Vol.2018-DPS-174 No.29
Vol.2018-CSEC-80 No.29
2018/3/6

エッジコンピューティング基盤のための局所性を考慮したオブジェクトストアの提案

白石 裕輝[1,a)]　樫原 茂[1]　Doudou Fall[1]

概要：Multi-Access Edge Computing (MEC) や Fog Computing といったエッジコンピューティングと呼ばれるパラダイムは，リアルタイム性を必要とする Internet of Things (IoT) サービスの要件を満たすために設計された．低遅延が求められるリアルタイムサービスの実現には高速なストレージが必要であり，エッジコンピューティングにおけるオブジェクトストアの需要は極めて高い．本論文では，このようなエッジコンピューティング環境に適したオブジェクトストアを実現する手法を提案する．提案手法は，複数のエッジノードにより局所性を有する構造化オーバレイネットワークを構築し，サイト内で独立したオブジェクトストアを提供する．提案手法により構成されるオブジェクトストアのトラフィックの流れを考察することにより，提案手法が (1) オブジェクトが属するエッジノードを効率的に探索できること，(2) サービスやネットワークの分断における影響が局所的に抑えられることを示した．

キーワード：オブジェクトストア，エッジコンピューティング，Internet of Things (IoT)，構造化オーバレイネットワーク

A Proposal of a Proximity-Aware Object Store for Edge Computing Infrastructures

Youki Shiraishi[1,a)]　Shigeru Kashihara[1]　Doudou Fall[1]

1. はじめに

インターネットや電子機器の発達によって，コンピュータやスマートフォンだけでなく身のまわりのあらゆるデバイスがインターネットに接続されるようになった．多くのデバイスから収集される膨大なデータを分析することで生まれる新たな価値は，ビジネスに大きな影響をもたらしている．このような Internet of things (IoT) は今後もますます発展していくと考えられており，2020 年にはおよそ 200 億の IoT デバイスが使用されるとの予想もある [1]．Amazon Web Services や Microsoft Azure といったクラウドコンピューティング基盤は，限られた計算資源しか持たない IoT デバイスに対して豊富な計算資源を提供することができるため，IoT アプリケーションのプラットフォームとして広く使用されている．しかしながら，クラウドコンピューティング基盤には処理のリアルタイム性関する課題があり [2]，IoT アプリケーションはその真の価値を発揮することができない．Multi-access Edge Computing (MEC) [3] や Fog Computing [4] といった新たなコンピューティングパラダイムは，従来のクラウドコンピューティングの計算資源を地理的に分散したネットワークエッジに分散配置するものである．図 1 はエッジコンピューティング基盤の概略図である．これらのエッジコンピューティングと呼ばれる技術は，IoT アプリケーションに求められる低遅延などの要件を満たすことができるように設計されている [3], [4]．

エッジコンピューティングにおいて，IoT アプリケーションをサポートするためのストレージサービスを実現す

[1] 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology, 8916–5 Takayama, Ikoma, Nara 630–0192, Japan
[a)] shiraishi@computer.org

Vol.2018-DPS-174 No.29
Vol.2018-CSEC-80 No.29
2018/3/6

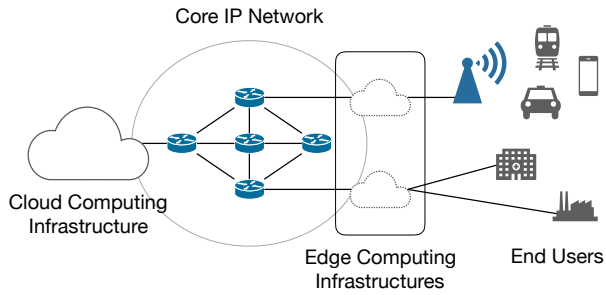


図 1 エッジコンピューティング基盤の概略図

るための研究が行われている [5], [6]．既存のオブジェクトストア [6] は，目的のオブジェクトが格納されているエッジノードを探索するためにさまざまな場所に位置する複数のエッジノードを経由する必要がある．オブジェクトを取得する際の遅延が大きくなる，一部のエッジノードの故障が他の無関係なエッジノードに影響を及ぼすといった問題点がある．

