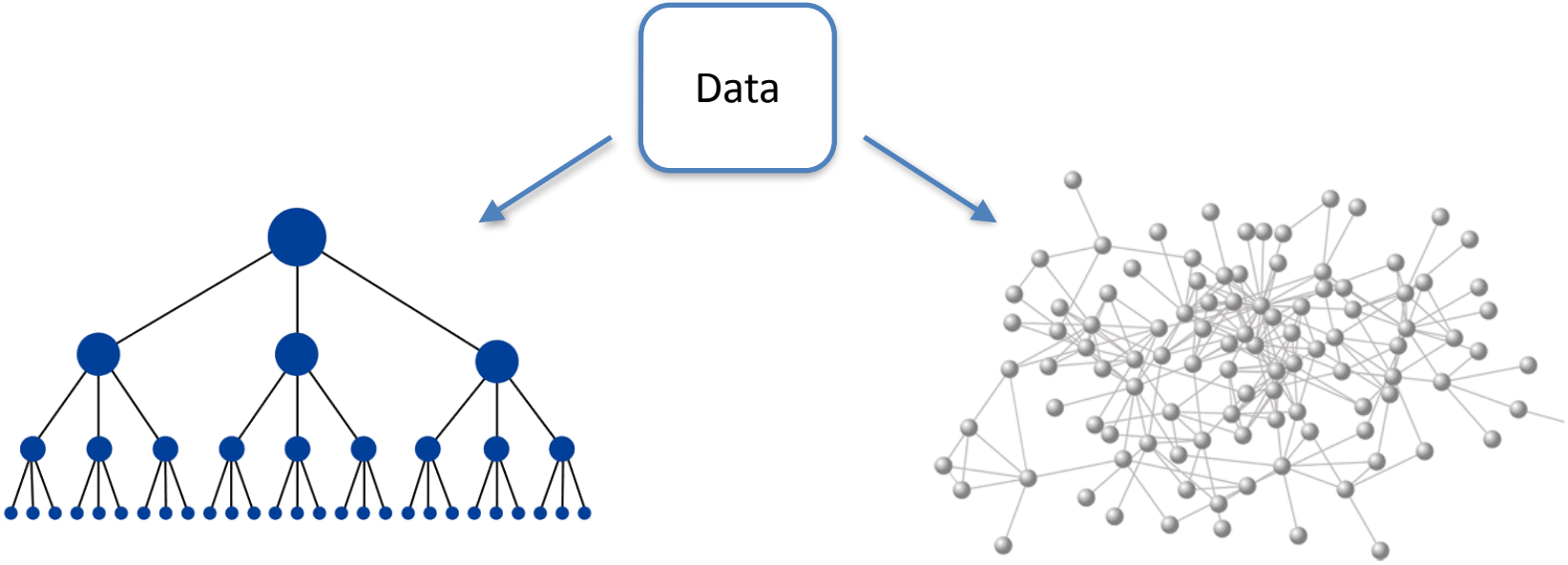
Algorithms for Building Highly Scalable Distributed Data Storages

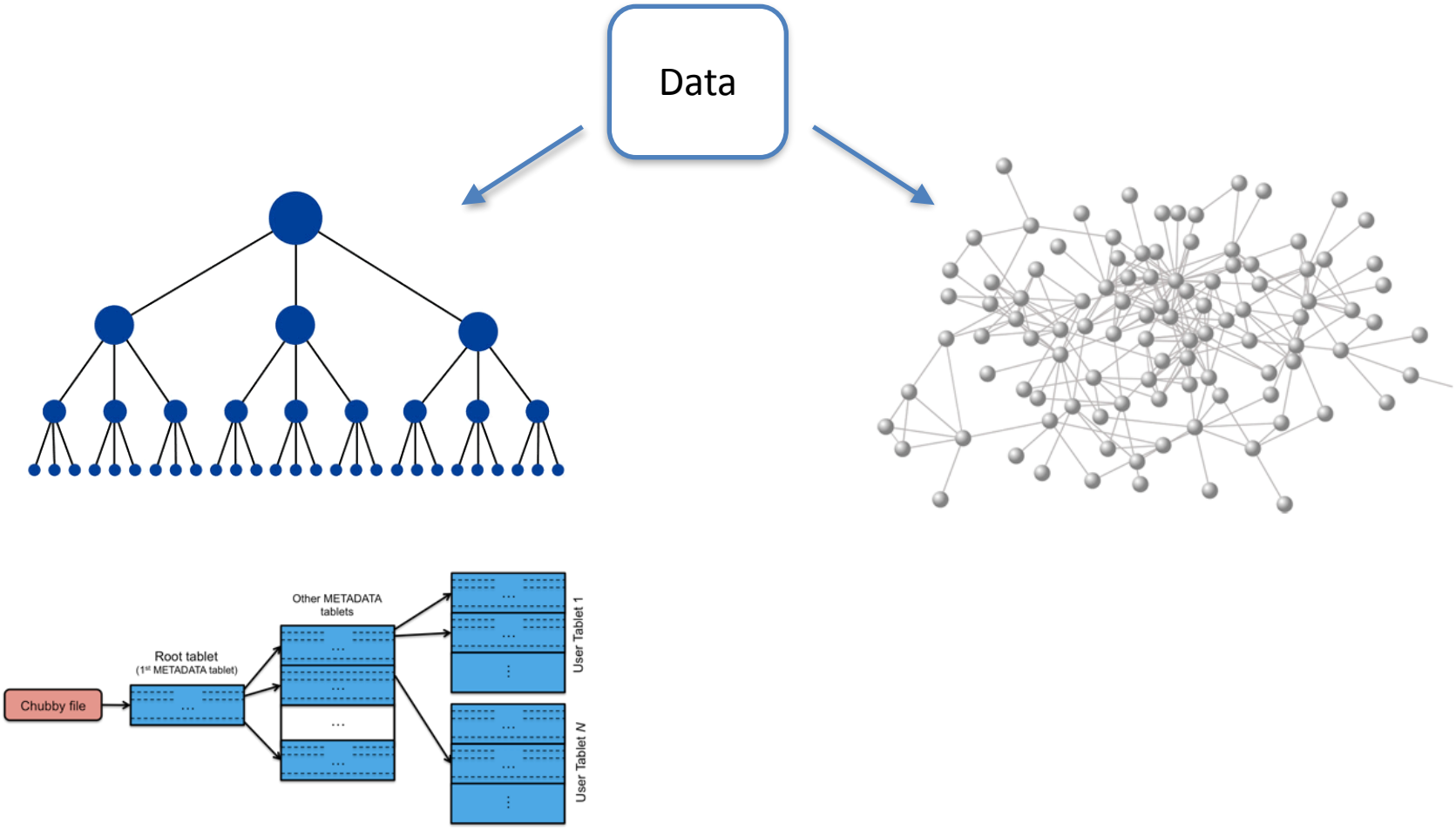
Alexander Ponomarenko

2nd International Scientific Conference “SCIENCE OF THE FUTURE”
September 20-23 2016, Kazan, Russia.

