Publishing Scheme for Skewed Data Set

Zhenhao Cao
SEIEE
Shanghai Jiao Tong University
Shanghai, China
Hazelnut@sjtu.edu.cn

Yichen Zhu
SEIEE
Shanghai Jiao Tong University
Shanghai, China
zyc_IEEE@sjtu.edu.cn

***Abstract*—Efficient distributed indexing scheme is critically important for the data management of data center network. Since the distribution of data has a great influence on the efficiency of data retrieval, data publishing scheme is a key part of the design of distributed indexing scheme. In this paper, we propose one basic and two advanced publishing schemes in BCube data center network for skewed data set, in which data points are distributed unevenly. In sorting-based mapping scheme, data points are mapped to their corresponding sorting number. In casting-based mapping scheme, we reduce the complexity of the basic scheme by approximation algorithm. In modified casting-based mapping scheme, we append another mapping to casting-based mapping to achieve better distribution. We simulate a case of $BCube_2$ and define the metrics for uniformity distribution. We also compare the results generated by an existing scheme, direct-mapping, and our schemes.**

***Index Terms*—data center, data publishment, skewed data set**

I. Introduction

With the rapid development of data-intensive applications, cloud storage systems are put forward to support these applications. Data center is a special kind of cloud storage system, where a large number of servers and associated components are connected as a network to accelerate the communication between servers and meet the requirements of reliability, scalability and regularity. One challenge for data management in data center networks is how to improve the efficiency of data retrieval. Designing a distributed indexing scheme is a typical solution to this challenge.

In the indexing scheme, an indexing space is created and divided into several partitions, and each server in the data center network is responsible for one partition of the whole indexing space. Data publishment stands for the process of determining the data that should be stored on a certain server. Since data distribution among servers has a tremendous influence on the efficiency of data retrieval, data publishing scheme is of great significance in the design of distributed indexing scheme.

We focus on BCube in this project, which is a typical type of data center network. Some works have already been proposed for data publishment in BCube network, yet they have some drawbacks that may lead to a degradation in retrieval efficiency. In this project, we first present an intuitive and basic scheme for data publishment. Moreover, we improve the basic scheme by approximation and present a data publishing scheme with higher efficiency. Finally, we conduct experiments to show the results and performance of our scheme.

II. Related Works

BCube is a typical type of data center network [1]. The structure of BCube is recursively defined. Basic structure of BCube is $BCube_0$, which contains one switch and n servers. The unit of a higher-level $BCube_k$ is $BCube_{k-1}$, and $BCube_k$ is composed of n units. Figure 1 shows an example of $BCube_2$ with $n = 4$, which consists of 64 servers.

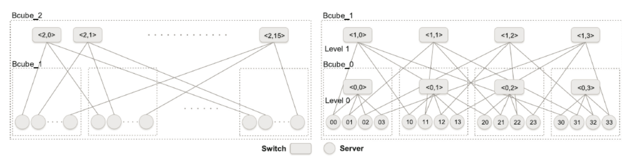


Fig. 1. Structure of $BCube_2$ with $n = 4$

Based on the characteristic of the structure of BCube, a hyper-cube indexing space is created. Take the example in Figure 1. Each dimension is divided into 4 parts and the whole space is divided into 64 parts. Each part is managed by one server in the network. Data publishment is to determine the data that should stored on each of these 64 servers. Figure 2 shows an example of the indexing space of BCube.

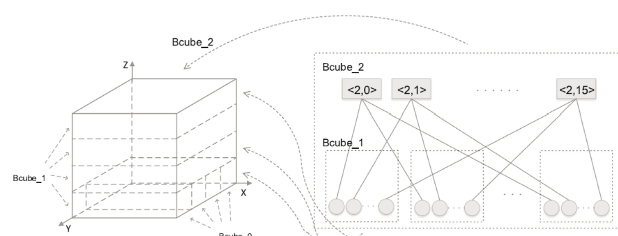


Fig. 2. Indexing space for $BCube_2$ with $n = 4$

Essentially, data publishment is the process of mapping a point in the data set to a point in the indexing space. The data point will be managed by the server whose range covers the point mapped to in the indexing space. Some existing works directly maps the data point into the indexing space. However, this may leads to a degradation in retrieval efficiency. For a skewed data set, in which the distribution of data points is uneven, some servers manage more data and some manage less. Assuming the visiting possibility of each data point is equal, those servers that manages more data will be visited more frequently, leading to hot spot problem.

