

Research Paper

A distributed information retrieval manner based on the statistic information for ubiquitous services

Takeshi TSUCHIYA¹, Hiroaki SAWANO², Marc LIHAN³, Hirokazu YOSHINAGA⁴, and Keiichi KOYANAGI⁵

^{1,2,3,4,5}Graduate School of Information, Production, and Systems, WASEDA University

²JSPS Research Fellow

ABSTRACT

We propose and clarify the manner in which a distributed information retrieval (IR) is conducted using a Peer-to-Peer (P2P) technology that has approximately the same accuracy as a current index-integrated search engine with the statistical information calculated from the contents. We need to solve the following three problems to develop distributed IR services that use statistic information. The first one is the manner of the calculation of the statistic information from all the contents and then generating the manner of the indices using the collaboration among distributed peers (nodes). The second problem is the efficiency of the index distribution on the logical space collaborated by the peers which is named the Multiple-Ring. The last problem is having efficient IR strategies that are highly accurate. We discuss the relationship between the propagation time for generating the statistics information and the system scalability by using simulations and emulations to evaluate the results. Our proposed system is suitable for flexible IR services that use the content environment of small to mid-scale networks such as those in an office.

KEYWORDS

Peer-to-Peer, distributed retrieval, overlay network, statistic information from contents

1 Introduction

Current information retrieval (IR) services that centralize the indices in the servers such as Google, collect the content information on the web servers by using crawlers. The generated indices, including content pointers, are managed in the centralized servers, and they provide the IR service. However, many kinds of services in ubiquitous network environments are provided by the peers (nodes), such as sensors and electronic home appliances. In addition, these retrieved objects have been changed according to the appearance and disappearance of service peers.

When using and providing the services and contents distributed over the Internet, these peers need to con-

struct their own logical data planes that are appropriately called overlay networks by using collaboration [1]. All of the services and contents provided by the collaborated peers are used on it, as shown in Fig. 1 [2] [3]. Therefore, the IR services focused on ubiquitous environments assume that the provided services and contents on the same logical plane are the retrieved objects. However, the indices of the centralized IR manner based on the crawler could not correspond to the continuous change caused by peer migration, so highly accurate retrieval would be highly unlikely. These overlay networks are independently constructed by the demand for a group and service attribute. Therefore, it is difficult for an IR service in the overlay network services to allocate a fixed peer-like server. It requires a flexible service applied to the service and content scalability, such as the ability to change an index peer manager on demand. We propose a distributed IR manner, and discuss the management collaboration with the peer

Received January 6, 2008; Revised November 28, 2008; Accepted December 1, 2008.

¹⁾ tsuchiya@suou.waseda.jp, ²⁾ sawano@suou.waseda.jp,

³⁾ lihan@fuji.waseda.jp, ⁴⁾ yoshinaga@akane.waseda.jp,

⁵⁾ keiichi.koyanagi@waseda.jp

DOI: 10.2201/NiiPi.2009.6.8

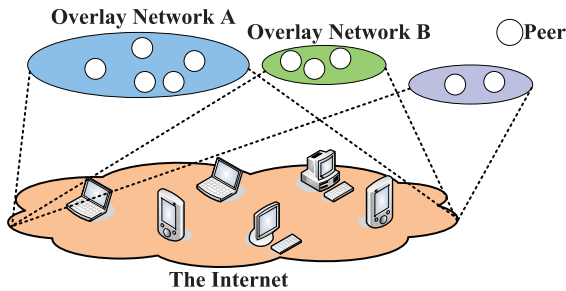


Fig. 1 Services on Overlay network environment.

participants, which could be one of the IR services used in ubiquitous environments.

The peers used to construct a P2P overlay network are classified with an address from the logical peer ID that is uniquely 128 bits. A join and leave of each peer occurs to the overlay network because of an on/off power operation by the user. Each peer is not an IP address that does not depend on the physical location, like the Internet. It has the basic function to routinely advertise its own peer ID. Therefore, providing a mobile function is a feature used as a basic function in a logical network. To create a model of a host in which a peer dynamically moves and on/off operations frequently occur, is efficient enough for use as a peer in a P2P network. The distributed retrieval discussed in this paper provides a service that can search for a targeted object from any of the peers. The indexes about the objects distributed in the peers are created and managed by collaboration between peers. Redistributed services are provided with the retrieval demand from any peer.

Some distributed IR services have recently been proposed, such as pVSM in pSearch [4] and a keyword set search [5], that are based on a logical infrastructure called the structured routings, such as Chord [6] and CAN [7]. These services are applied to the distributed hash table (DHT) for use as the key technology for indices management. However hashed meta-data, which are generated from the keywords of service information and the location pointer, do not contain the service details and are managed as the key indices for retrieval. Each user uses this service every time. It has to correspond to a hashed query sent by a user, and also has to retrieve the key indices. An unstructured routing service based on query flooding, such as Gnutella [8], is also capable of providing retrieval services. When getting the appropriate result from the above-mentioned services, the services do not take the statistical information for generating the key indices and queries into account. Therefore, these services require only the basic functions for the retrieval service using the similar-

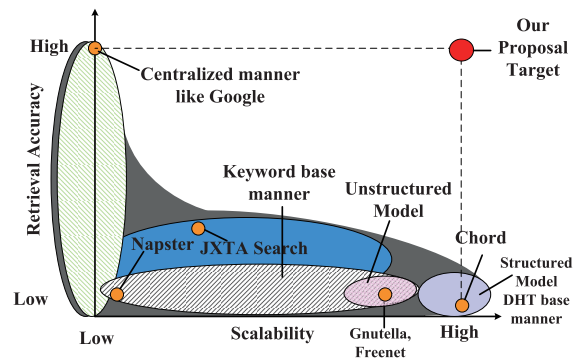


Fig. 2 Comparison between IR manners.

ity and partial content. The retrieval load with unstructured models proposed by Gnutella and Freenet [9] is bigger than that of a load with a constructed model in a retrieval service because all the peers send and receive traffic. Such unstructured models also are applied to retrieval services using partial matches with meta-information. Therefore, the retrieval quality is equal although the scalability of an unstructured model is lower than one of a constructed model.

Consequently, an accurate IR service is impossible in the ubiquitous environment. Fig. 2 shows the classification of the characteristics of our IR manner from the viewpoint of the scalability and retrieval accuracy. The distributed IR manners above have scalability, but the retrieval accuracy is not high. Therefore, we had to consider and clarify the distributed manner that includes a high scalability and retrieval accuracy.

We propose a distributed IR manner that uses the statistical information by the P2P (Peer-to-Peer) overlay network technology to decrease the retrieval accuracy in the distributed manner. However, all the indices that generated the statistical information have correlation between the semantic similarity and sequence. Therefore, we clarify the distribution of the indices while simultaneously maintaining its sequence. The accuracy of our distributed IR service is the same as that of a current centralized IR. However, communicating the periodical propagation among the peers that manage the indices is necessary for treating the statistical information. More importantly, the indices distribution manner generated with the content information has never been discussed for an unstable overlay network. We have already proposed and discussed several distributed functions and manners that could provide best-effort services in the references [1] [10]. Therefore, these achievements are taken to the following proposal manners. First, the handling manner is considered for indices in an unstable P2P user-participant network.