本論文では，エッジコンピューティング基盤のためのオブジェクトストアの構成手法を提案する．提案手法に基づくオブジェクトストアでは，オブジェクトが属するエッジの位置を表すドメインという概念を導入し，オブジェクトの探索を局所的に行う．これにより，オブジェクトを取得する際のトラフィックが最適化され，また，エッジノードの故障の影響範囲をドメイン内のみに抑えることができる．

2. 関連研究

2.1 エッジコンピューティング基盤のための分散ストレージシステム

エッジコンピューティング基盤のためのストレージシステムが満たすべき性質として，以下の 5 つが挙げられている [5]．

- 高速にアクセスが可能であること
- サイト間のネットワークが制御されていること
- サービスやネットワークの分断時であってもローカルサイトのストレージにアクセスが可能であること
- ユーザーのモビリティのサポートされていること
- サイトやユーザー，オブジェクトの数に対してスケールすること

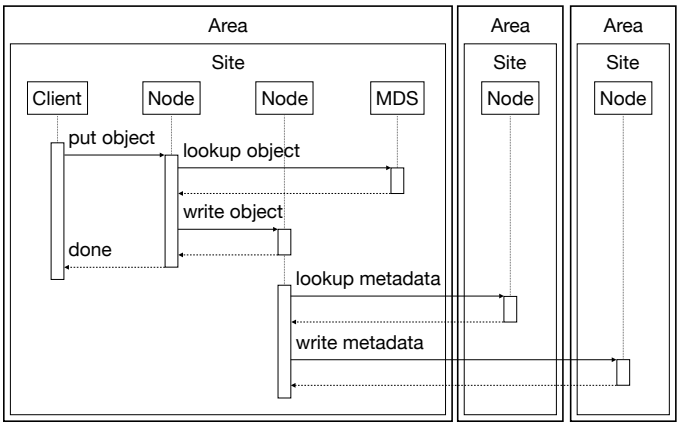
Confais らは，Scale-Out Network Attached Storage (SONAS) と InterPlanetary File System (IPFS) を用いた分散オブジェクトストアを提案している [6]．同一サイト内の複数のストレージノードを SONAS で接続し，オブジェクトの格納場所を IPFS を用いて管理している．

オブジェクトの挿入，取得を行う際のトラフィックの流れは図 2 のようになる．図 2 (a) は，ローカルサイトのストレージノードにオブジェクトの格納を行う場合のトラフィックの流れである．クライアントは，ローカルサイトからストレージノードを一つ選択し，そのノードを介して SONAS にオブジェクトを格納する．このとき，クライアントにより選択されたノードは，SONAS のメタデータサーバー (MDS) にオブジェクトの格納先ノードの問い合わせを行う．また，クライアントにより選択されたノードは，オブジェクトの格納場所を IPFS に登録する．図 2 (b) は，ローカルサイトのストレージノードからオブジェクトの取得を行う場合のトラフィックの流れである．クライアントは，ローカルサイトからストレージノードを一つ選択し，そのノードを介して SONAS からオブジェクトを取得する．図 2 (c) は，リモートサイトのストレージノードからオブジェクトの取得を行う場合のトラフィックの流れである．クライアントは，ローカルサイトからストレージノードを一つ選択し，そのノードを介して目的のオブジェクトがローカルサイトの SONAS に存在しないことを確認する．次に，クライアントにより選択されたノードは，目的のオブジェクトに関するメタデータを IPFS から取得し，オブジェクトが存在するサイトのストレージノードに対してオブジェクトの取得のクエリを送信する．クエリを受信したノードは，図 2 (b) と同様の手順でオブジェクトを取得し，クライアントに結果を返す．クライアントにより選択されたノードは，取得したオブジェクトをキャッシュとしてローカルサイトの SONAS に保存する．