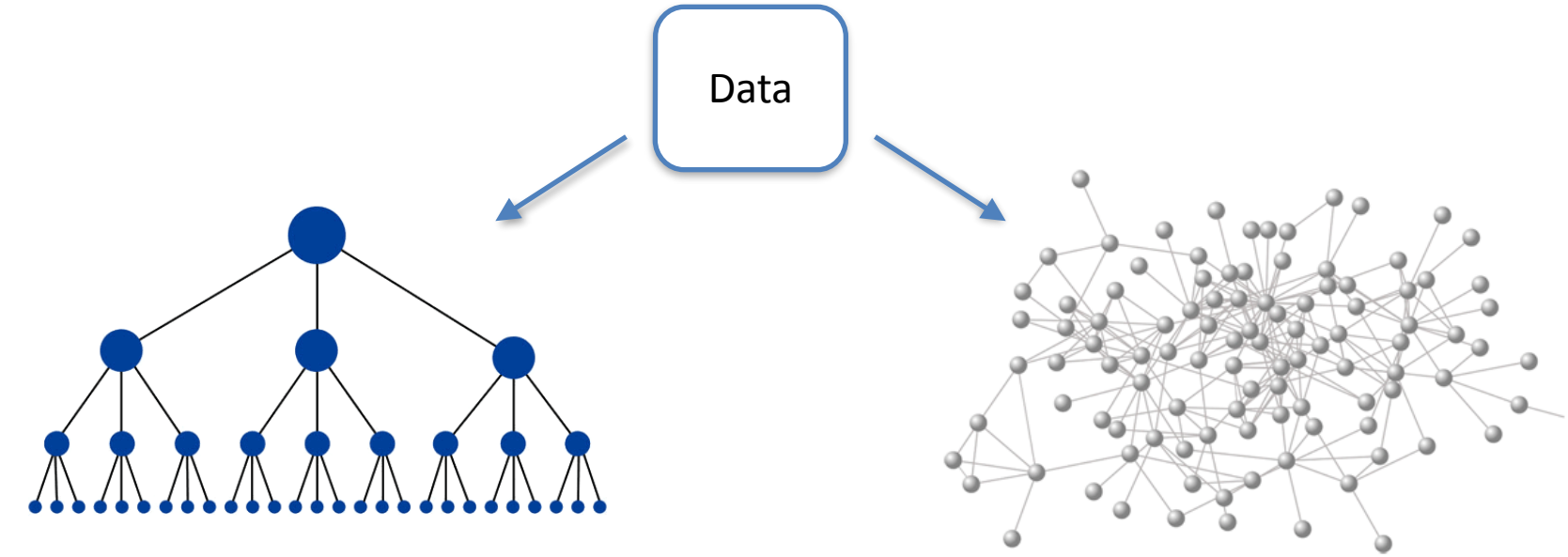
Hierarchy vs Heterarchy

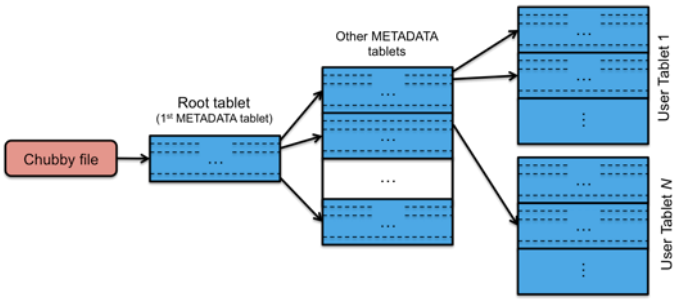


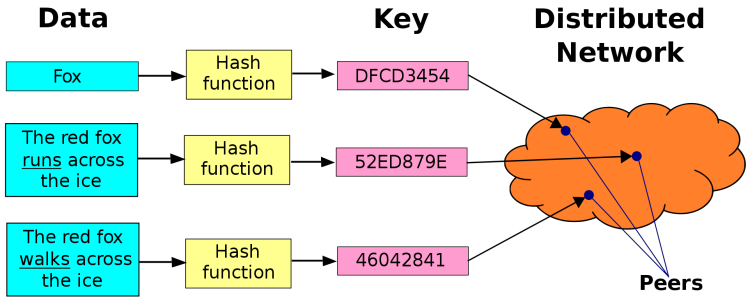
Hierarchy vs Heterarchy



Hierarchy vs Heterarchy

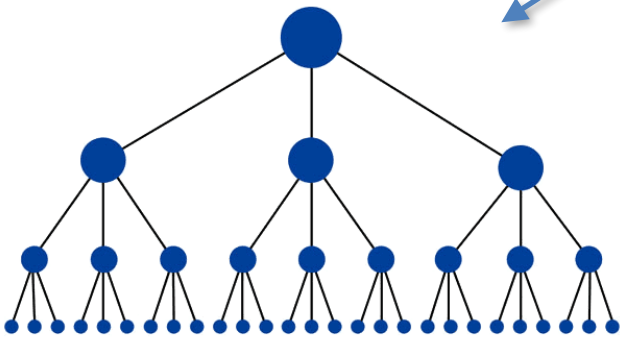


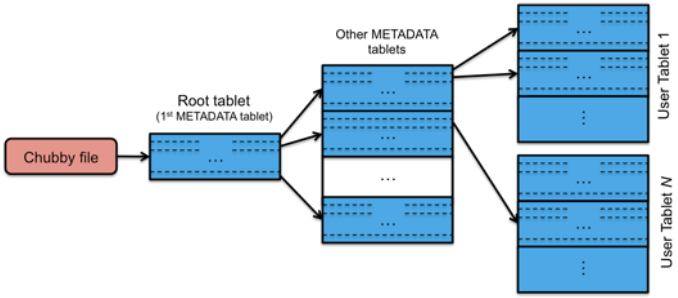


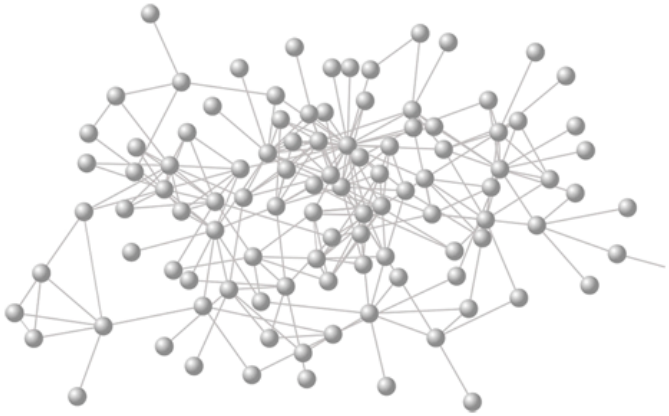


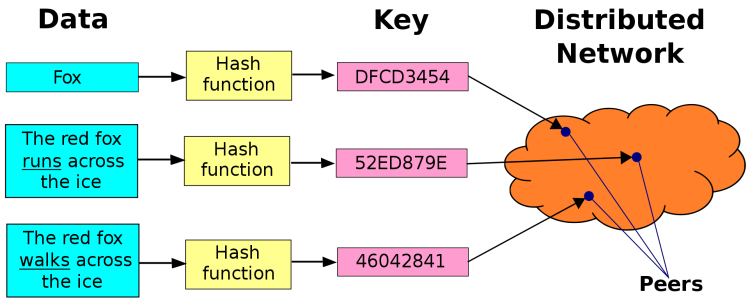
Hierarchy vs Heterarchy

Data









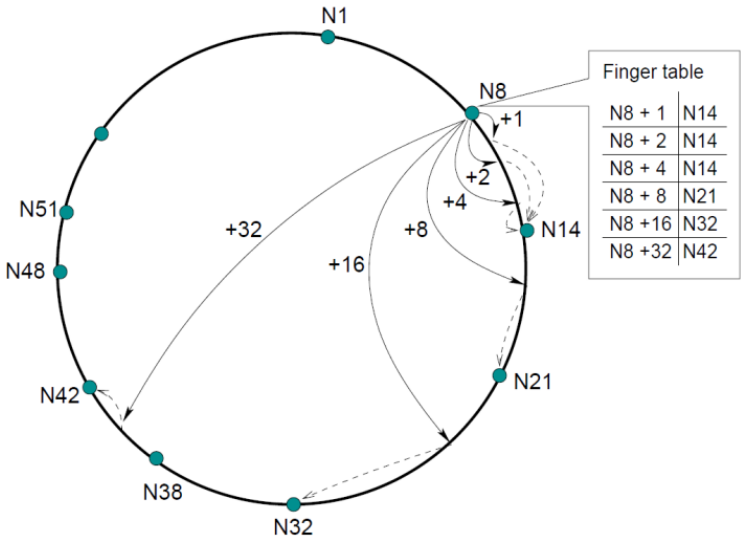
DHT protocols and implementations

- Aeropike
- Apache Cassandra
- BATON Overlay
- Mainline DHT - Standard DHT used by BitTorrent (based on Kademlia)
- CAN (Content Addressable Network)
- Chord
- Koorde
- Kademlia
- Pastry
- P-Grid
- Riak
- Tapestry
- TomP2P
- Voldemort