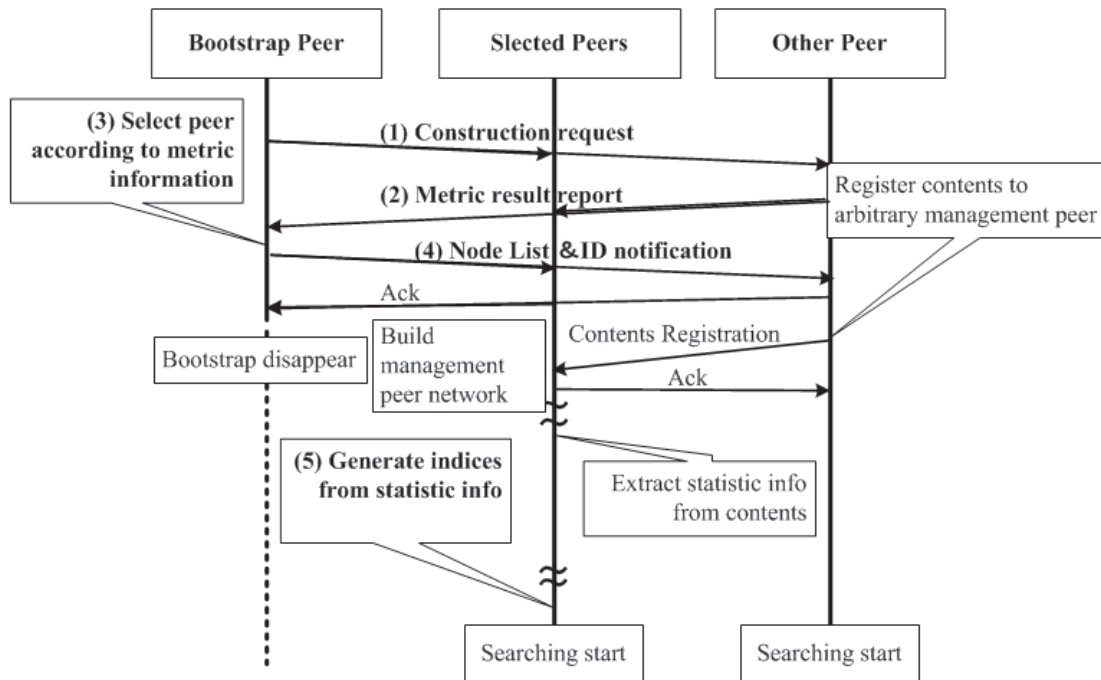


Fig. 3 Interaction sequence for indices generation.

Therefore, we discuss how to sequentially allocate distributed indices and also how to efficiently retrieve indices, which is called conical-searching. The relationship between the propagation time and peer manager scalability are described using simulations and an emulation in our evaluation. In addition, we conclude that our proposed manner contains the key factor for an IR service in a ubiquitous environment.

2 Distributed index generation

In this section the selection manner of peer managers is clarified using an estimation of the metrics and derivation of the statistical information. First, the model is defined for a retrieved object, which is called a simple overlay network. The managers of the indices are selected. The selection of the managers is utilized because it avoids the allocation of indices to unstable peers. As a result, the processing and network speed should be higher.

2.1 Peer selection

Peer managers are selected on a P2P overlay network. The following steps describe the protocol details where the step number corresponds to it in Fig. 3.

(1) A peer (Bootstrap) notifies the construction of a distributed IR service to all the neighboring peers on the same overlay network.

(2) Each peer who receives the notification responds by sending the number of retrieved objects by and the estimated result of some metric, such as the round trip time (RTT) to Bootstrap.

(3) Bootstrap calculates the received information, and selects the managers based on the metric estimation using an Adaptive Coordinator Election (ACE) platform [10].

(4) Each neighboring peer receives the results with the selected managers based on a list, and begins to compose the network based on the listed managers. The statistical information is calculated from the managed contents and is propagated.

(5) Each manager generates the indices of the contents, and distributes it to the network in a protocol sequence.

The specifications corresponding to the number are mentioned as follows.

(1) The initial process of the service is that Bootstrap sends construction requests to the neighboring peers in the overlay network. The requests ask the peers to perform and return metric reports.

(2) Each peer who receives a notification from Bootstrap answers with the results from a metric estimation and the number of retrieved objects. Bootstrap determines the dimension of the manager network based on these responses. In the latest version of

implementation, there are at least 10,000 indices managed by each peer.

- (3) Bootstrap decides on a manager based on the resulting metric estimations using the ACE platform. The metric evaluations of each peer are ordered based on a defined policy with a weighted unstable peer on an overlay network in the platform. The list based on the order is decided. In the latest implementation, the metrics for estimation adopt five independent metrics, such as the RTT, the processing capability, the throughput and connecting time to the network, and the time from service startup. Each peer executes a netperf [11], which is a benchmark software, to estimate the network environment and also executes a JavaBench [12] for the processing ability.
- (4) Bootstrap decides on a manager based on the evaluation order. It notifies each peer with the response in (2) to the peer that sent it in the list order of the selected results. The selected managers configure their IDs based on the notified list. Bootstrap relinquishes itself after the configuration is set up. The peers that were not selected, but hold services and contents, select a manager from the list for registering their service and contents to the network. A manager ID is given according to the estimated results in sequence. The current implementation has a 16-bit integer value. Each manager derives statistical information by using a morphological analysis and term frequency based on the registered services and contents. The managers propagate these statistical information in a chronological order of the manager IDs, and derives them from all the services and contents in the manager network.
- (5) Propagating the above-mentioned information, the TF-IDF method [13], and latent semantic indexing (LSI) [14], which degenerates the analyzed contents to a hundred dimension vector, are used for each manager. Then, the index is generated from the results.

The relationship between the number of managers and the propagating time on the manager network is discussed in Sec. 6. During the initialization of the management peer network and the generation of the indices, the IR service is required to propagate it at least twice. Then, a periodical propagation over the manager network is required for adding new contents, and confirming the existence of a manager. Another technique to reduce the cost and processing time is considered that uses a vocabulary corpus for the static contents. However, the corpus has a high possibility of being bias or separating the specification field from the managed contents, and this decreases the retrieval service accuracy. Since the accuracy is more important than the cost

and time, a periodical propagation of the information is used in the manager network.

2.2 Join and leaving

When indices are changed due to the migration of peers, the number of peer managers is adaptively changed, depending on the number of indices per peer. Then, the peer that has the smallest network ID is chosen as Bootstrap, and updates the manager list. When a new peer would like to join this service, this peer notifies Bootstrap to execute a manager selection. As mentioned above, Bootstrap updates the peer list according to estimation results. The new peer managers that receive a notification begin to share their management of the indices and periodically propagate their information.