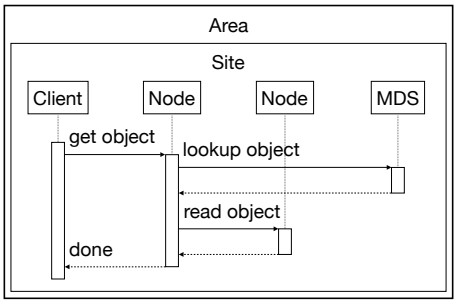
以上のようにオブジェクトストアを構成することによって，ローカルサイトのオブジェクトや一度アクセスしたリモートサイトのオブジェクトを低遅延で取得することができる．しかしながら，複数のサイトを分散ハッシュ表のフラットな識別子空間に並べているため，オブジェクトのメタデータを登録したり取得したりする際に，サイト間をまたぐ通信が多数発生し，大きな遅延が生じるという問題点がある．これと比較して，提案手法は，分散ハッシュ表を用いずエッジノードに与えられるドメイン名の階層構造に基づいて目的のオブジェクトを格納しているエッジノードを探索する．提案手法では，あるドメイン内から同一ドメイン内のオブジェクトを探索する際にサイト間をまたぐ通信が発生せず，また，他のドメイン内のオブジェクトを探索する際でもサイト間をまたぐ通信は必要最小限に抑えられる．そのため，オブジェクトの挿入や取得といった操作をより低遅延で実行可能であるという特徴がある．

2.2 構造化オーバレイネットワーク

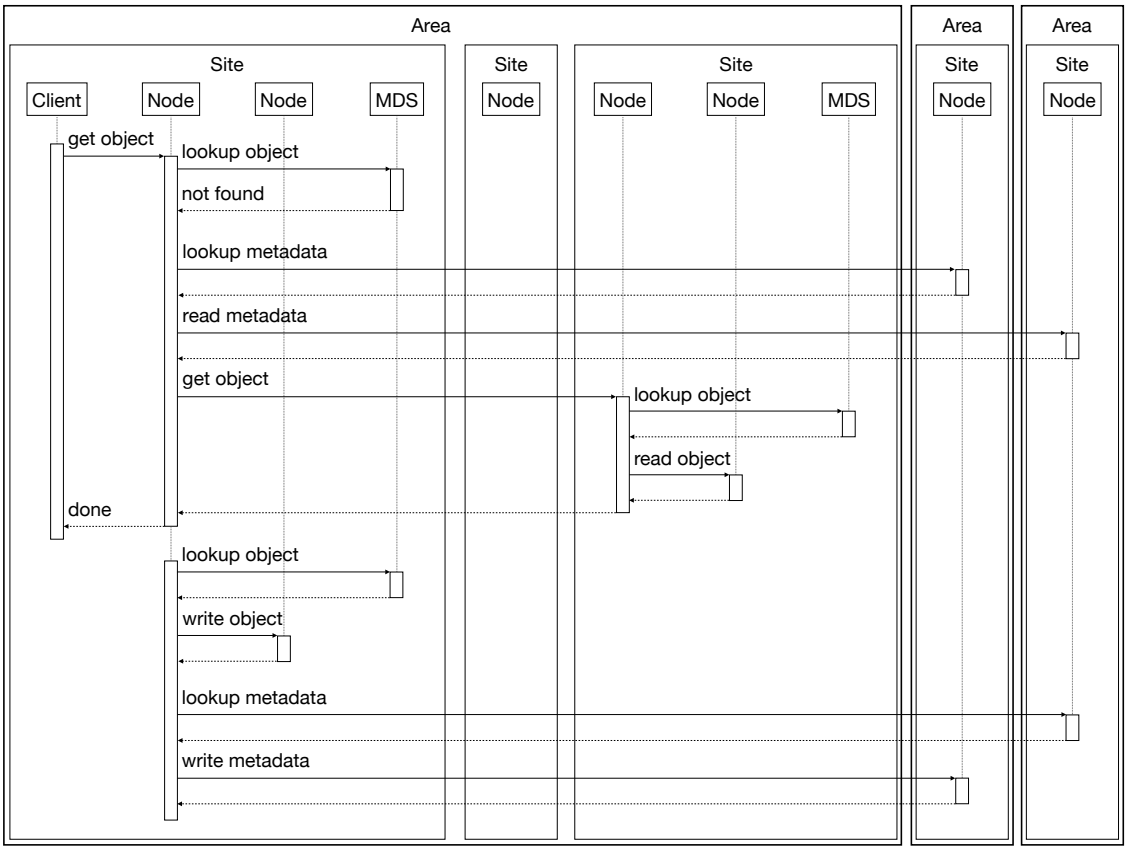
構造化オーバレイネットワークは，IP ネットワークなどの上位に構築される論理ネットワークである．分散ストレージシステムを構成するノードが自律分散的に構造化オーバレイネットワークを構築することで，耐障害性やスケーラビリティを向上させることができる．構造化オーバレイネットワークは，ノードに対して付与される ID に基づいてネットワークトポロジを決定する．提案されている