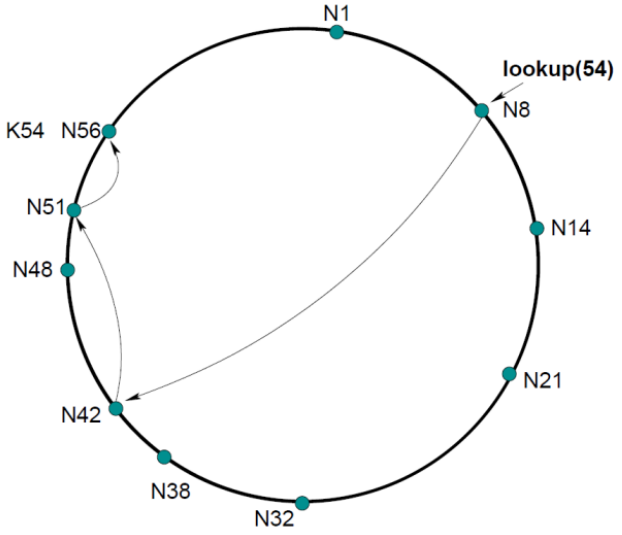
Applications employing DHTs

- BTDigg: BitTorrent DHT search engine
- cjdns: routing engine for mesh-based networks
- CloudSNAP: a decentralized web application deployment platform
- Codeen: web caching
- Coral Content Distribution Network
- FAROO: peer-to-peer Web search engine
- Freenet: a censorship-resistant anonymous network
- GlusterFS: a distributed file system used for storage virtualization
- GNUnet: Freenet-like distribution network including a DHT implementation
- Hazelcast: Open-source in-memory data grid
- I2P: An open-source anonymous peer-to-peer network.
- I2P-Bote: serverless secure anonymous e-mail.
- JXTA: open-source P2P platform
- Oracle Coherence: an in-memory data grid built on top of a Java DHT implementation
- Retroshare: a Friend-to-friend network[17]
- YaCy: a distributed search engine
- Tox: an instant messaging system intended to function as a Skype replacement
- Twister: a microblogging peer-to-peer platform
- Perfect Dark: a peer-to-peer file-sharing application from Japan

Structured Peer-to-Peer Networks: Chord Protocol



Routing table of node «N8»

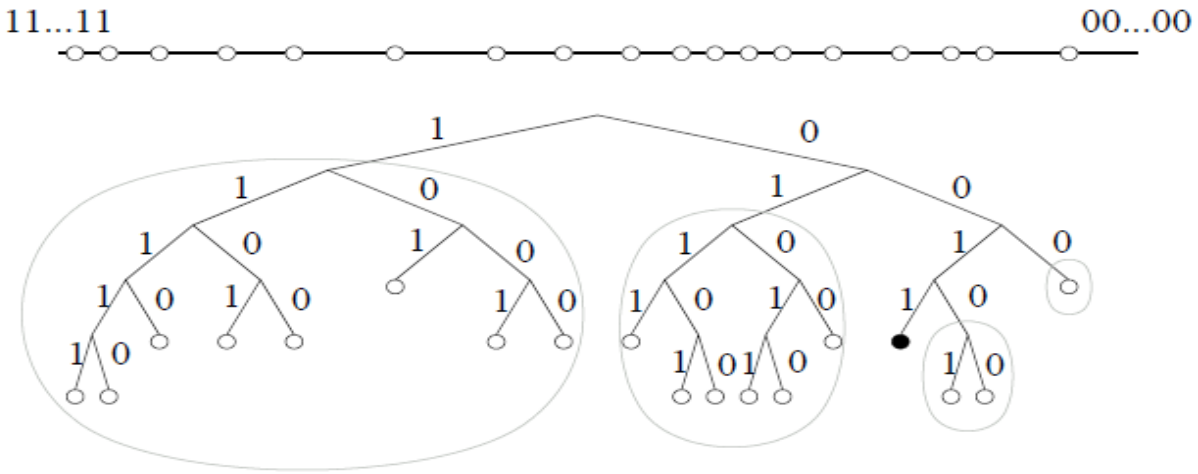


Searching of key 54 staring from «N8».

Distance function: $d(x,y) = (y - x) \mod 2^m$

Each node, *n*, maintains a routing table with (at most) *m* entries, called the *finger table*. The i-th entry in the table at node *n* contains the identity of the first node, s, that succeeds *n* by at least $2^{(i-1)}$ on the identifier circle, i.e., s = successor($n+2^{(i-1)}$), where $1 <= i <= m$

Structured Peer-to-Peer Networks: Kademlia



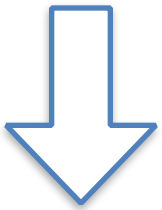
Identifier space of Kademlia

Distance function: *d(x,y)= x xor y*

Maymounkov P., Mazieres D. Kademlia: A peer-to-peer information system based on the xor metric // Peer-to-Peer Systems. – Springer Berlin Heidelberg, 2002. – С. 53-65.

to Overcome DHT Disadvantages

- DHT uses very simple distance functions
- Hashing destroys semantic of the data
- It’s hard to perform complex queries



Use nearest neighbour search in high dimensional metric space instead of exact search

Nearest Neighbor Search

Let D – domain

$d : D \times D \rightarrow R_{[0;+\infty)}$ - distance function which satisfies properties:

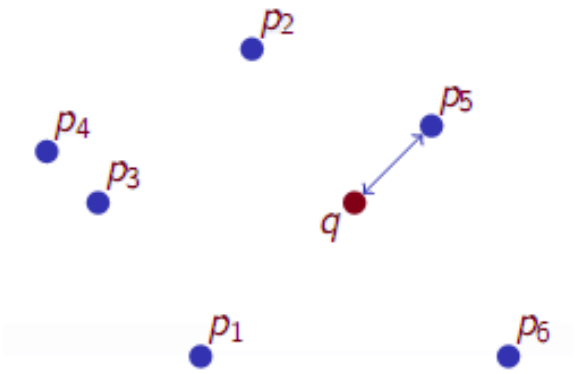
– strict positiveness: d(x, y) > 0 ⇔ x ≠ y,

– symmetry: d(x, y) = d(y, x),

– reflexivity: d(x, x) = 0,

– triangle inequality: d(x, y)+d(y, z) ≥ d(x, z).