On the other hand, when a peer would like to leave the network, it notifies the proximate peer managers, and terminates its connection. Despite the fact that it is difficult to predict that a manager might leave, this fact also causes a separation of the managers. At this moment, the retrieval service reconstructs the network. However, although the manager left, its data is still stored in the service by peers called back-up peers. The back-up peers are set at the beginning of the service in order to avoid such faults. The synchronizations of the indices and related information are periodically executed among each manager and the back-up peers on demand, in order to perform their functions as one group. Therefore, if a separation of the managers suddenly occurs, the service simply promotes one of back-up peers as the new peer manager. Having a back-up peer avoids any sudden termination of the service, and also saves time and memory from the migration of the indices. Since our proposal deals with multiple back-up peers from the beginning, it is readily able to detect any separation by a periodical propagation, and promotes the back-up peer to peer manager. Currently, the back-up peers are selected based on the node list accumulated by Bootstrap for the peer managers.

2.3 Management peer network scalability

The scalability of selecting a manager is discussed in the evaluation of our previous work [10], which clarifies that the number of Bootstraps increases as the waiting time increases for the metric estimation. However, an increase in the estimation time for the peer managers may cause the peers in the estimation to misunderstand the status of the manager selection, such as being in a deadlock state or that a system fault has occurred. Therefore, in the latest implementation a selection timeout is defined as 60 sec, and best-effort peer managers are selected by the estimation of the metrics in this term.

2.4 Indices generation

First of all, the peer managers extract catchwords from a target object using a morphological analysis when generating indices. The number of catchwords differs by the size of the object and field, and then it is necessary to delete the unnecessary and unused terms. This process does not only accomplish the required tasks, but also enhances the generation of the indices, such as a compression of the data field, and it also improves the calculation speed. The TF-IDF method [13] is used for the weighting of a derived catchword in our proposal. Each weighted value expresses the product of two independent functions. First, is the term frequency (TF), which is the rate of appearance per catchwords in an object [15], and it shows exhaustive characteristics. The other is the inverse document frequency (IDF), which means the inverse of the appearance rate of objects including the specific catchword [16], and it shows the specification characteristics derived from all the objects. Therefore, the products of TF and IDF give the appearance probability of each catchword.

The statistical information of the contents and services represents the times that each catchword appears in all the objects. The TF value $tf(t)_d$ in Eq. (1) is derived by the weighted value of catchword t in object d , and it uses a summation of the times for the derivation. The weighted value $idf(t)$ in Eq. (2) is calculated with the inverse of the appearance times $df(t)$ including catchword t in N objects, which means the number of all the target objects for retrieval. Therefore, the $tf \cdot idf(t)$ value in Eq. (3) is a product of the above-mentioned values based on the appearance probability of term t for all the catchwords.

$$tf(t)_d = \frac{tf(t, d)}{\sum_{s \in d} tf(s, d)} \quad (1)$$

$$idf(t) = \log \frac{N}{df(t)} + 1 \quad (2)$$

$$tf \cdot idf(t) = tf(t)_d \cdot idf(t) \quad (3)$$

Even the catchwords are automatically derived from the target objects, but their semantic relationships (e.g. an automobile car) are not taken into consideration. It is necessary to project each word using the semantic dictionary, which generalizes the relations of similar words, and to decrease the dimensions of the vector that shows the characteristics of an object. LSI [14] is applied in order to reduce the number of words including each object and to form the fixed vector size. As a result, the indices are composed of these vectors and the location pointers. The size of the vector dimensions that express their characteristics of an object must be from 100 to 300 dimensions [17]. In the following sec-

tions, the number of vector dimensions is fixed to 100 dimensions.

2.5 Characteristics of indices distribution

The probability to approximate each index vector dimension is high in a similar object in TF-IDF. Therefore, the sufficient retrieval accuracy is supported. The indices must be located to maintain the order of the vectors.

3 Indices allocation

In this section, we discuss the allocation manner of the indices generated in previous section, and also clarify the management manner.

3.1 Logical model

Several types of topologies can be considered for allocating indices composed by distributed peer managers, such as the star topology, ring topology, and fractal topology. A ring topology is used in this paper to avoid any heavy load centralization of a single peer, to have a manageable recovery from sudden changes in peers, and to easily cooperate in a logical space. Then, each index vector dimension is independently managed to avoid losing the relationship between the weighting values of the index. As a result, each user is able to start the IR service by dynamically indicating the most characteristic dimension of the index.

The logical space is formed based on vector space of the index generated by the cooperation between the managers, as shown in Fig. 4. An independent ring is constructed to each dimension of the index vectors in the logical space. The index is distributed and managed in the space based on the order of the vector values.

The evaluation order mentioned in subsection 2.1 is utilized as the ID for the construction of a logical

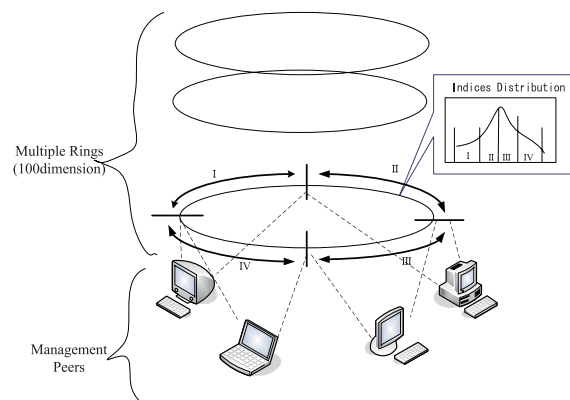


Fig. 4 Multiple Rings for index vector.

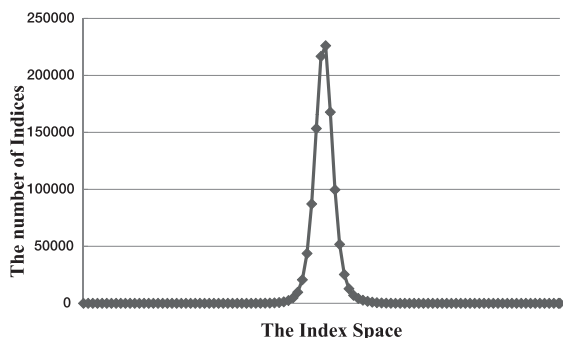


Fig. 5 Index distribution on given dimension.

model, and the ring model is constructed based on the values. Each manager notifies the ones with the neighboring IDs, and decides on a location in the logical space. The space of the index vectors is coordinated and managed. Each ring keeps the order of each dimension of the index vectors, and each index is divided into a hundred in each dimension. The location pointer information to the object and a pointer to the dimension of an index vector are held in the divided index. The list structure is used for a weighted value in a neighboring dimension, and it is able to retrieve the entire dimension.

3.2 Indices distribution

This research is applied to the test document NPL [18], which aggregates ones in an electronic system as objects for retrieval to generate indices. Fig. 5 shows the index distributions in some dimensions. The x -axis is the weighted value of an index, while the y -axis is the number of indices.

The weighted value of a word in Eq. (3) is calculated and composed of two functions. The weighted value of the object derived from Eq. (1) is the same in a specific vector dimension. The index distribution is decided by the appearance rate derived from Eq. (2) by a specific object. This is possible by using the central limit theorem [19], and a deviation of the distributed indices would be possible in a normal distribution. The distribution is assumed to be a normal distribution. It is expected that the concentration of the management load will be in a specific manager in a logical space. Therefore, it is important to discuss the avoidance of this centralization and to manage indices in sequence.