In order to address the hop spot problem and improve retrieval efficiency, we present publishing schemes especially for skewed data set.

III. Mapping Scheme

A. Basic Scheme: Sorting-Based Mapping (SBM)

We focus on the mapping of one-dimension data. Sorting-based mapping maps a data point to its sorting number. Let $D = \{d_1, d_2, \cdots, d_n\}$ be a data set in ascending order of value and $v(d_k)$ be the mapping value of d_k. Then $v(d_k)$ is defined in Equation (1).

$$v(d_k) = \frac{k}{n}, \quad k = 1, 2, \cdots, n \tag{1}$$

We take data set $D^{'} = \{1, 5, 10, 3, 18, 15, 2, 6\}$ as an example. Obviously, data set $D^{'}$ is a skewed data set. First, sort the data points in ascending order and we have $D^{'} = \{1, 2, 3, 5, 6, 10, 15, 18\}$. According to Equation (1), the mapping value of the data points are $1/8, 2/8, \cdots, 8/8$ respectively. Figure 3 shows how we realize the mapping. The horizontal axis represents the original data points $\{1, 2, 3, 5, 6, 10, 15, 18\}$, which is unevenly distributed. The vertical axis represents the mapping values of these data points, which is evenly distributed.

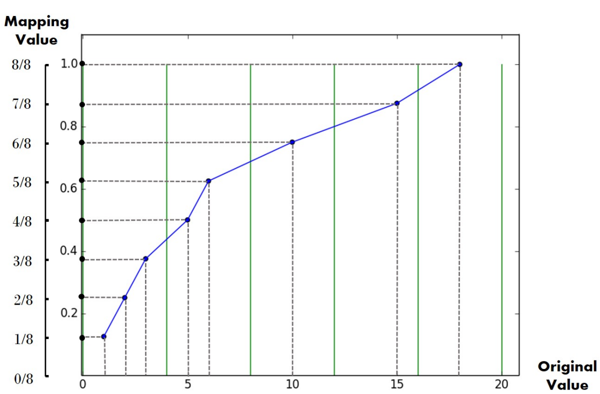


Fig. 3. Example of Sorting-Based Mapping

In sorting-based mapping, mapping values of data points are uniformly distributed in $[0, 1]$ in one-dimension space. Therefore, we achieve a mapping scheme that generates a uniform distribution. Apply sorting-based mapping to all dimensions, the data points are mapped into a unit hyper-cube space.

B. Advanced Scheme: Casting-Based Mapping (CBM)

For sorting-based mapping, obtaining the mapping values of data points requires sorting all the data points, which is expensive. In order to reduce the complexity, we use approximation algorithm to optimize the mapping scheme. In Figure 3, the curve is segmented at each data point. Instead, we use one segment to fit several segments to achieve a reduction in complexity.

We use an example shown in Figure 4 to show the process of obtaining the mapping values. We divide the data range into 5 buckets with boundaries 0, 4, 8, 12, 16 and 20. Then we calculate the counting value of these boundaries, that is the number of data points which is smaller than this boundaries. The counting values of the boundaries are 3, 5, 6, 7 and 8. Use the same method in sorting-based mapping, we get mapping values of these boundaries $0, 3/8, 5/8, 6/8, 7/8, 8/8$ and generate the red curve. Those larger red points stands for the boundaries. We project the data points to the red curve and get those smaller red points. Finally, we project the smaller red points to the vertical axis and get the mapping values of these points.