(a) ローカルエッジサイトのストレージノードへ
オブジェクトの挿入



(b) ローカルエッジサイトのストレージノードから
オブジェクトの取得



(c) リモートエッジサイトのストレージノードからオブジェクトの取得

図 2 Confais らのオブジェクトストア [6] におけるオブジェクトの挿入，取得を行う際のトラフィックの流れ

多くの構造化オーバレイネットワークは，ノード数に対してクエリの転送回数を対数オーダに抑えることができるという特徴がある．

分散ハッシュ表は，構造化オーバレイネットワークの応用のひとつで，Apache Cassandra や Amazon Simple Storage Service (S3)，InterPlanetary File System (IPFS) のベース技術となっている．提案されている代表的なアルゴリズムには，Chord [7]，Kademlia [8]，Pastry [9]，Tapestry [10] がある．分散ハッシュ表は，オーバレイネットワーク上でキーと値から構成されるオブジェクトの探索を可能にする．分散ハッシュ表は，ハッシュ値を利用してノードとキーを対応付けるが，これにより，キーの範囲探索といった複雑な探索クエリの処理は困難である．

一方，Skip Graph [11] や SkipNet [12] などのスキップリストに基づく構造化オーバレイネットワークは，ノード間の順序関係を維持するトポロジのネットワークを構築する．そのため，キーに基づいた範囲探索などの複雑な探索クエリが処理可能となる．

Vol.2018-DPS-174 No.29
Vol.2018-CSEC-80 No.29
2018/3/6

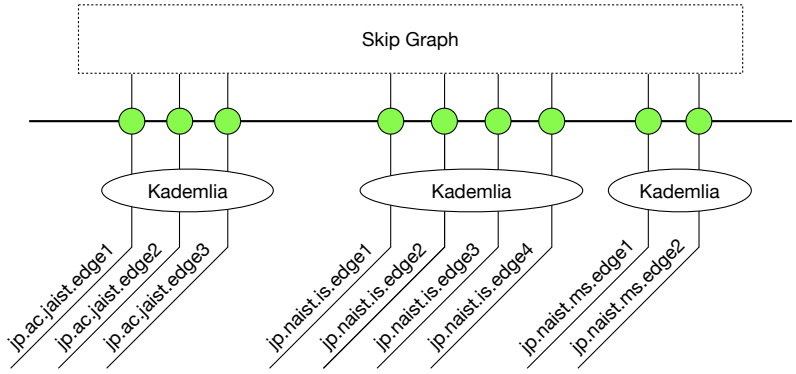


図 3 提案手法のアーキテクチャ

3. 提案手法

エッジコンピューティング基盤のためのオブジェクトストアの構成手法を提案する．図 3 は，提案手法のアーキテクチャである．提案手法では，各エッジノードに対してドメイン名を付与し，それに基づいて Skip Graph [11] を構築する．また，同一ドメイン配下のエッジノードは，それらでオブジェクトストアを提供する．これにより，クライアントが同一ドメイン内のオブジェクトを取得する場合，他のドメインに属するエッジノードと通信する必要がなくなる．他のドメインに属するオブジェクトを探索する場合でも，ドメインをまたぐ通信は必要最小限に抑えられる．

提案手法は，ドメインに基づくオブジェクトの操作をクライアントに提供する．クライアントは，ドメイン名とオブジェクト名を指定して (1) オブジェクトを格納する操作と (2) オブジェクトを取得する操作, (3) オブジェクトを削除する操作の 3 つの操作を行うことができる．例えば，ドメイン名を `jp.example`，オブジェクト名を `example-object` とした場合，クライアントは `jp.example/example-object` に対して (1)–(3) の操作を実行可能である．

以降では，提案手法のオブジェクトストアをどのように実現するかを詳細に述べる．

3.1 オブジェクトストアの構築手法

3.1.1 ドメイン名の付与

各エッジノードにドメイン名を付与する．ドメイン名は，ドットで区切られる以下の形式の文字列とする．

`first.second. ... .last`

同一ドメイン配下のエッジノードでオブジェクトストアを提供するために，上記の `last` の部分はエッジノードの識別子として使用することとし，ドメイン内でユニークな文字列とする．また，同一サイト内のエッジノードは，`last` を除いた部分文字列をドメイン名として共有していることとする．これは，サイトにドメインが対応することを意味する．複数のサイトを包含するサイトをエリアとする．エリアはドメインの階層構造に相当するものである．