Given a finite set $X = \{p_1,\ldots,p_n\}$ of n points in some metric space (D, d), need to build a data structure on X so that for a given query point $q \in D$ one can find a point $p \in X$ which minimizes $d(p, q)$ *with as few distance computations as possible*



Examples of Distance Functions

- L_p **Minkovski distance** (for vectors)
  - L_1 – city-block distance
  - L_2 – Euclidean distance
  - L_∞ – infinity

$$L_1(x,y) = \sum_{i=1}^{n} | x_i - y_i |$$

$$L_2(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

$$L_\infty(x,y) = \max_{i=1}^{n} |x_i - y_i|$$

- **Edit distance** (for strings)
  - minimal number of insertions, deletions and substitutions
  - d('application', 'applet') = 6

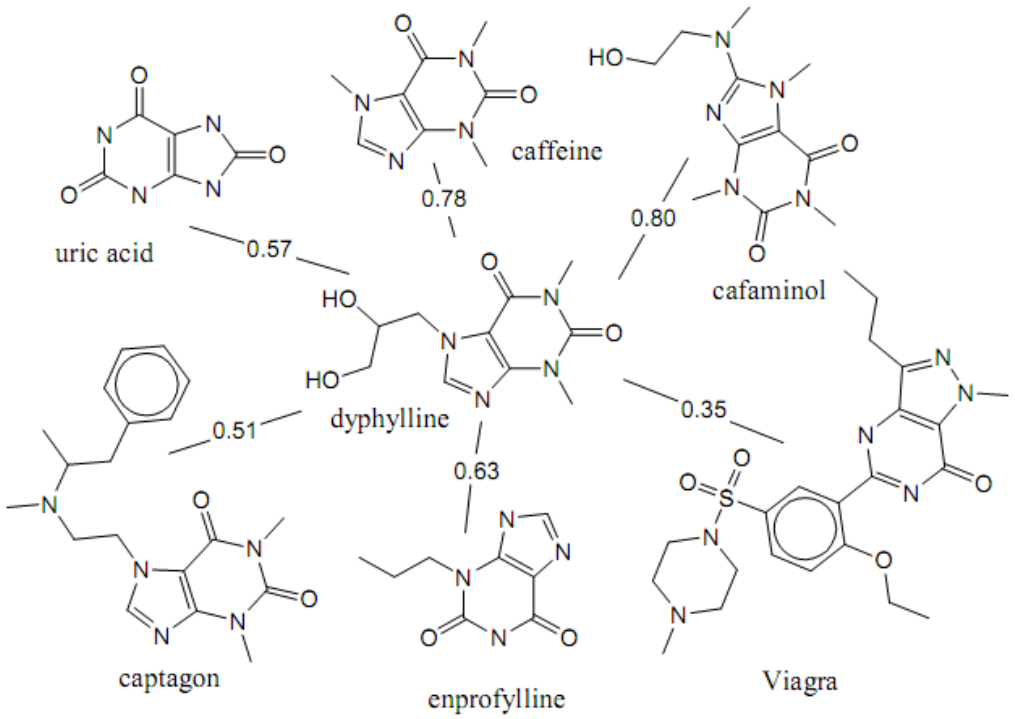
- **Jaccard's coefficient** (for sets A,B)

$$d(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

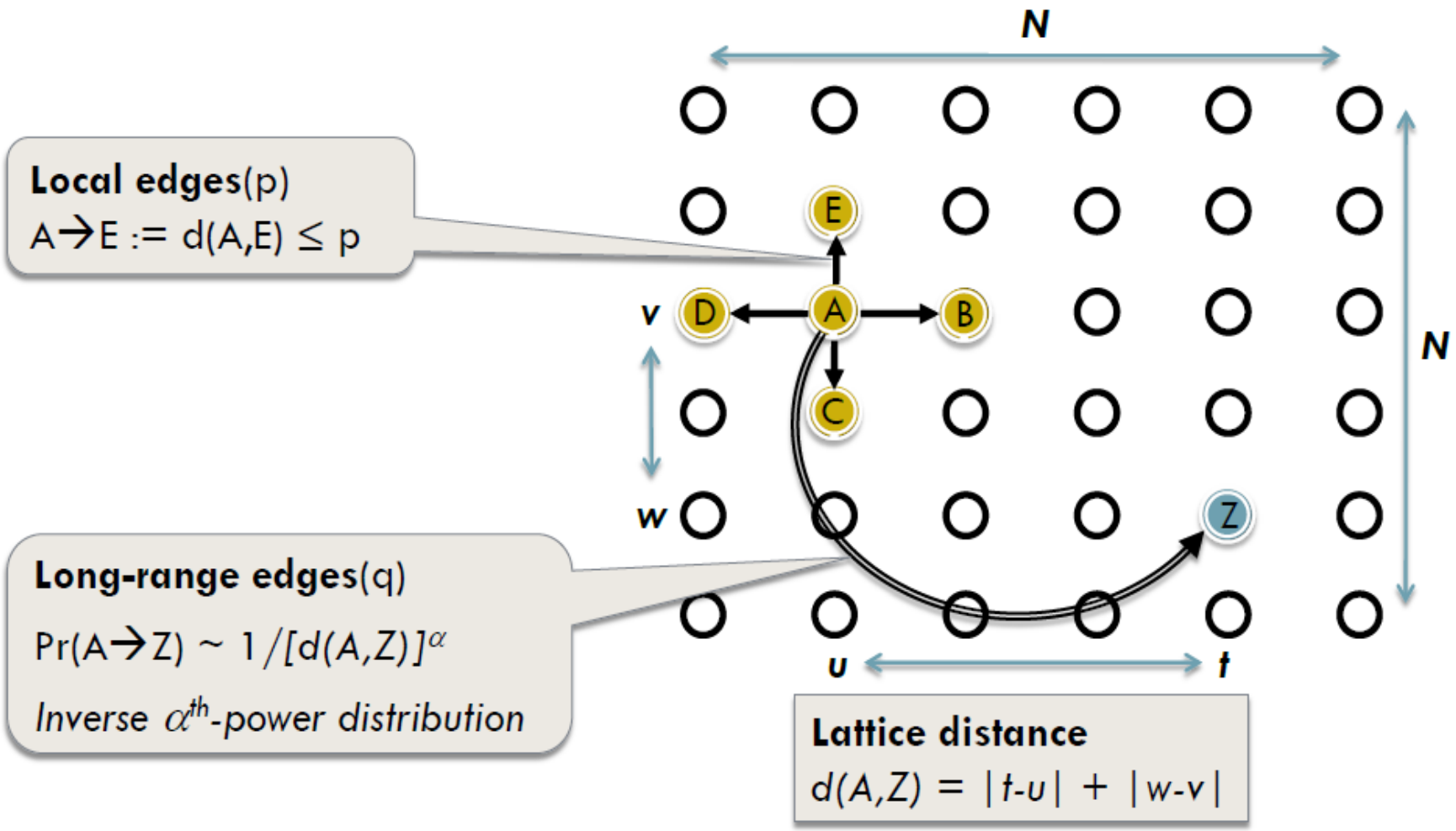
Max Common Subgraph Similarity

$$sim(G1,G2) = \frac{(|V(G1,G2)| + |E(G1,G2)|)^2}{(|V(G1)| + |E(G1)|) \cdot (|V(G2)| + |E(G2)|)}$$