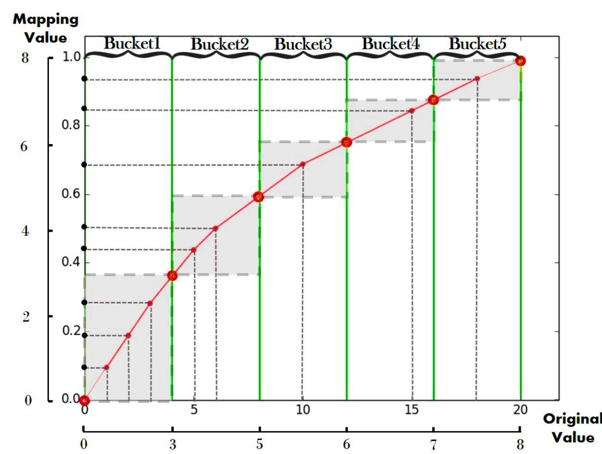


Fig. 4. Example of Casting-Based Mapping

Formally, suppose the data range $[0, L]$ is divided into p buckets with boundaries $b_0, b_1, \cdots, b_p$ and corresponding counting values $c_0, c_1, \cdots, c_p$. Let the number of data points be n. For a data point d casted to the $(k+1)^{th}$ bucket, its mapping value $v(d)$ is defined in Equation (2).

$$v(d) = \frac{\frac{c_{k+1} - c_k}{b_{k+1} - b_k}(d - b_k) + c_k}{n}, \quad k = 1, 2, \cdots, n \tag{2}$$

Instead of sorting all the data points, we divide the whole data range into several buckets and cast a data point to its corresponding bucket. The mapping values of boundaries can be obtained by a scan of the data set and do not require any sorting. In Figure 4, we use the segment of a bucket to approximate the corresponding segments of data points casted to this bucket.

We use a data set containing 25 data points to intuitively show the approximation process. Figure 5 shows the result. The red curve represents casting-based mapping and the blue curve represents sorting-based mapping. By observation, it is obvious that the red curve well fits the blue curve.

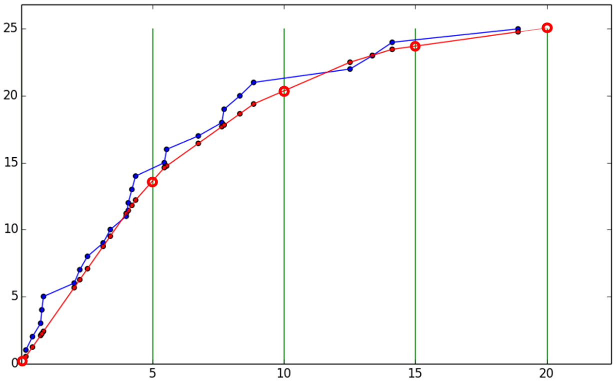


Fig. 5. Approximation of 25 Points

An improvement of Casting-Based Mapping, named Modified Casting-Based Mapping, is illustrated in the discussion part, and it reaches a better performance.

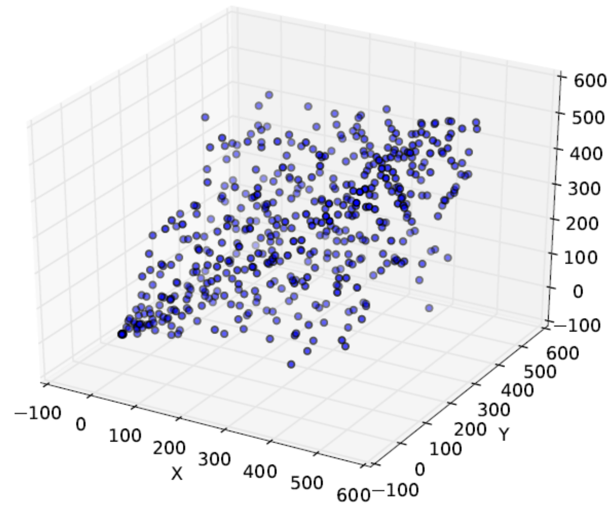


Fig. 6. Distribution after Casting-Based Mapping

IV. Discussion

In sorting-based and casting-based mapping scheme, we focus on one-dimension mapping and thus achieve even distribution in one-dimension. However, for multidimensional data, such mapping schemes may result in mapping points approaching the diagonal of the indexing space. In this section, we raise some of our ideas of this problem and analyze the advantages and disadvantages of them.

A. Random Mapping (RM)

In random mapping scheme, we generate a point in the indexing space randomly for a data point. For each data point, we maintain a list to record the point it is mapped to. Since the point in indexing space is randomly generated, mapping points are evenly distributed in multidimensional indexing space. However, for this scheme, we need to maintain a list that records the mapping point of each data point, and we need to traverse the list to search for data, which greatly reduces the efficiency of data retrieval.