ドメイン名の決定方法としては，

- 地理的位置情報に基づいて決定する方法
  例 `jp.nara.ikoma`, `jp.osaka.suita`, ..., etc.
- 組織構造に基づいて決定する方法
  例 `jp.naist.is`, `jp.naist.bs`, `jp.naist.ms`, `jp.ac.jaist`, ..., etc.

の 2 つが考えられる．

3.1.2 構造化オーバレイネットワークの構築

全サイトのすべてのエッジノードにより，前節で付与したドメイン名に基づく Skip Graph を構築する．このとき，エッジノード間の順序関係にはドメイン名の辞書式順序を採用する．辞書式順序とは，以下で定義される順序関係である．

定義 1 (辞書式順序)．全順序集合を Σ とする．直積集合 $\Sigma \times \Sigma$ 上の辞書式順序 $\leq_{\mathrm{dic}}$ は次のように定義される．

$$(a, b) \leq_{\mathrm{dic}} (a', b') \Leftrightarrow (a < a') \cup (a = a' \cap b \leq b'). \quad (1)$$

ここで，$a, b \in \Sigma$ とする．

アルファベットを $\Sigma_{\mathrm{mv}} = \{0, 1\}$ とする．$w_1, w_2, \ldots \in \Sigma_{\mathrm{mv}}$ としたとき，Σ_{mv} 上の語を $w = w_1 w_2 \ldots$ と表記する．また，語 w の長さを $|w|$ で表す．特に，$|w| = 0$ となる語 w を ε と書く．$|w| < \infty$ となる w 全体からなる集合を Σ^*_{mv} とし，$|w| = \infty$ となる w 全体からなる集合を $\Sigma^\omega_{\mathrm{mv}}$ とする．

ノード x は，いくつかの双方向連結リスト S_w に属する．ノード x が属する双方向連結リストは，ランダムなメンバーシップベクタ $\mathrm{mv}(x) \in \Sigma^\omega_{\mathrm{mv}}$ により決定される．双方向連結リスト S_w は，w が $\mathrm{mv}(x)$ の接頭語となるようなノード x がドメイン名の辞書式順序 $\leq_{\mathrm{dic}}$ の昇順で並んだリストである．S_ε はすべての要素 x が昇順で並んだリストとなる．以上の手順で双方向連結リストの族 $\{S_w\}$，すなわち Skip Graph を構築する．双方向連結リストのリンクは，オーバレイネットワークにおける辺を意味する．

Skip Graph を採用することによって，ドメインを探索する際のトラフィックが最適化され，必要最小限のドメイン間で探索が実現できるようになる．例えば，ドメイン

jp.naist.is からドメイン jp.naist.bs の探索を行う場合，探索は，ドメイン jp.naist の配下のエッジノードのみを経由して実現される．また，Skip Graph は，探索に必要なオーバレイネットワーク上のホップ数をノード数に対して対数オーダに抑えられるという特徴があるため，膨大なサイト数が広域に分散するエッジコンピューティング環境において Skip Graph の採用によるメリットは大きい．

3.1.3 オブジェクトストアの構成

同一ドメイン内のエッジノードをストレージノードとして分散ハッシュ表を構成し，サイト内のオブジェクトストアとして利用する．ここでは，実用性の観点から分散ハッシュ表として Kademlia [8] を採用する．クライアントが同一ドメイン内のオブジェクトを探索する場合，リクエストを受け取ったエッジノードは分散ハッシュ表を参照して対象のオブジェクトを探索する．クライアントが他のドメイン内のオブジェクトを探索する場合，リクエストを受け取ったエッジノードは Skip Graph を探索してドメインの探索を完了後，探索したドメインの分散ハッシュ表を参照して対象のオブジェクトを探索する．