$$d(G1,G2) = 1 - sim(G1,G2)$$

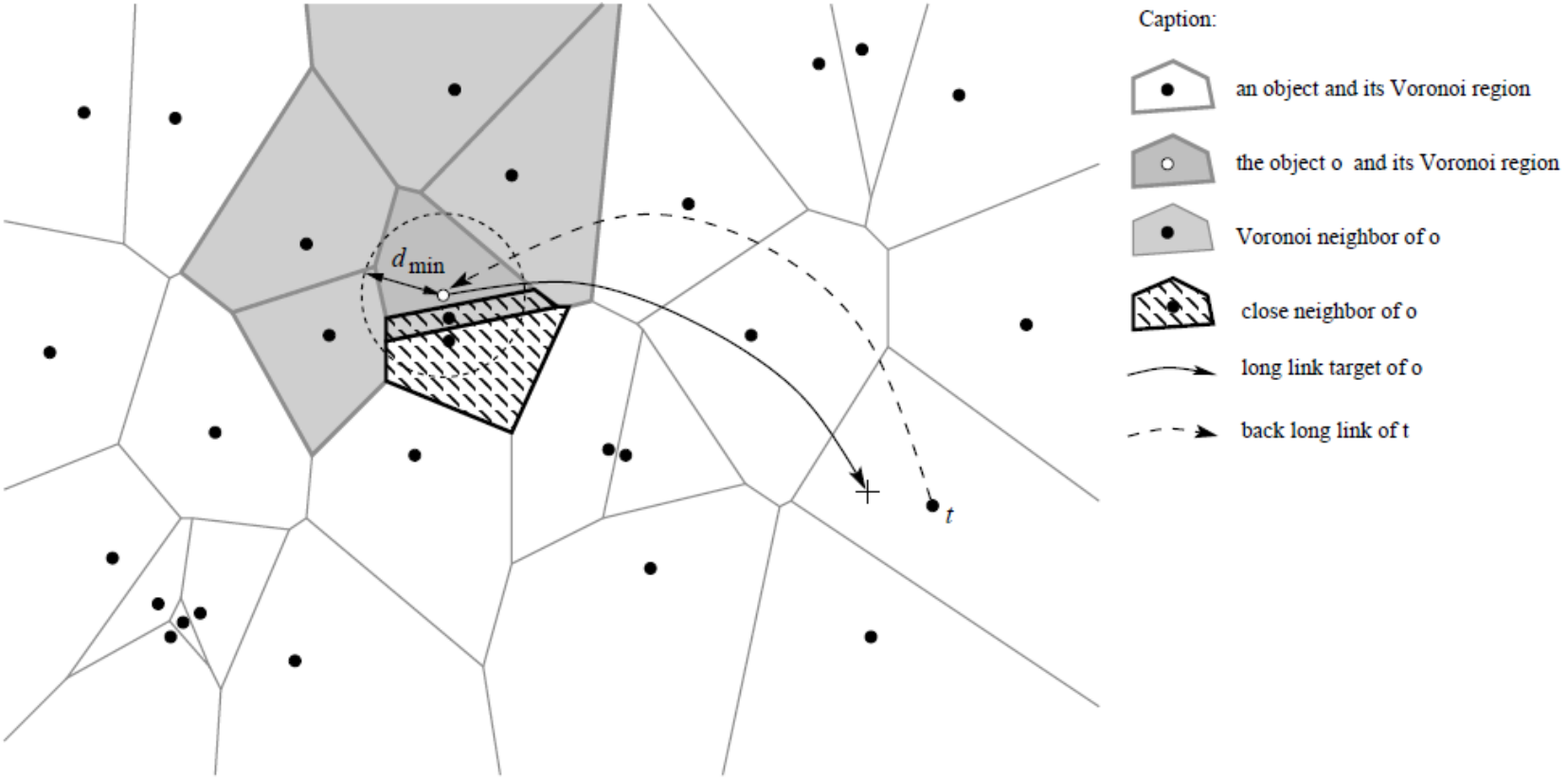


Kleinberg's Navigable Small World



[Kleinberg J. The small-world phenomenon: An algorithmic perspective //Proceedings of the thirty-second annual ACM symposium on Theory of computing. – ACM, 2000. – С. 163-170.]

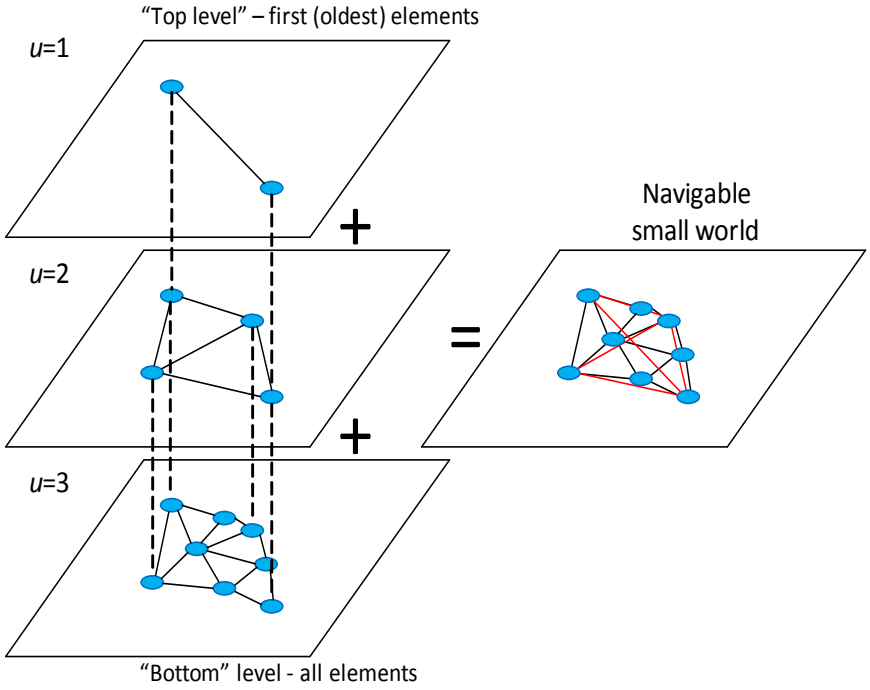
VoroNet, RayNet : A scalable object network based on Voronoi tessellations

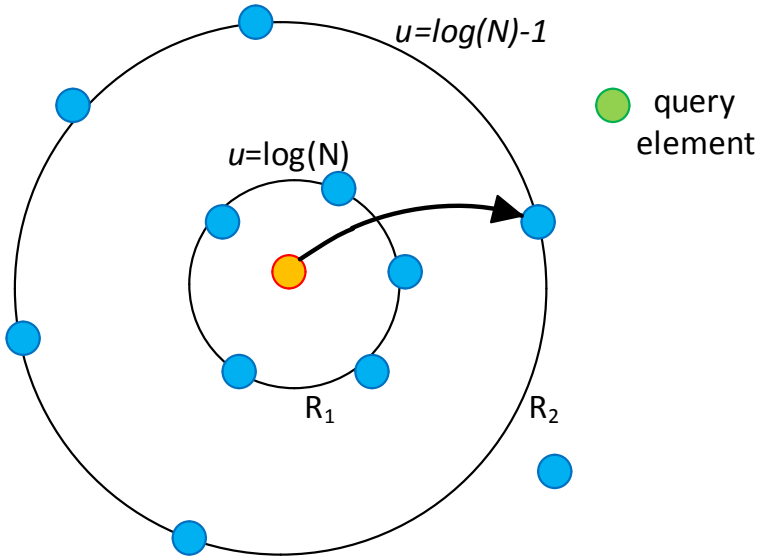


Distance function: $d(x, y) = \sqrt[2]{x^2 + y^2}$

Beaumont O. et al. VoroNet: A scalable object network based on Voronoi tessellations. – 2006.

Metrized Small World Algorithm





[Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov, "Scalable distributed algorithm for approximate nearest neighbor search problem in high dimensional general metric spaces," in Similarity Search and Applications. Springer, 2012, pp. 32–147]

[Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov, "Approximate nearest neighbor algorithm based on navigable small world graphs," Information Systems, vol. 45, 2014, pp. 61–68.]

[Ponomarenko A. Query-Based Improvement Procedure and Self-Adaptive Graph Construction Algorithm for Approximate Nearest Neighbor Search //International Conference on Similarity Search and Applications. – Springer International Publishing, 2015. – С. 314-319.]