B. Modified Casting-Based Mapping (MCBM)

After casting-based mapping, data points are mapped to an even distribution in one-dimension data. However, mapping points are still unevenly distributed to some extent in multi-dimensional indexing space as shown in Figure 6. Note that data points tend to gather around the spatial line $x = y = z$. In fact, most of our experiments using SBM and CBM have shown this phenomenon, and the reason is actually the high symmetric between dimensions. Previously, we only focus on the uniformity in single dimension and ignore the failure of uniformity mapping brought by inter-dimensional symmetry.

To settle this problem, we have to destroy or avoid the highly symmetric patterns between dimensions. The 'Destroy' method means shuffling the data order in at most $(dd?1)$ dimensions. The drawbacks are apparent: shuffling will also destroys the locality of data distribution and cause the loss of support for range querying; additionally, an extra $O(N)$ space is required for storage of the after-shuffle index. Since practical data is massive, this space cost is extremely high and unacceptable. Thus, here we choose 'Avoid' method to cope with the symmetry, and formed our Modified Sorting-Based Method.

a) Main Idea: The key idea of Modified Casting-Based Method is to take advantage of the single-dimension-uniformity based on CBM. After casting-based mapping, we choose the dimension in which mapping points are most evenly distributed. The dimension we select is called 'dominant dimension'.

b) Selection of Dominant Dimension: We define Relative Uniformity for single dimension, which can be calculated by the following equation, where $Slope_j$ represents the slope of j^{th} bucket's line segment in dimension i.

$$RU_i = \prod_{j=1}^{p} Slope_j$$

Note that the larger is the value of RU_i is, the more uniform is the i^{th} dimension. The proof can be completed by induction. Here we show the case with two buckets. The proportion of the points in the first and second bucket is α_1 and α_2. We have $\alpha_1 \times \alpha_2 = \max_{\alpha_1 \in \alpha} \alpha_1(1 - \alpha_1)$. Clearly, the object reaches maximum when α_1 is equal to α_2, and that is when all the buckets share the same number of points, which means the best uniformity.

c) Single Dimension to D-Dimension: We choose the value of all mapping points in this dimension and convert it to a 3-bit $k - hex$ number, where k^3 just exceeds the number of data points. For example, if the number of data points is 500, k should be 8 because $8^3 = 512$ just exceeds 500. We map the data point to a point in the indexing space according to the 3-bit number. For example, if k is equal to 2 and a mapping value is 7, then this point should be mapped to coordinate $(1, 1, 1)$ in the indexing space. Also, we can just use binary translation and bit-segmentation to realize the mapping from one-dimension to three-dimension. The internal rationale is that if data have reached a good uniformity (single dimension by CBM), then the distribution on every bit of a binary is well uniform.

Formally, let dimension i be the dimension with most even distribution, $CBM_i(d)$ be the mapping value of data point d in i^{th} dimension, and $HEX_k(x)$ be a triple in k-hex of number x, the mapping coordinate in indexing space $cor(d)$ is defined in Equation (3)

$$cor(d) = HEX_k(CBM_i(d)) \tag{3}$$

d) Performance Analysis: In MCBM, the former part is CBM with the time complexity of O(N), and the latter part is a binary translation which only requires constant time cost due to the limited data scale. So, the entire time complexity of MCSM is O(N). The time complexity for query is O(1), only requiring to calculate the CBM mapping value then do the binary translation once. The space complexity for MCBM is constant because the number of buckets is constant. Bucket structure only includes the lower bound, higher bound and the slope, assisting to calculate the mapping value when querying. In this scheme, another mapping is appended and mapping points get more evenly distributed in the indexing space without reducing the efficiency of data retrieval and increasing complexity.

V. Experiment Evaluation

In this section, we describe our experimental data set and define the evaluation metrics for mapping uniformity in high dimension space. Thereafter, we illustrate the procedure of experiments, show the results of our experiment and then compare the outcomes of directly-mapping method, sorting-based method, casting-based method and the modified casting-based method.