3.3 Linear conversion of indices and its distribution

The indices distribution manner is retained in its sequence based on the statistical characteristics, and avoids centralization.

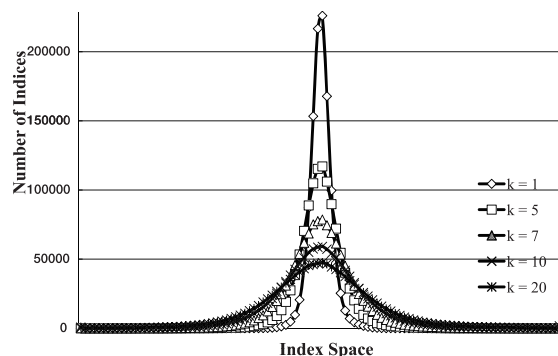


Fig. 6 Distribution of index with different k.

3.3.1 Conversion of indices value

Let the index value in a specific dimension be x_i for all $1 \leq i \leq n$ and then let the average of x_i be \hat{x} , while the standard deviation is σ .

$$\hat{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (4)$$

$$\begin{aligned} \sigma^2 &= \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x})^2 \\ &= \frac{1}{n} \sum_{i=1}^n x_i^2 - \hat{x}^2 \quad (\because (4)) \end{aligned} \quad (5)$$

For the value of \hat{x} , σ is statistically calculated based on the indices. The linear conversion of the weighted value of all the indices makes the statistical value linearly change. Modifying σ ensures the linear conversion of all the indices while maintaining the sequence. The constant value k decided when σ changes, is called a conversion constant. The distribution of the indices would range depending on k . The measurement l represents the length of a logical space and \hat{x}_i is the converted value of an index in a logical space. This helps clarify the change in distribution corresponding to the average of the index values and to the midpoint of the logical space.

Fig. 6 shows the range of index distribution by a range of k .

3.3.2 Characteristics of normal distribution and examine

In normal distribution, measured data states that 95.45% of all data are in the range between -2σ and $+\sigma$. Due to this characteristic the following equation is derived.

$$l = 4\sigma k \quad (6)$$

In Sec. 3.2, the distribution of the derived indices corresponds to the normal distribution based on the central

limit theorem. However, this theorem is only applicable for approximations and for examining the normal distribution use in [19]. Fig. 7 shows the normal distribution expected and the actual number of indices in a logical space. The distribution of the curve is not identical to the expected measurement, and cannot be approximated to the normal distribution expressed by a significant value, α ($0 < \alpha < 100$). Therefore, the conversion constant k in Eq. (6) cannot be automatically determined from the normal distribution.

When approximating the derived indices using a normal distribution, we first assume a provisional conversion of constant k to allocate the indices. Afterwards, the peer managers modify the converted constant k based on the distribution indices.

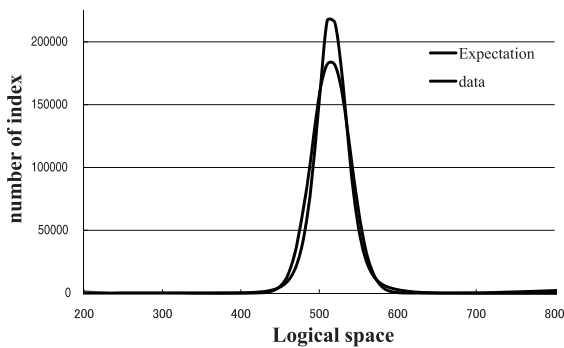


Fig. 7 Normal distribution and converted indices value.

3.3.3 Allocation to the logical space

We set the threshold value to 20,000 indices for each peer manager, and modify the value of k such that the number of indices will not exceed the threshold on each peer. Although the average index value corresponds to the midpoint in the logical space, our proposed method can cyclically allocate all the indices. The size of a logical space is equal to the length of a 10-bit long ring.

4 Conical search

After discussing the allocation of indices in a logical space in the previous sections, this section discusses an efficient retrieval method using a conical search.

4.1 Outline of conical searching

The proposed retrieval manner is called a Conical Search, which is illustrated in Fig. 8. This shows that a manager derives an l dimension ($0 < l \leq 100$) of vectors that shows the characteristics of an object based on the query inputted by a user. The retrieval starts from the value of the query on the n th ring with range d . [The retrieved objects in that range are considered candidates, and will be similarly calculated using a vector with an l dimension/Slightly unclear. Is this what you mean?]. The combined results are informed by a manager to the peer who sent the query.

4.2 Query

The user inputs an optimal query, and executes the above stated retrieval method. The peer (user) sends the query to the selected manager from the received peer

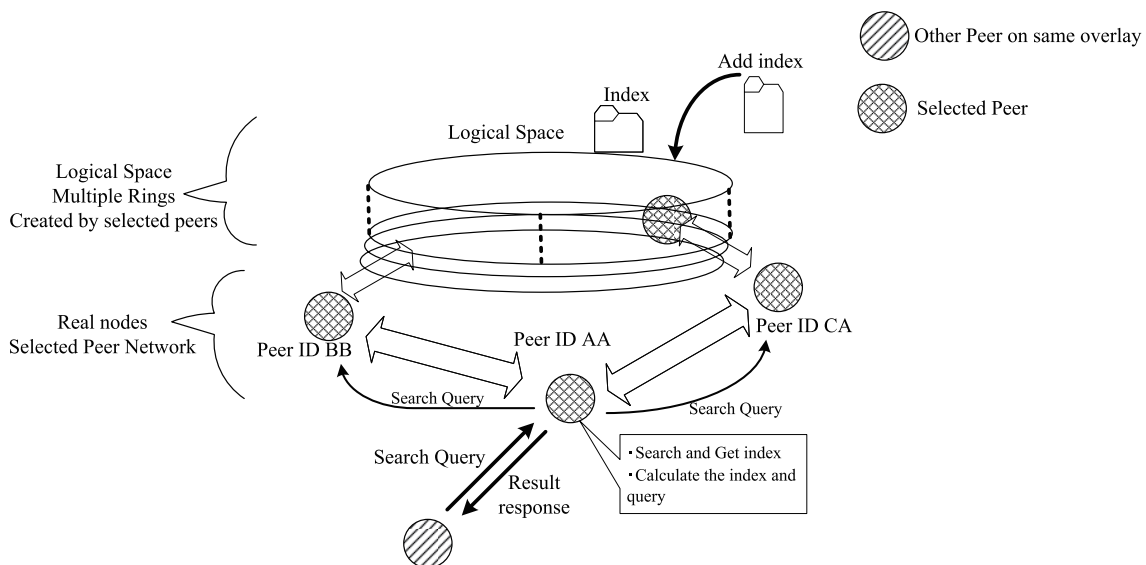


Fig. 8 Relationship of selected peers.

list. The selected manager who received the query derives the query vector in order to compare the vectors of objects, and start the retrieval.

4.3 Selection of rings

The number of dimensions is one less than the number of indices, so the managers who receive the query have to select the ring with the most characteristics for the retrieval. Since the object vector with the most characteristics has a large weighted value based on the TF-IDF method, the dimension of the query will also have a large value. In order to select the object with the most characteristics, the manager who receives the query has to select the largest dimensional value, and start the retrieval corresponding to a ring.

