. Most commonly, curves are used to provide an ordering on a set of n dimensional input points that is used to assign the data to individual processors. Such curves may also be used to rank the processors themselves.

The problem of determining the efficiency of any particular curve for a given application has been studied extensively [1, 3], particularly in the context of data selection and attribute clustering [4, 6]. In 2012, Xu and Tirthapura showed that all continuous curves were asymptotically equivalent under the most commonly used clustering metrics [8].

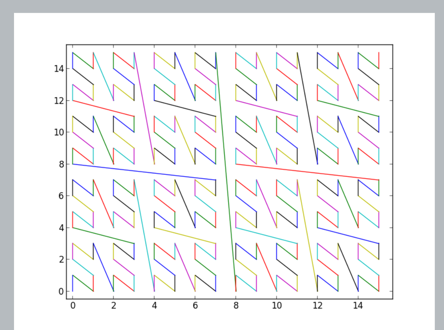
This led us to construct a metric (Average Communicated Distance) that permits algorithm designers to differentiate between the expected communication costs of various curves. Here, we define this metric and use it to evaluate an abstraction of the Fast Multipole Method for n body problems. These empirical results allow us to provide recommendations for scientists working in parallel computing.

Definition 1 (SFC)

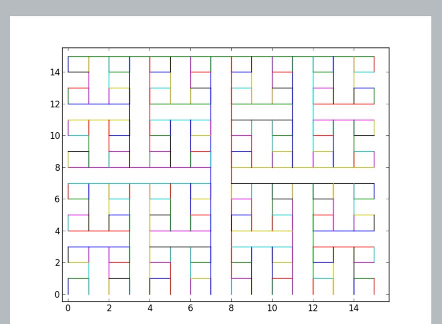
For our purposes, a Space–filling Curve (SFC) is a mapping from a multi–dimensional space to a linear ordering that allows for unique indexing of the points in that space.



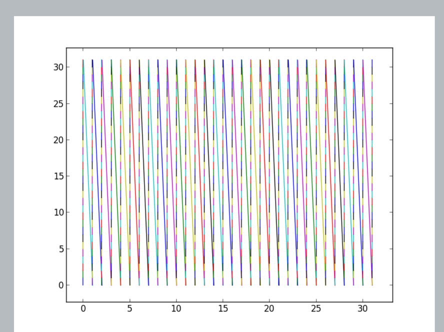
(a) Hilbert Curve $\mathcal{H}_4$



(b) Z-Curve $\mathcal{Z}_4$



(c) Gray Order $\mathcal{G}_4$



(d) Row/Column–Major

Figure 1 : An example illustration of the Space-Filling Curves considered in our study.

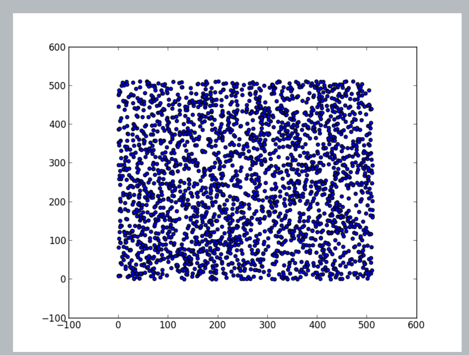
Defintion 2 (ACD)

Given a particular problem instance, the *Average Communicated Distance (ACD)* is defined as the average distance for every pairwise communication made over the course of the entire application. The communication distance between any two communicating processors is given by the length of the shortest path (measured in the number of hops) between the two processors along the network intraconnect.

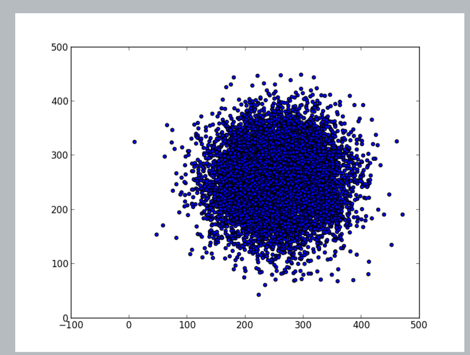
Definition 3 (FMM)

The Fast Multipole Method (FMM) is an algorithm for computing the interactions in an n body problem [2]. We modeled the communications structure of this algorithm as a case study because it relies on computing the Near Field Interactions (NFI) and Far Field Interactions (FFI) separately. Each of these sets of computations has a different communications profile and requires distinct analysis under the ACD metric.