A. Data Description and Simulation

Typically, the practical skewed data set is randomly distributed and even formed as an overlap of multiple existing distributions in sole or several types. Due to the high variability and uncertainty of transmit data in factual publishing scheme, no known patterns exist so far that could perfectly feature a certain distribution mode for sure. So, we come up with several approaches to generate some data distribution patterns using the most common distributions as well as their combinations, and test the performance of our methods comprehensively by adjusting the parameters and data scales. Without the loss of generality, our experiment is performed in three-dimension space, requiring the generated data to be three-dimensional.

We first make a try to simulate an originally cone-shaped distribution of 3-dimensional data points using an overlap of data sets obeying uniform distribution. For the sake of good intuition and visualization, we make the data scale 500 and set the cone-vertex at origin. To generate the distribution, we performed 50 iterations and generated 10 points as x_i ($0 \leq i \leq 10$) obeying uniform distribution each time. In j^{th} iteration, the following expression should be met

$$x_{ij} \sim U(0, 10 * j)$$

Thus, a cone-shaped data distribution can be obtained as iteration time goes up and the overlap is formed. Similarly, we constructed overlapping distributions of normal distributions, binomial distributions, Poisson distributions, beta distributions, exponential distributions and some of their combinations within or between dimensions. The original data in cone-shaped is shown in Figure 7.

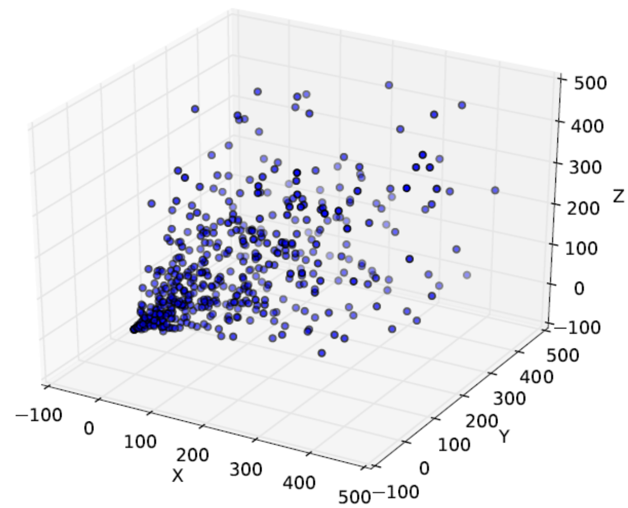


Fig. 7. Original Data in Cone-Shaped

B. Evaluation Metrics for Uniformity

Researches on uniformity measures of sample points in metric space are relatively sufficient so far, and we select two most common generalized metrics for mapping-uniformity evaluation in this part. The concept of gap ratio [2], [8] were first defined by Teramoto et al. to measure the uniformity of a finite point set sampled from S, which is a bounded subset of R2, and thereafter generalized by Arijit et al. in extension to overall metric spaces by appealing to covering and packing radius. motivated by combinatorial This measure is mainly inspired by approaches and applications in digital halftoning [3]–[6]. It defines the minimum gap as

$$r_P := \min_{p,q \in P, p \neq q} \frac{\delta(p,q)}{2}$$

Where (M, δ) is a metric space and P is a set of k points sampled from M. The maximum gap stands for the interrelation between the metric space M and $P(\subset M)$, the set sampled from M. It is defined as

$$R_P := \sup_{q \in M} \delta(q, P) \quad where \delta(q, P) := \min_{p \in P} \delta(q, p)$$

Then gap ratio for the point set P is defined as

$$GR_P := \frac{R_P}{r_P}$$

In our experiment, r_P can be confirmed by calculating the Euclidean distances between all data points, and R_P can be obtained by only considering the distances between candidate points and the sample points. Candidate points can be narrowed down to only the vertexes of the hypercube, because the following conditions are supposed to be met in a metric space

$$\begin{array}{l} d(x,y) \geq 0 \\ d(x,y) = 0 \Leftrightarrow x = y \\ d(x,y) = d(y,x) \\ d(x,z) \leq d(x,y) + d(y,z) \end{array} \tag{4}$$

Note that the more uniform the point sample, the lower will be the gap ratio. And the outcomes for different mapping solutions will be shown in next part.