4.4 Retrieval in the range

Although the value of the indices in each dimension is kept allocated in a sequence, the manager, who received a query, can begin the retrieval from a vector value that has the most characteristics within the range d . The objects allocated in the range do not correspond to the query, but have the possibility of having a similar one because the retrieved ring is the same as the most characteristic dimension. So, the objects retrieved in the range are assumed to be candidates for the retrieval result. The relationship between range d and the retrieval accuracy is discussed through simulation in Sec. 5. The retrieval for the i -th ring (converted to a constant k_i) is the same as the retrieval in the dk_i area of the logical space composed of managers. Using a finger table, proposed by Chord [6], is an efficient retrieval service because the quantity of indices is calculated as only $O(\log N)$, where N is the number of rings. If the calculation of the quantity from its value in the range takes $O(n)$ time, then its amount for retrieval is $O(\log n + n) = O(n)$.

4.5 Comparison of indices and query

Although the proposed manner starts from the most characteristic value in the index, the candidate result for retrieval on the first ring is considered the most important since the values are similar. Therefore, our method does not execute the retrieval of other rings (dimension) except for the one with the most characteristics. It follows that the manager, who received the query, corrects all the values of the candidates in the dimension, and also compares the value's similarity to the query.

The quantity of vector dimensions in order to realize the efficiency and accuracy is discussed in Sec. 6. A decrease in the vector dimension lessens the loads of the peer managers and reduces the network traffic. Equation (7) shows how to calculate the similarity of the indices and queries. X and Y are the vector of the query

and index, respectively, while θ represents the similarity.

$$\cos \theta = \frac{X \cdot Y}{|X||Y|} \quad (7)$$

5 Simulation and emulation

This section evaluates our proposed manner through simulation and emulation.

5.1 Simulation environment

The simulation was constructed with a multiple-ring in a logical space with the defined specifications listed in Table 1. The object for retrieval uses the NPL [18] by Cornell, while SVDPACK [20] are used for the LSI. The constructed logical space has a length of 10-bit and collaboration of 512 managers.

5.1.1 Simulation scenario

In this simulation, 11,429 indices based on the source are retrieved using 100 queries derived from the same statistical information. Although the indices are fairly distributed in the 10-bit logical space, the managers are divided into a 10-bit logical space.

Registering 11,429 indices in a logical multiple ring resulted in 1,142,900 allocations (100 dimensions). Each index was separated by a dimension, and these indices contained 100 bytes of management information, which included an index ID, a pointer to an approximate manager, and a weighted value. The index distribution changed by applying the above-mentioned converted constant k in each dimension. All the k_i values in the ring were in the range of $8 \leq k_i \leq 14$, where i is the ring number.

We used following comparison with a related distributed retrieval system to evaluate our proposed system

- An ordinary centralized retrieval service with the environment listed in Table 1,
- A method that used a structured model.
- A retrieval service with an 11-points recall-precision curve with an unstructured model by Gnutella,
- A distribution system based on the precision-reachable curve.

5.2 Simulation result

Fig. 9 shows the result of the simulation. The x axis represents the number of dimension used for calculation in Eq. (7), while y axis represents the retrieval accuracy by the squeezing rate from the ring retrieval. The

Table 1 Specifications and environment for simulation.

Environment	Specifications
Machine	Intel Celeron 2GHz
RAM	384MByte
Corpus	npl, 11,429 contents, 100 Queries
Indexing	SMART & SVDPACK
Logical space	10 bit space
Dimension	100 rings
Chord	Overlay Weaver [21]
Gnutella	Neuro Grid [22]

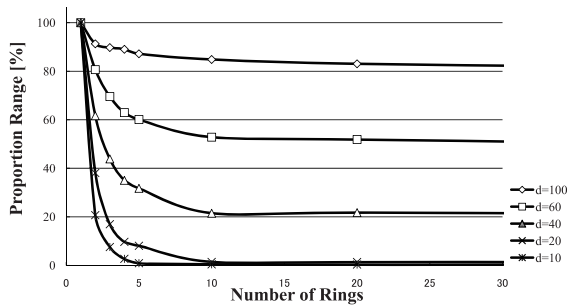


Fig. 9 Search Accuracy and Vector Dimensions.

parameter d means the retrieval range in logical space before conversion. If the range d is set to 100%, then all objects on the rings will be retrieved. However, if the range d is 40%, then there is a possibility to retrieve all over the ring with higher accuracy, because the actual retrieval range depends on the converted constant k . It also shows in the graph that the squeezed number of indices decreases as the dimension increases, and decreases by the inverse size of d . It is possible to consider the definition of similarity in this simulation caused by loose qualification such as $0.5 \leq \theta \leq 1$. Hence, as the ranges d for retrieval area increase, the similar objects also increase. Finally, Fig. 10 shows the efficiency of the squeezing method for the most characteristic ring.

In the case of comparing the query and indices using of vector information from 2 to 10 dimensions, the increase of vector dimension makes the retrieval accuracy decrease exponentially. However in the case of vector information with more than 10 dimensions, its increase does not affect the accuracy. The squeezing curve is asymptotic to the x -axis until the dimension is equal to 100%. The exponential decrease of retrieval accuracy mentions the decrease of the noise caused by increase of the object information. So comparing the similarity is necessary to get the characteristic vector information of more than 10 dimensions which have no the relation

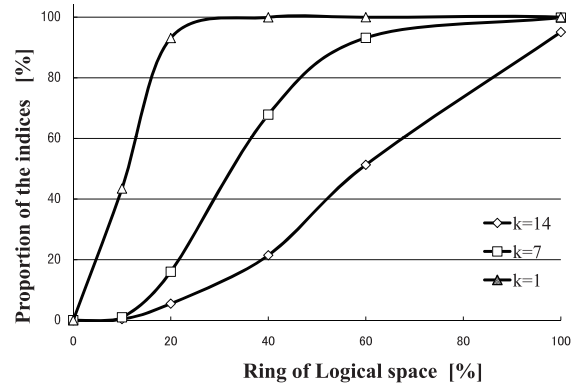


Fig. 10 Index distribution to Logical space.

to the value of range d . On the other hand, if the query has more vector information than the size of 10 dimensions, the retrieval accuracy will not greatly differ from having only 10 dimensions. Therefore, it is necessary to have an efficient and accurate distributed retrieval system and also the retrieval query should be capable of at least 10 dimensions of vector information. When the range d for retrieval equals to 100%, the accuracy of conical search corresponds to the current centralized retrieval system.

The retrieval of a ring from an index value in range d does not necessarily satisfy the accuracy of similar evaluations. Although the candidate results include the objects in the query and similar objects with high similarity, it is necessary for a similar evaluation to be calculated based on Eq. (7) using different dimensions. However, the indices, which are allocated outside of the range i , but on the same ring, will also have the same effect due to the relevance among the indices. The targeted objects for retrieval are not only the documents that have a lot of text and letters, but also even the network services and sensors that contain a small amount of text. So, these services cannot take complete advantage of the statistical information. Therefore, we need to take into consideration the similarity of the indices. As a result, the retrieval of a ring for similar objects corresponds to the increment of the candidate index, so the range d must be increased as much as possible.