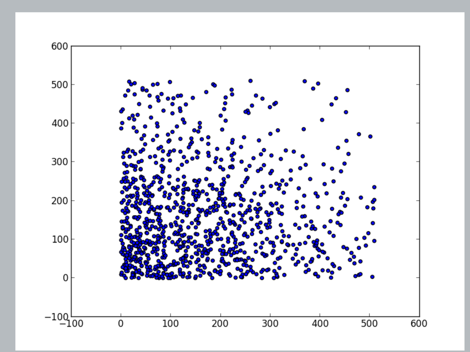
Probability Distributions



(a) Uniform Distribution



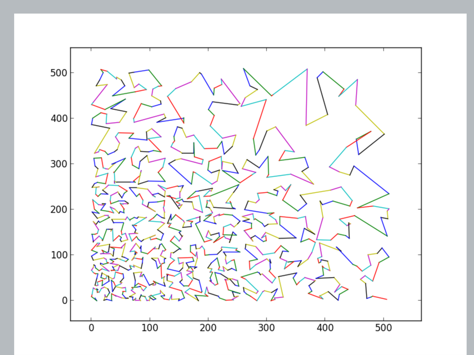
(b) Normal Distribution



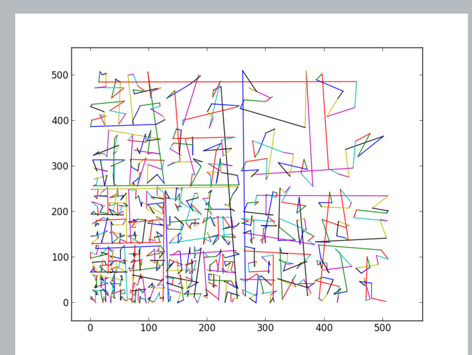
(c) Exponential Distribution

Figure 2 : This figure shows examples of the two dimensional probability distributions considered in this paper.

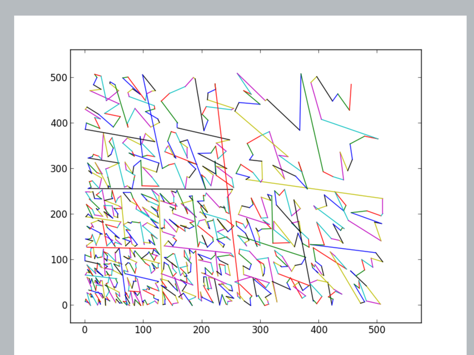
Ordered Points



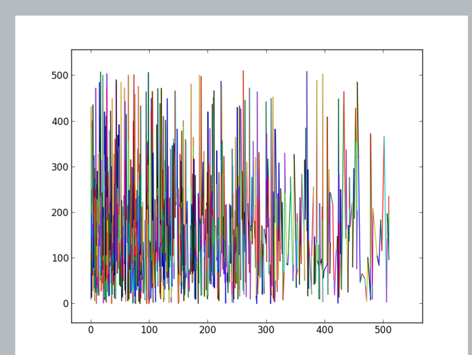
(a) Hilbert Ordering



(b) Gray Ordering



(c) Z Ordering



(d) Row Major Ordering

Figure 3 : As an example of particle–ordering SFCs, this figure shows the linear order of the exponentially distributed particles displayed in Figure 2(c) by each of the SFCs respectively. It is interesting to observe the large "jumps" that occur in the orderings by the discontinuous curves, (b), (c), and (d), especially along the lines of symmetry [8].