Boolean non-linear programming formulation for optimal graph structure

Decision variables

$$x_{ij} = \begin{cases} 1, & \text{if edge } (i, j) \text{ belongs to the solution} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$y_{ij}^{k} = \begin{cases} 1, & \text{if vertex } k \text{ belongs to the greedy walk from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Objective function

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} O(i, j) \quad (2)$$

$$O(i, j) = \left| \left\{ l \in V : \exists k \, x_{lk} = 1 \text{ and } y_{ij}^{k} = 1 \right\} \right| \quad (4)$$

Constraints

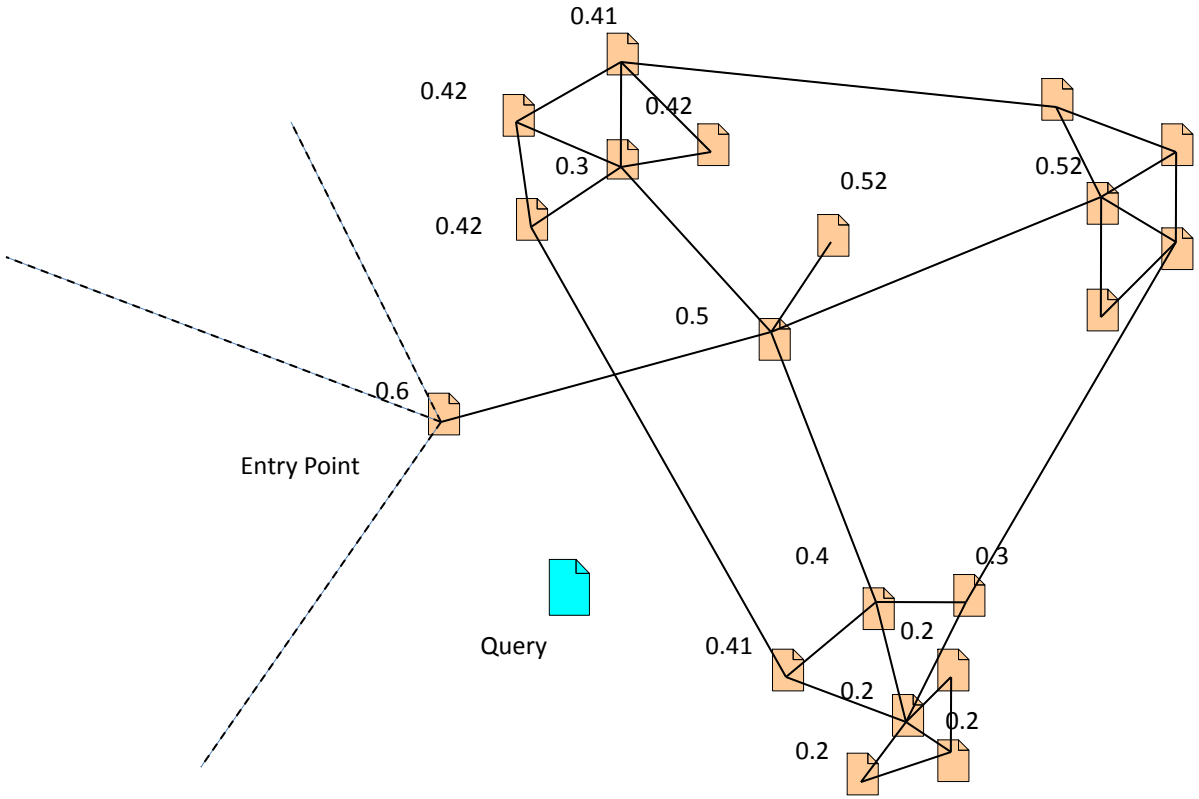
$$x_{ii} = 0 \quad \forall i \in V \quad (5)$$

$$y_{ij}^{i} = y_{ij}^{j} = 1 \quad \forall i, j \in V \quad (6)$$

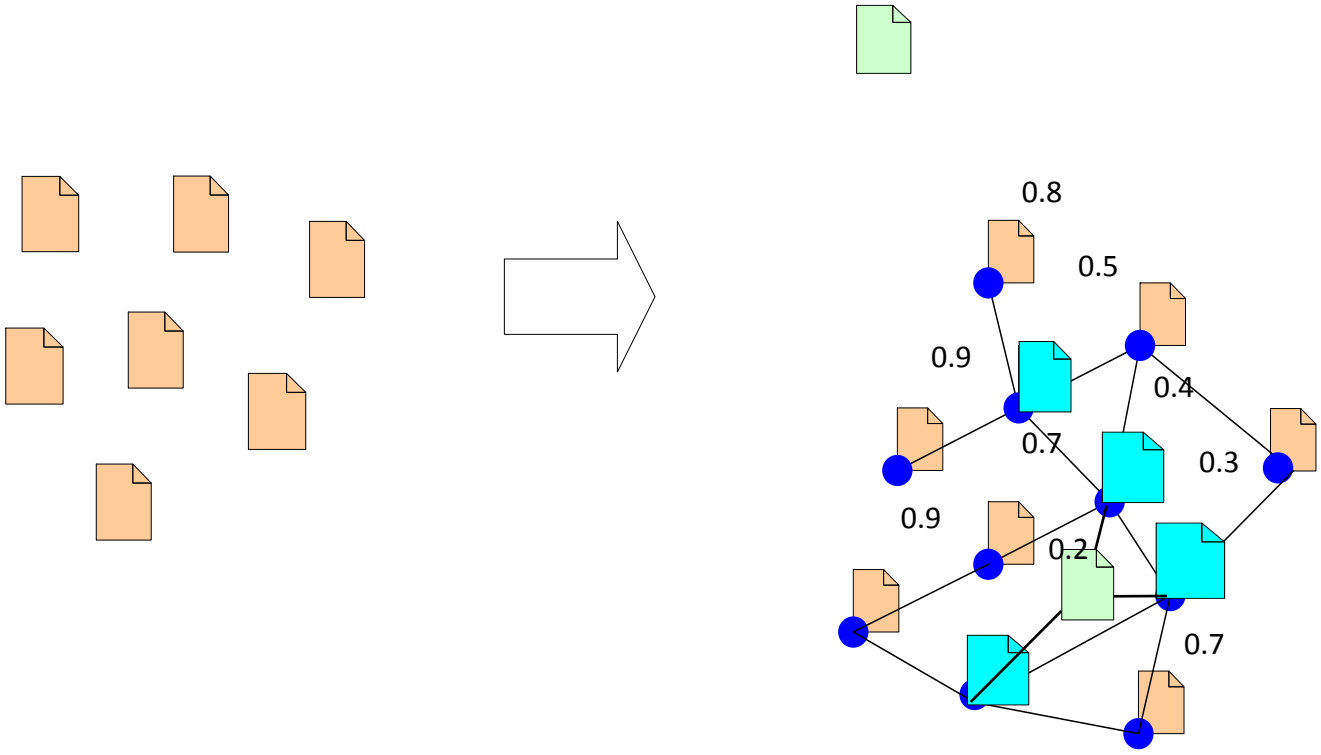
$$\sum_{k=1}^{n} x_{lk} y_{ij}^{k} \geq y_{ij}^{l} \quad \forall i, j, l \in V \quad (7)$$

$$l^{*} = \arg\min_{l \in V : x_{kl} = 1} (d(l, j)) \Rightarrow y_{ij}^{l^{*}} \geq y_{ij}^{k} \quad \forall i, j, k \in V, j \neq i, k \neq j \quad (8)$$