Fig. 10 shows the range of indices distribution on the logical space by conversion constant k . The distribution curve, $k = 1$, means index distribution without linear conversion (the raw data of indices), $k = 7$, means that it was linearly converted by a minimum, and $k = 14$, means the ideal and suitable conversion. The curve $k = 1$ shows centralization of indices in only 20% of logical space, while the curve $k = 7$ shows an increment of indices exponentially caused by bias of indices.

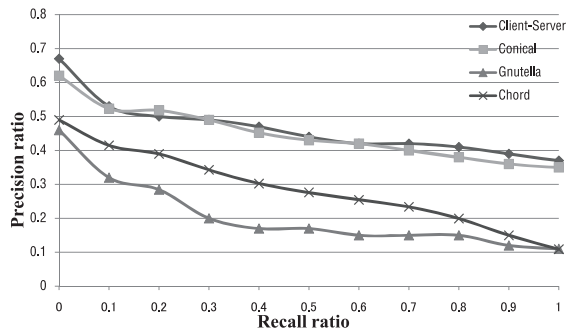


Fig. 11 11-points recall-precision curves.

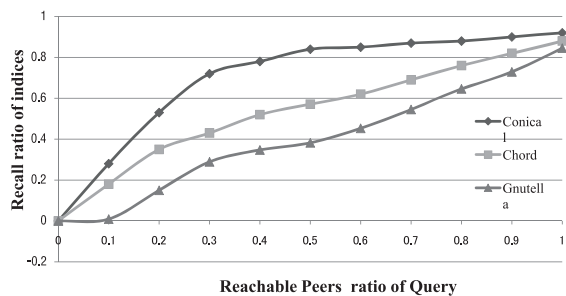


Fig. 12 Recall-reachable curves.

Lastly, the case for the most suitable curve, $k = 14$, is analogically determined based on the normal distribution and the propagation of management information among peer managers. This shows that distribution of indices is equilibrated by the linear increment of indices.

Fig. 11 shows the relationships between the 11-points recall-precision curves in each distributed retrieval environment. The retrieval quantity of the proposed method including the statistical information of the target object is the same as that with an ordinary centralized retrieval service. However, our quantity is not that much better than the ones by the distribution methods using the keyword-type structured model based on word feature with an unstructured model. The efficiency of the packet propagation is lower in the unstructured model by Gnutella than the one in the structured model by Chord.

Fig. 12 shows the relationship between the recall ratios of the index and that reachable ratio to distributed managers. A higher recall ratio is obtained with the proposed structured models and the Chord method than that with the unstructured model by Gnutella when there is a low reachable ratio of managers. It can reach a manager that holds the target index, and the number of the hops is low according to the retrieval query in dis-

Table 2 Specifications and environment of emulation.

Peer		Specifications
Odd	CPU	Intel Celeron 2.4 GHz
	RAM	1GByte
Even	CPU	Intel Pentium 4 2.8 GHz
	RAM	1GByte

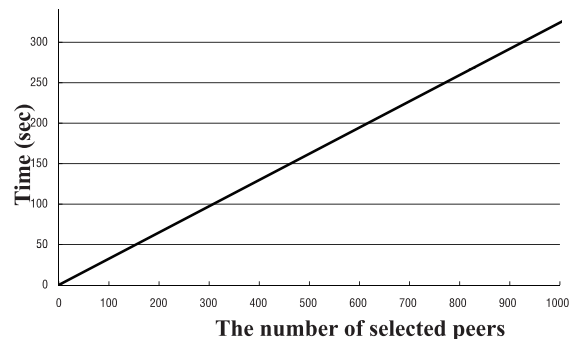


Fig. 13 Relationship between the number of selected peers and management time.

tributed index in the structured model. The efficiency of reaching a manager is better in the proposed model than that by Chord, because it uses not only the feature of the object, but also the statistical information.

5.3 Emulation

In the above-mentioned proposal, the periodical propagation of generating and distributing indices reduces the scalability of the peer distribution. In this regard, we have to discuss and clarify the scalability of the system by using emulation in a real environment while making some assumptions.

Table 2 shows the specifications of the emulation environment. Multiple rings composed of one hundred 10 bit rings are constructed using two PCs. In this emulator, we evaluate and clarify the relationship between the number of managers and the processing time for propagation over the management peer network.

5.4 Emulation result

Fig. 13 shows that the management time linearly increases with the amount of management caused by the periodical propagation of management information.

In order to improve the propagation, we considered using a finger table [6] to reduce its processing time. However, this propagation is executed sequentially based on the manager ID's, then cycles again because of the complex implementation. Although the order of the propagation time would not change, the divi-

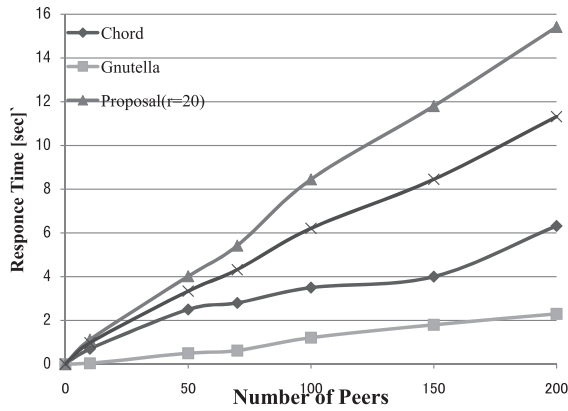


Fig. 14 Response time and peers.

sion management of a ring using different tokens would be improved for the propagation. The trend of the curve is about 0.32, so the management information is capable of propagating three managers per second.

Fig. 14 shows the relationship between the number of managers and the response time until the retrieval results are received. $r = 10, 20$ is a dimension with no retrieval query. Since there is an increase in the query, the number of dimensions of the index query make the increase a similarity, and the response is decreased until the retrieval results are received. The response is decreased more than in a retrieve by DHT, because plural rings are constructed in each index vector dimension in our method. The retrieval result by Gnutella is independently returned by each peer that receives the query, and sequentially obtaining the result from a neighboring of the query inputter is available.

Fig. 15 shows the relationship between the average numbers of messages that each manager sends per unit of time, and the number of managers. Chord mentions that the retrieval system can be constructed by using the constant number of messages through the number of managers since the number of messages decreases per unit manager. On the other hand, in our and the Gnutella methods, the available number of messages linearly increases according to the increase in the number of managers, because the average number of messages is not greatly changed although the number of managers increases. A sending message is not generated for anything except for the tree structure it manages in an unstructured model by Gnutella, and the number of messages is smaller than that in the proposed method although it is linearly increasing. However, the synchronization of statistic information is periodically generated according to the logical model between peers that constructs a ring model in our method, and

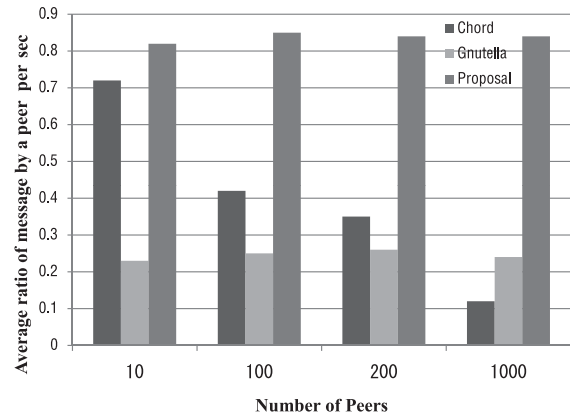


Fig. 15 Message sending ratio and the number of managers.

the number of messages increases in direct proportion to the number of managers. The appearance frequency and the object name are in each index word as communications traffic of the statistical information. The size of the information derived from npl utilized in emulation is 32 M[Byte] and a message with a lot of data, are sent by a manager, and the traffic cost is high.