4. 考察

オブジェクトの挿入や取得時のトラフィックの流れを観察し，提案手法について考察する．

図 4 は，提案手法により構成したオブジェクトストアを用いてオブジェクトの探索を行なったときのトラフィックの流れを示すシーケンス図である．図 2 (a) は，ローカルサイトのストレージノードにオブジェクトの格納を行う場合のトラフィックの流れである．クライアントは，ローカルサイトからストレージノードを一つ選択し，そのノードを介して分散ハッシュ表にオブジェクトを挿入する．図 4 (b) は，ローカルサイトのストレージノードからオブジェクトの取得を行う場合のトラフィックの流れである．クライアントは，ローカルサイトからストレージノードを一つ選択し，そのノードを介して分散ハッシュ表にアクセスしてオブジェクトを取得する．図 4 (c) は，リモートサイトのストレージノードからオブジェクトの取得を行う場合のトラフィックの流れである．クライアントは，ローカルサイトからストレージノードを一つ選択し，そのノードを介して Skip Graph にアクセスして指定されたドメインに属するストレージノードを探索する．次に，探索したストレージノードは，指定されたドメインの分散ハッシュ表から目的のオブジェクトを探索し，探索結果をクライアントに返す．

提案手法は，オブジェクトの格納場所を分散ハッシュ表を用いずドメイン名に基づいて管理している．以上からわかるとおり，ドメイン名に基づいた Skip Graph は，ドメインをまたぐトラフィックを必要最小限に抑えることができるという特徴がある．そのため，ネットワークの分断時やサービスの停止時であってもシステム全体への影響が局所的となる．

5. まとめ

本論文では，エッジコンピューティング基盤のためのオブジェクトストアの構成手法を提案した．提案手法により構成されるオブジェクトストアは，サイトに相当するドメインという概念に基づいて構造化オーバレイネットワークを構築することにより，サイト間をまたぐトラフィックを最適化する．これにより，オブジェクトの格納や取得にあたって，そのオブジェクトがどのエッジノード上に存在するかというメタデータを管理する必要がなくなる．他にも，構築される構造化オーバレイネットワークは，パスやオブジェクトに局所性を有するためサービスやネットワークの分断時におけるストレージサービスへの影響を局所的に抑えることができる．また，オブジェクトの操作を行う際のトラフィックのシーケンス図を作成し，提案手法が上述の性質を満たすことを確認した．

今後は，提案手法を実際に動作させ，性能の調査を行う予定である．具体的には，OpenStack 上にノードを複数台立ち上げてエッジコンピューティング環境を模した仮想的なネットワークを構築し，その環境の中でワークロードにより，オブジェクトの操作に関する I/O 性能や遅延時間を計測する予定である．

参考文献

[1] Gartner: Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016, Gartner (online), available from ⟨https://www.gartner.com/newsroom/id/3598917⟩ (accessed 2017-02-07).

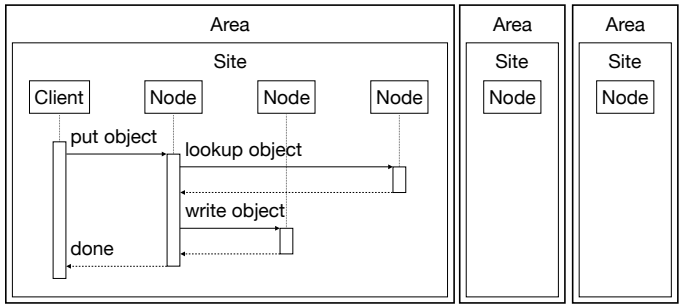
[2] Botta, A., de Donato, W., Persico, V. and Pescap, A.: Integration of Cloud computing and Internet of Things: A survey, *Future Generation Computer Systems*, Vol. 56, pp. 684 – 700 (online), DOI: https://doi.org/10.1016/j.future.2015.09.021 (2016).

[3] ETSI: ETSI - Multi-access Edge Computing, ETSI (online), available from ⟨http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing⟩ (accessed 2017-02-07).

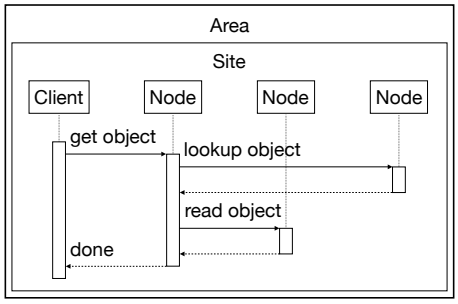
[4] Bonomi, F., Milito, R., Zhu, J. and Addepalli, S.: Fog Computing and Its Role in the Internet of Things, *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, New York, NY, USA, ACM, pp. 13–16 (online), DOI: 10.1145/2342509.2342513 (2012).