Research Questions

We addressed the following four research questions using our empirical models:

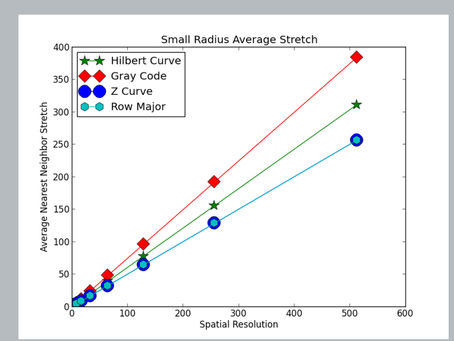
Q1) *What is the nearest-neighborhood preservation efficacy achieved by different particle–order SFCs?*

Q2) *What is the effect of different combinations of {particle-order, processor-order} SFCs on the Average Communicated Distance metric?*

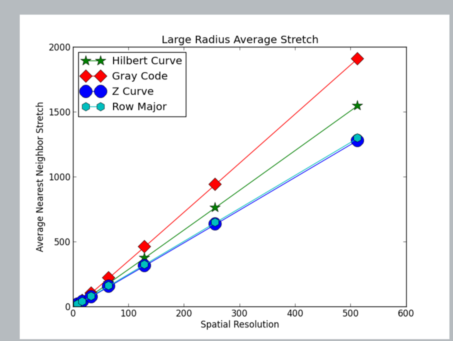
Q3) *What is the performance of each of the particle-order SFCs under the ACD metric, for a given network topology? Similarly, what is the performance of each of the network topologies under the ACD metric, for a given input distribution?*

Q4) *How does the Average Communicated Distance vary as a function of processor size, input size and input distribution, for each SFC?*

A1) ANNS Results



(a) Standard ANNS



(b) Large Radius ANNS

Figure 4 : This figure shows the ANNS values [7] of the SFCs under consideration as the spatial resolution varies. Expanding the radius (b) does not affect the relative ordering of the SFCs. This confirms the theoretical calculations of Xu and Tirthapura on the Z and Row Major curves, and suggests that proximity preservation is not the best measure of SFC effectiveness for scientific computing [5].

A2) Main Results (NFI)

Table 1 : A comparison of different particle/processor-order SFC combinations for NFI under various distributions. The lowest ACD value within each row is displayed in **boldface blue**, while the lowest ACD value within each column is displayed in *red italics*. The best option for each distribution is displayed in ***bold green italics***.

	Particle Order			
Processor Order ↓	Hilbert Curve	Z–Curve	Gray Code	Row Major
Hilbert Curve	***4.008***	*4.308*	*4.939*	*13.117*
Z–Curve	**5.486**	5.758	6.573	18.127
Gray Code	**5.802**	6.010	6.970	19.220
Row Major	**9.126**	9.763	11.713	70.353

(a) Uniform Distribution

	Particle Order			
Processor Order ↓	Hilbert Curve	Z–Curve	Gray Code	Row Major
Hilbert Curve	***8.561***	*9.297*	*10.123*	*20.340*
Z–Curve	**11.003**	11.551	12.984	26.842
Gray Code	**11.881**	12.595	13.249	28.188
Row Major	**20.143**	22.221	24.053	66.719

(b) Normal Distribution

	Particle Order			
Processor Order ↓	Hilbert Curve	Z–Curve	Gray Code	Row Major
Hilbert Curve	***5.238***	*5.654*	*6.271*	*14.943*
Z–Curve	**6.943**	7.070	8.235	20.851
Gray Code	**7.276**	7.663	8.760	22.269
Row Major	**12.483**	13.017	15.289	61.227

(c) Exponential Distribution

A2) Main Results (FFI)

Table 2 : A comparison of different particle/processor-order SFC combinations for FFI under various distributions. The lowest ACD value within each row is displayed in **blue boldface**, while the lowest ACD value within each column is displayed in *red italics*. The best option for each distribution is displayed in ***bold green italics***.