6 Considerations

In this section, the evaluation of the proposal will be discussed based on the simulation and emulation of the previous sections.

6.1 From evaluation results

The proposed distributed retrieval manner is composed of a manager network and collaboration between peers. It can be used to conduct the registration and disappearance of any object by any user (peer) in the same overlay network. The indexing is executed by the periodical propagation of the management information. The data, generated from time to time by sensors or other devices, can be used to retrieve target objects as well as documents for use as statistical information. Therefore, this proposed manner is superior for services and objects modified in unpredictable situations, such as occurring and expiring peers. It is also used for ensuring the service stability of the time sequence of a periodical propagation. The explanation of the periodical propagation time among peer managers makes the characteristics of real-time services decrease. From the viewpoint of the retrieval accuracy, it happened as expected because the standard indexing used statistical information.

The results of the simulations showed that the retrieval accuracy was related to range d , and also to

the number of dimensions (rings). As a trade-off for both the scalability and increased management load of the peer distribution, the propagation information was used. From the simulations, it turned out that the retrieval accuracy is related to retrieval range d , and also the number of vector dimensions of the indices that were included. In real service situations, the parameters discussed in this paper should be determined by its peer scalability or policy. Therefore, these values should not be set to constant values. Instead, they should be treated as parameters.

The ring (dimension) for the retrieval was dynamically determined by the demands of the input query. Then, it took a long time to get the information from the other dimensions of the query, because the rings had to be processed one by one. However, when the query is composed of only a few words for the retrieval, these are thrown into the current retrieval system. [This situation might produce retrieval results that contain some noise that was caused by a lack of characteristic information. In order to solve this problem, according to the simulations, the input query must get results with an accuracy of at least 10 dimensions of the vector information. As shown in Fig. 11, the same accuracy can be supported as the retrieval service of a current centralized type system.

In comparison with ordinary structured models, the specific index can arrive by using statistical information. In addition, Fig. 14 shows the improvement of the response time by the decrease of the communication traffic, and a calculation for similarity and working memory when the vector dimensions of the query are decreased. Also, it shows the effect of restraining the trend of the graph in Fig. 13. However, the periodical propagation of the management information in each ring, always occurs without relation to the decrease in the vector dimensions for maintaining the service. This is the core factor for determining the peer distribution scalability and service scalability of the proposal manner.

6.2 Applicable situation

Fig. 13 shows that it is possible to manage and propagate the information for three managers per second on multiple logical rings. In addition, for the management time, it also took time to get the vector information, depending on the number of dimensions. In recent years, cases of using group collaboration services and file sharing services among distributed offices have increased, such as Groove [2]. The proposed manner is suitable for a flexible retrieval service that can be used in an office with a scale of up to ten managers in a network. The retrieval service can still be provided in a network even with the frequent migration of peers.

Thus, our proposal is suitable when there are only a few peer managers with the assurance that the same peers are constantly present. While Google provides scalable centralized IR services through the collaboration of servers that are composed of distributed file systems, our proposal allocates peer managers on demand, and uses peer managers to improve the flexibility and fault tolerance of the services.

7 Related works

In recent years, some IR services that use key indices based on DHT technologies without statistical information have been proposed, such as [5] and [23]. These systems constructed IR services based on structured infrastructures, such as Chord [6] and CAN [7] among others. Using hash functions has the advantage of equally distributing the key indices. When the user retrieves some words (queries), the mentioned system corrects the result indices according to each query, and selects a result with the common denominator. These services require the additional cost of loading the retrieval information and a network of infrastructure. In order to reduce costs, multiple key indices are allocated according to the contents or services. The paper [24] by M. Cai *et al.*, proposes the Locality Preserving Hash Function on the Chord, and it uses the key indices for multiple attributes.

The paper [26] describes a method for providing IR services using statistical information and all the managers share all the catchword information including statistics as a “PeerList”. However, this method does not clarify the adaptation of the PeerList for the unexpected changing of objects and updating of indices. Therefore, it is difficult to provide a retrieval service in a ubiquitous services environment. The paper [27] has discussed and proposed a method for improving retrieval quality in keyword-based retrieval services registered by the key-set combined with some catchwords, including the characteristics of an object. This method increases the possibility that the key-set corresponds to the query. However, this method is only intended for use with only a few characteristic catchwords of each object, and the quality of retrieval is not very high in comparison to our proposed system.

The current centralized IR services of Google have their own file system that distributes data in their indices servers [25]. It distributes data with up to 10,000 order nodes, and provides a retrieval service that is highly dependable and accurate. However, all of the retrieved objects are limited to static contents such as web contents. Therefore, this results in the periodical synchronization and duplication of data. In a retrieval service model, a retrieved target object is collected by crawling, so the only static contents that exist in a server

are the targets.

The object is not able to become the target with a possibility of being distributed in a P2P overlay network, and the scalability of the object is restrictive as a retrieval service. A share method by advertising objects whose user holds it in the network is proposed, and a distributed service is able to be retrieved by a user like in a ubiquities environment since a retrieval service is constructed in this environment. Therefore, the scalability of the object is high.

The pSearch [4] is the most related work to this paper. Two types of decentralized retrieval manners are proposed, and they are pVSM and pLSI. The pVSM uses the key indices based on DHT technology, while the pLSI uses semantic indices based on the statistical contents, which is similar to our proposal. In the pLSI manner, each index vector is converted into a Cartesian coordinate and is allocated on the spherical model constructed by node collaborations. When the user uses this service, the query is flooded with the value of the converted vectors, and similar contents to the vector value are also retrieved in the flood. The vectors of the indices are divided into each dimension as sub-vectors according to the characteristics. They are allocated in the same dimension of CAN. What are these 'it's in reference to corresponding to the dimension of the CAN. The query is also divided into sub-vectors, and it is used for retrieval according to the same dimension of CAN. From the viewpoint of the distribution of indices, pLSI adaptively adjusts the management range of each manager according to the distribution indices, and achieves a load balance within it. This manner has almost the same characteristics and efforts as our proposal. However, peer managers in this research have fixed partial logical space, and an adjusted conversion constant k for a load balance by each dimension. Thus, when managers use propagation in its network it reduces the cost for the calculation.