The second metrics selected by us is known as discrepancy [7]. There are many variations of discrepancy. In our experiment, for a sample P of n points in a three-dimensional unit hypercube we consider the following quantity

$$D(P) = \sup_{x,y,z\in[0,1]} |xyz - \frac{|([0,x]\times[0,y])\cap P|}{n}|$$

This quantity above is called the star discrepancy, where the range space is the set of axis parallel rectangles anchored at $(0,0)$. Note that the data should be normalized beforehand. The more uniform the point sample, the lower will be the star discrepancy. Because the general idea of discrepancy measures is to measure the deviation from the "expected number of points" in various sizes/placements of similar objects.

C. Baseline for Comparison

Since our ultimate approach for uniformly-mapping is the modified casting-based method, which not only uses the arranged buckets and points-casting as approximation algorithms but the binary-translation as well, the previously-raised methods are used as baselines here to compare the results for uniformity measures (gap ratio and discrepancy). To be clear, our baselines include the directly-mapping method, sorting-based method and casting-based method, and we give the main formula here again.

- Direct-Mapping (DM):
  $v(d) = d$
- Sorting-Based Mapping (SBM):
  $v(d_k) = k/n$
- Casting-Based Mapping (CBM):
  $v(d) = \dfrac{\frac{c_{k+1}-c_k}{b_{k+1}-b_k}(d-b_k)+c_k}{n}$

And here we show our modified casting-based mapping (MCBM) again:

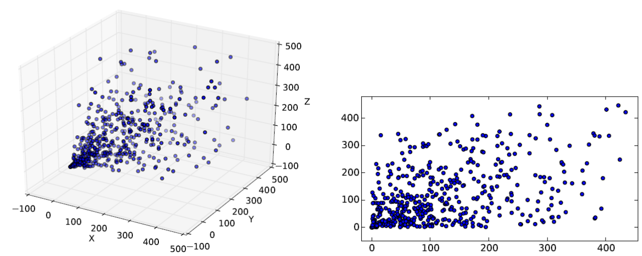
- Modified Casting-Based Mapping:
  $cor(d) = HEX_k(CBM_i(d))$

D. Experiment Results

In this part we give the average results of gap ratio and discrepancy in terms of DM, SBM, CBM and MCBM using data set in overlapping uniform distribution (cone-shaped) and overlapping normal distribution respectively. The corresponding value (average value in 10 times' experiments) includes the minimum gap, the maximum gap, gap ratio and discrepancy.

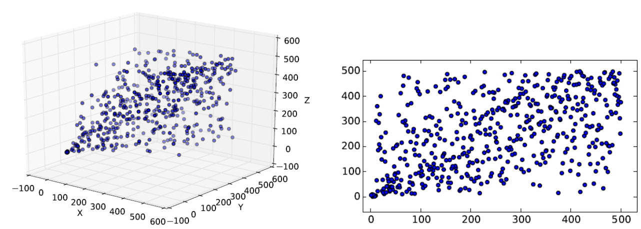
	Overlapping Uniform Distribution (cone-shaped)				Overlapping Normal Distribution			
	rp	Rp	Gap Ratio	Discrepancy	rp	Rp	Gap Ratio	Discrepancy
DM	0.0665	865.5078	13007.14	0.5248	1.3111	840.3373	640.9543	0.4080
SBM	1.2247	860.8304	702.87	0.0979	2.5495	850.4511	333.5743	0.0241
CBM	0.2283	864.2498	3784.28	0.1067	2.1956	855.9634	389.8527	0.0537
MCBM	4.0013	821.7348	205.43	0.0735	4.1059	866.0254	216.5064	0.0472

Also, Figure 8 to 11 show the corresponding outcome figures in three-dimension and two-dimension projection.