[5] Confais, B., Lebre, A. and Parrein, B.: Performance Analysis of Object Store Systems in a Fog/Edge Computing Infrastructures, *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 294–301 (online), DOI: 10.1109/CloudCom.2016.0055 (2016).

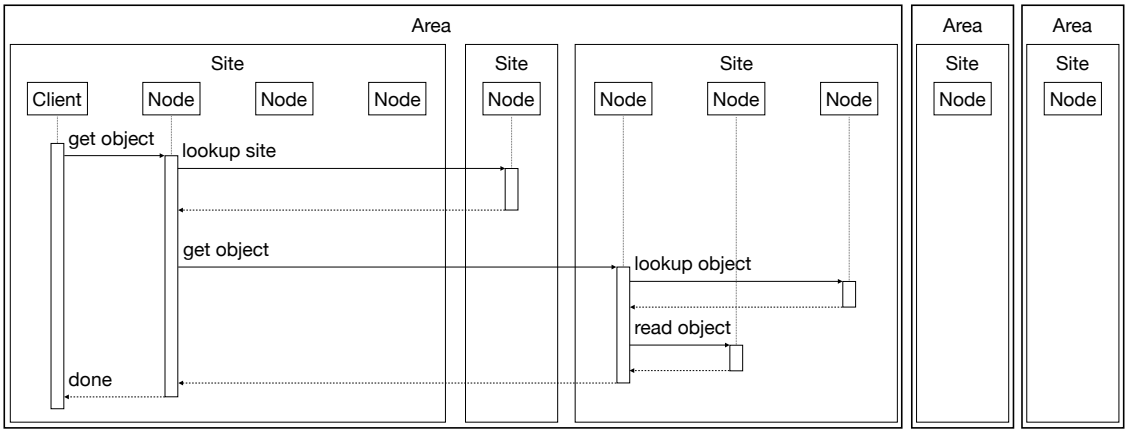
[6] Confais, B., Lebre, A. and Parrein, B.: An Object Store Service for a Fog/Edge Computing In-



(a) ローカルエッジサイトのストレージノードへ
オブジェクトの挿入



(b) ローカルエッジサイトのストレージノードから
オブジェクトの取得



(c) リモートエッジサイトのストレージノードからオブジェクトの取得

図 4 提案手法のオブジェクトストアにおけるオブジェクトの挿入・取得を行う際の
トラフィックの流れ

frastructure Based on IPFS and a Scale-Out NAS, *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pp. 41–50 (online), DOI: 10.1109/ICFEC.2017.13 (2017).

[7] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, *IEEE/ACM Trans. Netw.*, Vol. 11, No. 1, pp. 17–32 (online), DOI: 10.1109/TNET.2002.808407 (2003).

[8] Maymounkov, P. and Mazières, D.: Kademlia: A Peer-to-Peer Information System Based on the XOR Metric, *Peer-to-Peer Systems* (Druschel, P., Kaashoek, F. and Rowstron, A., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 53–65 (2002).

[9] Rowstron, A. and Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, *Middleware 2001* (Guerraoui, R., ed.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 329–350 (2001).

[10] Zhao, B. Y., Huang, L., Stribling, J., Rhea, S. C., Joseph, A. D. and Kubiatowicz, J. D.: Tapestry: a resilient global-scale overlay for service deployment, *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 1, pp. 41–53 (online), DOI: 10.1109/JSAC.2003.818784 (2004).

[11] Aspnes, J. and Shah, G.: Skip Graphs, *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics, pp. 384–393 (online), available from ⟨http://dl.acm.org/citation.cfm?id=644108.644170⟩ (2003).

[12] Harvey, N. J. A., Jones, M. B., Saroiu, S., Theimer, M. and Wolman, A.: SkipNet: A Scalable Overlay Network with Practical Locality Properties, *Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, USITS'03, Berkeley, CA, USA, USENIX Association, pp. 9–9 (online), available from ⟨http://dl.acm.org/citation.cfm?id=1251460.1251469⟩ (2003).