	Particle Order			
Processor Order ↓	Hilbert Curve	Z–Curve	Gray Code	Row Major
Hilbert Curve	***19.494***	*20.841*	*22.572*	*31.124*
Z–Curve	**24.217**	24.793	27.787	37.709
Gray Code	**24.622**	25.446	27.997	39.282
Row Major	**44.513**	48.762	50.118	57.880

(a) Uniform Distribution

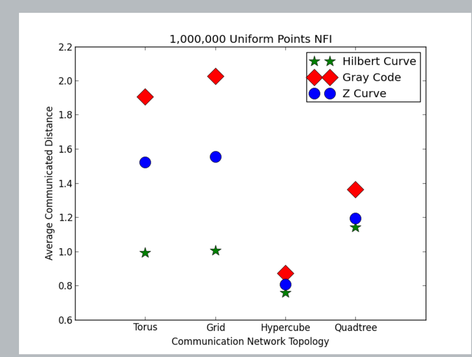
	Particle Order			
Processor Order ↓	Hilbert Curve	Z–Curve	Gray Code	Row Major
Hilbert Curve	***26.336***	*26.824*	31.963	*32.542*
Z–Curve	29.160	**28.036**	34.241	36.663
Gray Code	29.449	**27.981**	*31.909*	37.291
Row Major	**43.639**	44.636	49.133	45.475

(b) Normal Distribution

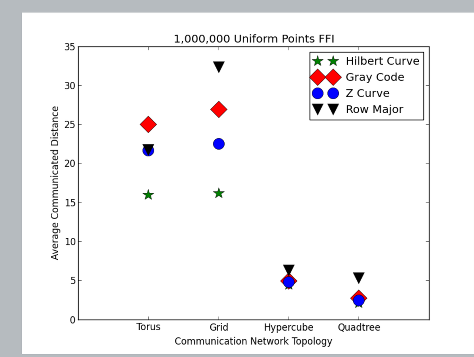
	Particle Order			
Processor Order ↓	Hilbert Curve	Z–Curve	Gray Code	Row Major
Hilbert Curve	***18.960***	*19.841*	*23.007*	*31.368*
Z–Curve	24.672	**23.316**	26.315	37.576
Gray Code	**23.762**	24.076	27.973	37.863
Row Major	**42.447**	44.067	46.872	50.963

(c) Exponential Distribution

A3) Topology Comparison



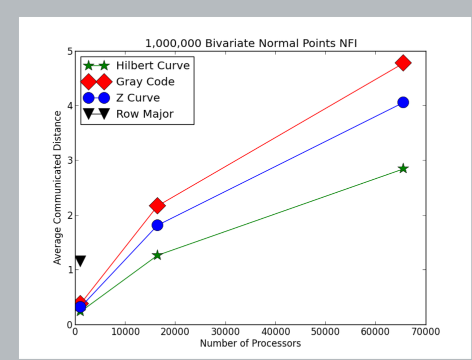
(a) Near–Field Interactions



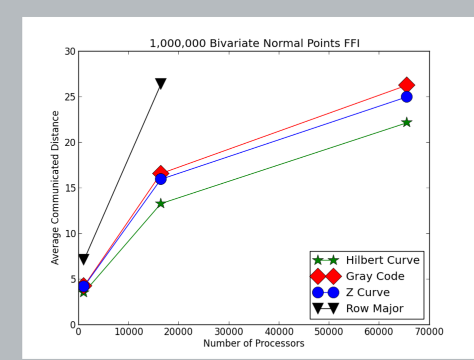
(b) Far–Field Interactions

Figure 5 : The charts show the results of comparing different network topologies for a) NFI and b) FFI, respectively. All experiments were performed using $1,000,000$ uniformly distributed particles on a 4096×4096 spatial resolution. This plot is representative of all the experiments we performed to evaluate the topologies. It is important to note that quadtree structures have disproportionately large issues with contention in high volume communications.