Search by greedy algorithm

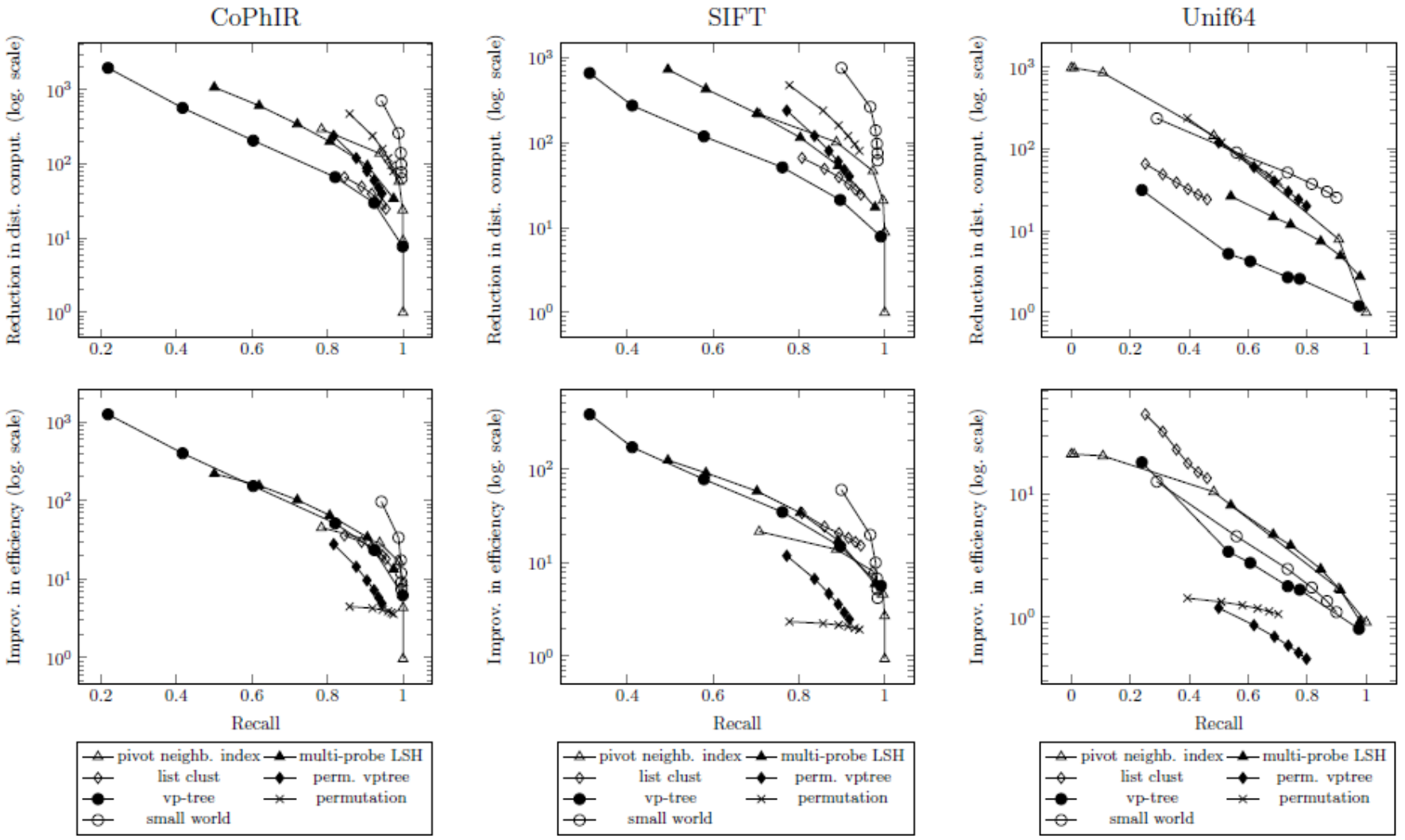


Construction algorithm



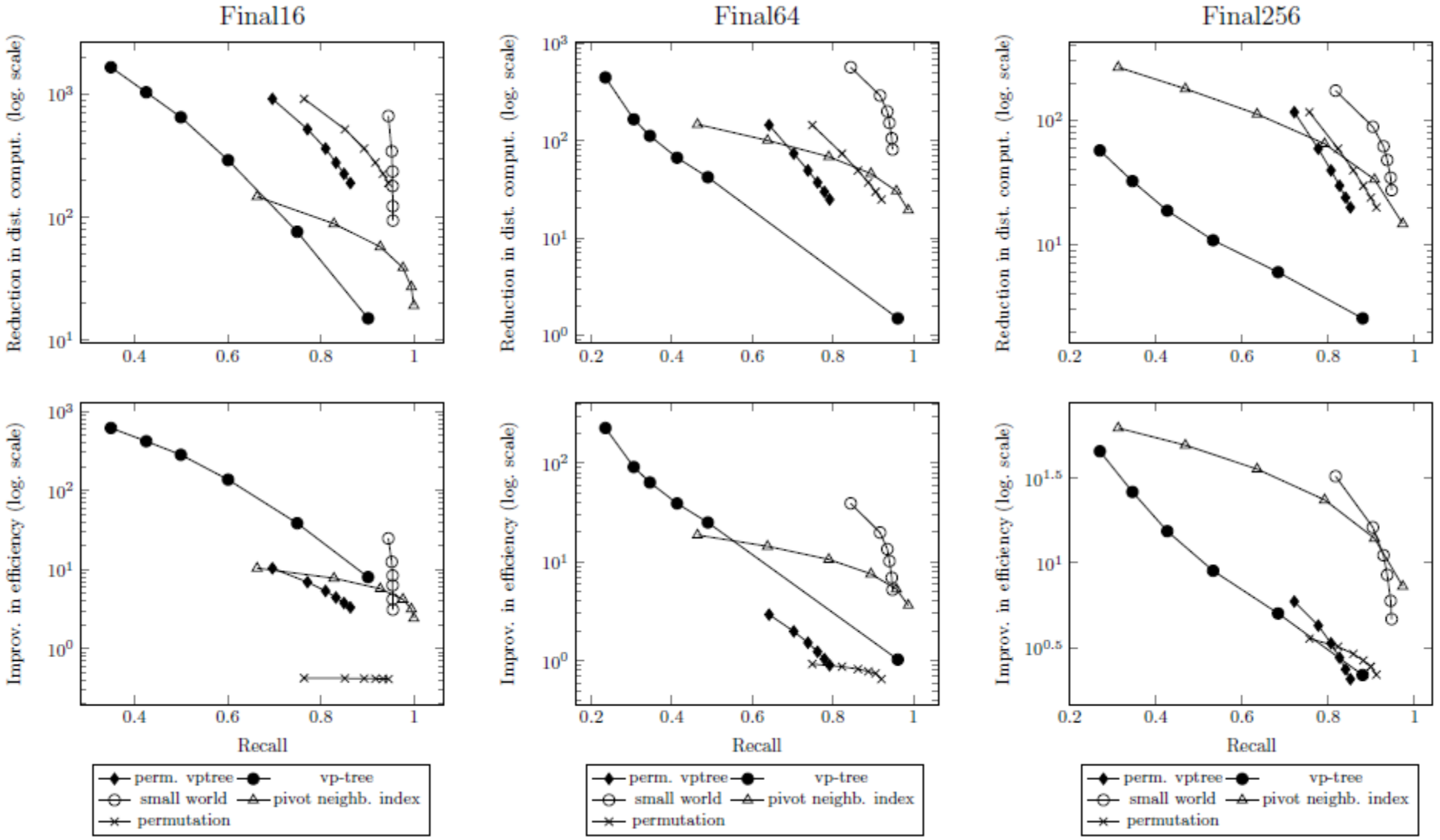
Data sets

- CoPhIR (L2) is the collection of 208-dimensional vectors extracted from images in MPEG7 format.
- SIFT is a part of the TexMex dataset collection available http://corpus-texmex.irisa.fr It has one million 128-dimensional vectors. Each vector corresponds to descriptor extracted from image data using Scale Invariant Feature Transformation (SIFT)
- Unfi64 is synthetic dataset of 64-dimensional vectors. The vectors were generated randomly, independently and uniformly in the unit hypercube.