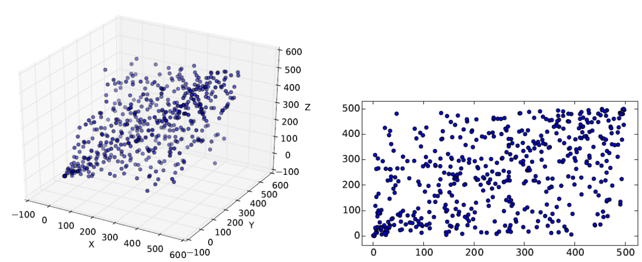
(a) 3-D Mapping (b) 2-D Projection

Fig. 8. Direct Mapping



(a) 3-D Mapping (b) 2-D Projection

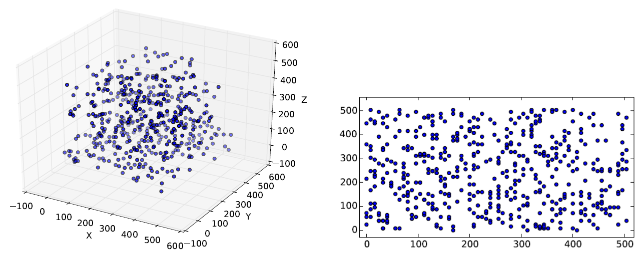
Fig. 9. Sorting-Based Mapping



(a) 3-D Mapping (b) 2-D Projection

Fig. 10. Casting-Based Mapping

From the outcomes , we can easily see the bad uniformity of Directly-Mapping (DM) because the servers around origin would be under the bulk of load and distant servers might be



(a) 3-D Mapping (b) 2-D Projection

Fig. 11. Modified Casting-Based Mapping

idle. Figure 9 and Figure 10 illustrate the result using Sorting-Based Mapping (SBM) and Casting-Based Mapping (CBM). It is obvious to see the improvement on mapping-uniformity through the comparison with the previous figure, however points tend to gather around the spatial line $x = y = z$ because of the high symmetry between dimensions, that is, the absolute or relative uniformity in each dimension can be obtained well but the uniformity of entire space with multiple dimensions performed badly. Fortunately, our ultimate method, Modified Casting-Based Mapping (MCBM) works well inter-dimensionally and reach a good result in both single dimension and multiple dimensions. The figure is shown in Figure 11.

VI. Conclusion

In order to manage large amount of data in data center networks, a distributed indexing scheme is adopted to improve the efficiency. In the indexing scheme, data publishment is of great significance due to the influence data distribution has on the efficiency of data retrieval. In this paper, we propose three data publishing schemes for skewed data set in BCube data center network. The basic scheme we present, sorting-based mapping scheme, maps a data point to its sorting number while it has low efficiency. The casting-based mapping scheme uses approximation algorithm to reduce the complexity of data retrieval. The modified casting-based mapping scheme appends another mapping to casting-based mapping and achieves better distribution results. In the evaluation, we define the metrics for distribution uniformity and compare the result and performance of an existing scheme and our schemes.

References

[1] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang and Songwu Lu, "BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers," SIGCOMM Spain, pp. 17-21, August 2009.

[2] S. Teramoto, T. Asano, N. Katoh, and B. Doerr, "Inserting Points Uniformly at Every Instance," IEICE (Institute Electronics, Information and Communication Engineers) TRANSACTIONS on Information and Systems, vol. 8, pp. 2348-2356, 2006.

[3] T. Asano, "Computational Geometric and Combinatorial Approaches to Digital Halftoning," Proceedings of the Twelfth Computing:The Australasian Theory Symposium, pp. 16-19, 2006.

[4] T. Asano, N. Katoh, K. Obokata, and T. Tokuyama, "Combinatorial and Geometric Problems Related to Digital Halftoning," Theoretical Foundations of Computer Vision, pp. 58-71, 2002.

[5] T. Asano, N. Katoh, K. Obokata, and T. Tokuyama, "Matrix Rounding under the L p-Discrepancy Measure and Its Application to Digital Halftoning," SIAM Journal Computing, vol. 32, pp. 1423-1435, 2003.

[6] K. Sadakane, N. Takki-Chebihi, and T. Tokuyama, "Discrepancy-Based Digital Halftoning: Automatic Evaluation and Optimization," Theoretical Foundations of Computer Vision, pp. 301-319, 2002.

[7] M. S. Ong, Y. C. Kuang, and M. P.-L. Ooi, "Discrepancy-Based Digital Halftoning: Automatic Evaluation and Optimization," Computational Statistics and Data Analysis, vol. 56, pp. 2159-2181, 2012.

[8] Bishnu Arijit, Desai Sameer, Ghosh Arijit, Goswami Mayank and Paul Subhabrata, "Uniformity of Point Samples in Metric Spaces Using Gap Ratio," Theory and Applications of Models of Computation: 12th Annual Conference (TAMC), Singapore, pp. 347-358, May 2015.