A4) ACD Scaling



(a) NFI



(b) FFI

Figure 6 : These plots show ACD values for a) NFI, and b) FFI, as a function of the number of processors and the SFC used. The input used was fixed at 1,000,000 uniformly distributed particles. This demonstrates the effect scale on processor ranking SFCs. Some of the row–major data has been excluded from these plots because for this SFC, the ACD values at larger processor numbers were significantly higher than the other data–points.

Analysis

Our results point towards a set of reccomendations for designers of parallel alogorithms for scientific computing. When the scientist has full control over both the data distribution and processor ranking, using the Hilbert Curve at both stages gives the lowest ACD values. Unfortunately, such control is not always feasible or desirable, in which case we present the following recommendation for SFC selection based on the ACD values:

$$\{Hilbert \approx Z\} < Gray << Row\text{-}major.$$

Future Work and Extensions

We intend to further extend our results by considering the following extensions:

- Adding a weighting function to evaluate dataintensive applications
- Extending our metric to consider contention based communications models
- Extending our evaluation to real world implementations and applications other than FMM.
- Providing a closed, asymptotic expression for the ANNS of more complex curves.
- One of the interesting notions encountered in this work is the mapping of points from a multi–dimensional space to a 2D torus or mesh. This is unlike the traditional SFC problem, and does not appear to have been explored yet in theory. In this paper, we used SFCs to move from 2D to a linear ordering back to 2D, but certainly there appears to be no restriction on a direct mapping into the processor space. This raises theoretical questions for further study.
- Finally, while we expect the conclusions of most of the studies conducted in this paper to extend to 3D, further experimentation is needed to corroborate such trends.

Conclusions

Our results empirically validate previously published theoretical results. In addition, based on our results, we provided a list of recommendations that could serve as benchmarks for effective use of SFCs in FMM-type applications. Our findings suggest both theoretical avenues of inquiry for future research and practical applications of particular SFCs, both for distributing the input data among parallel processors, and for canonical labeling of processors on a particular network topology, with an overall goal of minimizing communication network usage. In particular, the ACD metric presented here represents an important contribution to the study of SFCs for scientific computing.

References

[1] S. Aluru and F. Sevilgen: *Parallel Domain Decomposition and Load Balancing Using Space–Filling Curves*, Fourth International Conference on High Performance Computing, (1997) 230–235.

[2] R. Beatson and L. Greengard: *A Short Course on Fast Multipole Methods*, Wavelets, Multilevel Methods and Elliptic PDEs, Oxford University Press, Oxford, (1997) 1–37.

[3] B. Hariharan and S. Aluru: *Efficient Parallel Algorithms and Software for Compressed Octrees with Applications to Hierarchical Methods*, Parallel Computing, **31** (2005), 311–331.

[4] H. Jagadish: *Linear Clustering of Objects with Multiple Attributes*, Proceedings ACM SIGMOD (1990), 332–342.

[5] H. Jagadish: *Analysis of the Hilbert Curve for Representing Two–Dimensional Space*, Information Processing Letters **62** (1997), 17–22.

[6] B. Moon, H. Jagadish, C. Faloutsos, and J. Saltz: *Analysis of the Clustering Properties of Hilbert Space–Filling Curve*, IEEE Transactions on Knowledge and Data Engineering, **13 (1)** (2001), 124–141.

[7] P. Xu and S. Tirthapura: *A Lower Bound on Proximity Preservation by Space Filling Curves*, Proc. International Parallel and Distributed Processing Symposium 2012, (2012), 1295 – 1305.

[8] P. Xu and S. Tirthapura: *On the Optimality of Clustering Properties of Space Filling Curves*, Proc. Principles of Database Systems 2012, (2012), 215–224.

Acknowledgments

The authors would like to thank Professor Nairanjana Dasgupta and Professor Srikanta Tirthapura for their useful discussions. This research was supported by NSF grant IIS 0916463.

[1]Department of Mathematics [2]School of Electrical Engineering and Computer Science Created With LaTeX BEAMERPOSTER, OCTAVE, and MATPLOTLIB daryl.deford@email.wsu.edu