8 Conclusion

We proposed a distributed Information Retrieval (IR) manner by using the collaborations with statistical information, performed simulations, and evaluated it. First of all, there are two types of manners for indexing. One is the derivation of the statistic information from the registered contents, and the other is the generation of the indices based on such information. Secondly, we discussed the allocation manner of indices to a logical space that is constructed by the peer managers. The indices based on the statistical information are allocated in sequences, in order to enable all users to retrieve the contents in the similar manner. Finally, this paper has clarified the distributed retrieval manner called conical-searching for improving the logical

multiple rings accuracy. However, the statistical information must be periodically propagated for the indices management. In order to limit the cost of the periodical propagation of the statistical information made the scalability of the manager network liminary. As mentioned, this is not suitable for large-scale IR distributed systems such as the Internet. However, it is suitable for medium-scale overlay networks, which is an improvement from a small one based on systematical characteristics such as the flexibility by a peer estimation, and the accuracy. Therefore, we discussed the possibility of using realization in a group collaboration space or in an office. For future works, we must discuss the scalable management manner of peers, which would reduce the costs of propagation and query traffic.

Acknowledgement

This research was supported by "Ambient SoC Global COE Program of Waseda University" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

- [1] T. Tsuchiya, H. Yoshinaga, and K. Koyanagi, "Decentralized Information Retrieval Systems Based on Contents Information for Ubiquitous Services", *Proc. of workshop of IEEE 21st Int'l Conf. on Advanced Information Networking and Applications (AINA-07)*, pp.1013–1016, (May 2007).
- [2] Groove Networks, "<http://www.groove.net>".
- [3] Skype Networks, "<http://www.skype.com>".
- [4] C. Tang, Z. Xu, and M. Mahalingam, "pSearch: Information Retrieval in Structured Overlays", *ACM SIGCOMM Computer Communication Review*, pp.89–94, 2003.
- [5] O.D. Gnawali, "A Keyword-set Search System for Peer-to-Peer Networks", Master Thesis, MIT, (June 2002).
- [6] I. Stoica *et al.*, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", *ACM SIGCOMM'01* (Aug. 2001).
- [7] S. Ratnasamy *et al.*, "A Scalable Content-addressable Network", *ACM SIGCOMM'01* (Aug. 2001).
- [8] Gnutella, "<http://www.gnutella.com/>".
- [9] The Free Network Project, "<http://freenetproject.org/>".
- [10] H. Yoshinaga, T. Tsuchiya, and K. Koyanagi, "A Coordinator Election Using the Object Model in P2P Networks", *The Third Int'l Workshop on Agents and Peer-to-Peer Computing*, New York City, pp.120–131 (July 2004).
- [11] Netperf, "<http://www.netperf.org/netperf/NetperfPage.html>".
- [12] Java Benchmark, "<http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/bench/Bench.html>".

- [13] M. Berry, Z. Drmac, and E. Jessup, “Matrices, Vector Spaces, and Information Retrieval”, *SIAM Review*, vol.41, no.2, pp.335–362 (1999).
- [14] S. Deerwester *et al*, “Indexing by latent Semantic Analysis”, *J. of the American Society for Information Science*, vol.41, no.6, pp.391–407 (1999).
- [15] G. Salton and C. Buckley, “Term-weighting Approaches in Automatic Text Retrieval”, *Information Processing and Management*, vol.24, pp.513–523 (1988).
- [16] S. Jones, “A statistical Interpretation of Term Specificity an Its Application in Retrieval”, *J. of Documentation*, vol.28, no.1, pp.11–21 (1972).
- [17] K. Tokunaga, “Information Retrieval and Language Processing”, *Publishing on University of Tokyo*, ISBN4 – 13 – 065405 – 5 (written in Japanese).
- [18] NPL: “ftp://ftp.cs.cornell.edu/pub/smart/”.
- [19] T. Takemura, “Statistics”, *Kyoritsu Publishing*, ISBN4 – 320 – 01566 – 5 (written in Japanese).
- [20] SVDPACK, “http://www.netlib.org/svdpack”.
- [21] Overlay Weaver, “http://overlayweaver.sourceforge.net/”.
- [22] NeuroGrid, “http://www.neurogrid.net/”.
- [23] Y. Arakawa, H. Minami, H. Matsuo, H. Saito, “DHT Based Peer-to-Peer Searching System Using Pseudocandidate Key Indexing”, *IEICE(B)*, vol.J88-B, no.11, pp.2158–2170.
- [24] M. Cai, M. Frank, J. Chen, and P. Szekely, “MAAN: A Multi-Attribute Addressable Network for Grid Information Services”, *The Fourth Int’l Workshop on Grid Computing*, pp.184–191, Phoenix, Arizona, USA, (Nov. 2003).
- [25] S. Ghemawat, H. Gobioff, and S. Leung, “The Google File System”, *The 19th ACM Symposium on Operating Systems Principles*, Lake George, NY (Oct. 2003).
- [26] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer, “Improving Collection Selection with Overlap Awareness in P2P Search Engines”, *Proc’ of the 28th Int’l ACM SIGIR Conf’ on Research and development in information retrieval*, pp.67–74, Aug. 2005, Brazil.
- [27] S. Michel, M. Bender, N. Ntarmos, P. Triantafillou, G. Weikum, and C. Zimmer, “Discovering and Exploiting Keyword and Attribute-value Co-occurrences to Improve p2p Routing Indices”, *Pro’ of the 15th ACM Int’l Conf’ on Information and knowledge management*, pp.172–181, Nov. 2006, USA.



Takeshi TSUCHIYA

Takeshi TSUCHIYA received his B.S. and M.S. degrees from in 2001 and 2003 from Tokyo University of Science, Japan and Japan Advanced Institute Science and Technology. He currently is Ph.D student and RA of Ambient Global COE program at Waseda University. His research interests are in the area of distributed systems, distributed information retrieval, and P2P systems. He is a member of IEEE, IECE, and IPSJ.



Hiroaki SAWANO

Hiroaki SAWANO received his BE and ME degrees in computer science in 2004 and 2006 from Chubu University, Japan and Waseda University, Japan, respectively. Since 2006, he currently is a Ph.D candidate in the doctoral course at the Graduate School of Information, Production and Systems, Waseda University, Japan. His research interests are in the areas of computer vision, pattern recognition, augmented reality, and P2P network. He is a student member of ACM, IEICE Japan, and IPSJ.



Marc LIHAN

Marc LIHAN received his BE degree in computer science from Ateneo de Manila University, Philippines in 2006. He is a ME student Graduate School of Information, Production, and Systems, Waseda University, Japan. He research area includes indoor localization using wireless sensor networks, and mobile P2P networks. He is also a student member of ACM.



Hirokazu YOSHINAGA

Hirokazu YOSHINAGA received his BEc and ME degrees in computer science in 2000 and 2005 from Dokkyo University, Japan and Waseda University, Japan, respectively. He currently is a Ph.D student at the Graduate School of Information, Production and Systems, Waseda University, Japan. And he is founder of logly Inc. His research interests are in the areas of distributed storage systems, and P2P distributed systems. He is a student member of IEICE Japan, and IPSJ.

**Keiichi KOYANAGI**

Keiichi KOYANAGI received B.S. and M.S. degrees from Keio University, Tokyo, Japan, in 1975 and 1977, respectively, and a Ph.D degree from Osaka University, Osaka, Japan in 1997. Since he joined NTT in 1997, he has been engaged in R&D of wide range of digital switching systems. Since 2003, he has been a professor at Waseda University, Graduate School of Information, Production, and Systems. He is a senior member of IEEE, and member of IPSJ, ACM.