Performance of a 10-NN search for L_2: plots in the same column correspond to the same data set

[Ponomarenko A. et al. Comparative Analysis of Data Structures for Approximate Nearest Neighbor Search //DATA ANALYTICS 2014, The Third International Conference on Data Analytics. – 2014. – С. 125-130.]



KL-divergence: $d(x,y) = \sum x_i \log \frac{x_i}{y_i}$

Final16, Final64, and Final256: are sets of 0.5 million topic histograms generated using the Latent Dirichlet Allocation (LDA).

Wikipedia dataset

Vector Space Model

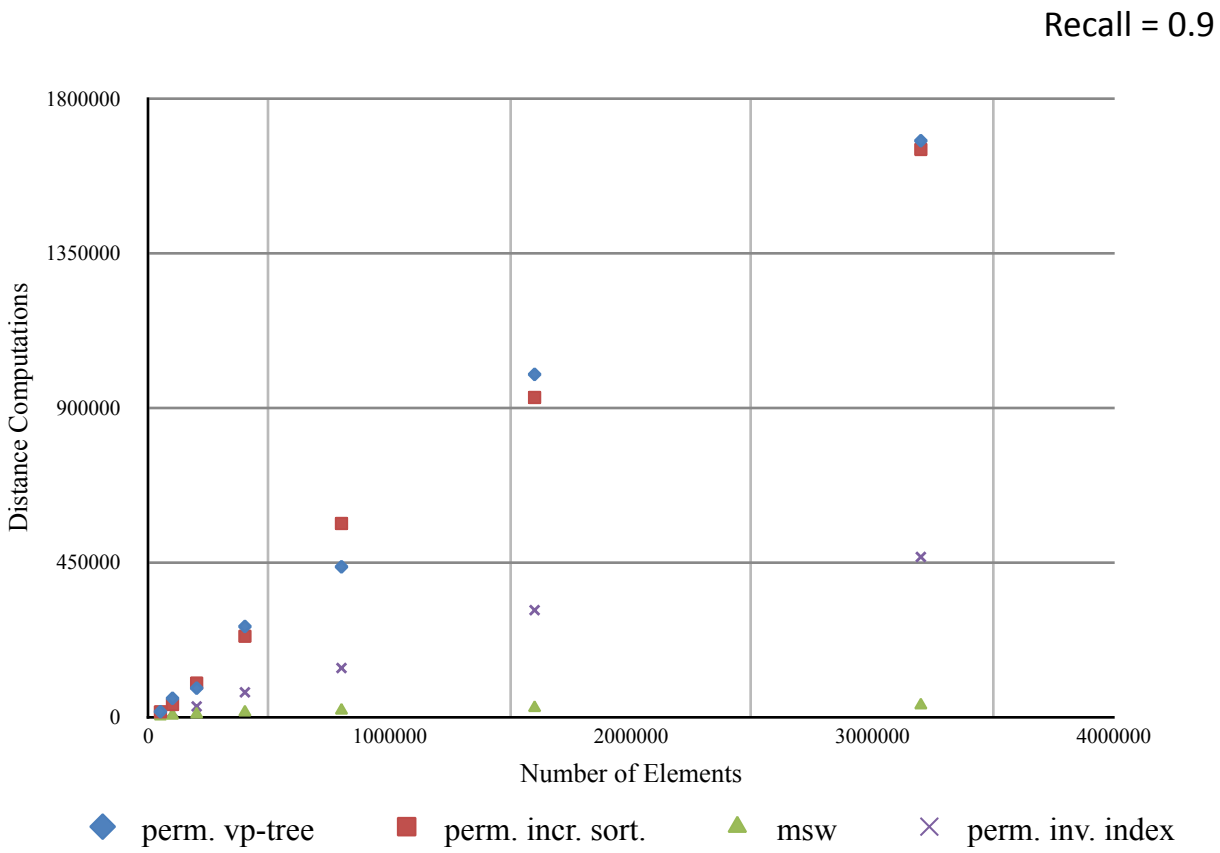
$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{n,j})$$

$$q = (w_{1,q}, w_{2,q}, \ldots, w_{n,q})$$

$$sim(d_j, q) = \frac{d_j \cdot q}{\| d_j \| \cdot \| q \|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$$

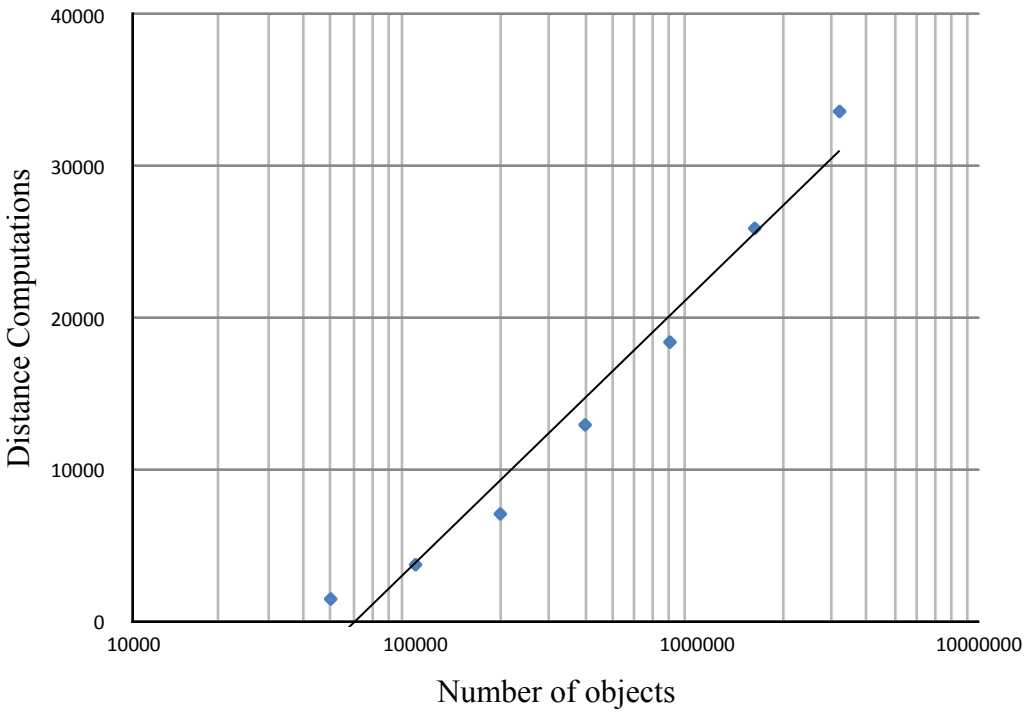
Wikipedia (cosine similarity): is a data set that contains 3.2 million vectors represented in a sparse format.
This set has an extremely high dimensionality (more than 100 thousand elements). Yet, the vectors are sparse: On average only about 600 elements are non-zero.

Scaling of methods on Wikipedia dataset



Wikipedia is dataset that contains 3.2 million vectors represented in a sparse format. Each vector corresponds to the frequency term vector of the Wikipedia page extracted using the gensim library. This set has an extremely high dimensionality (more than 100 thousand elements).

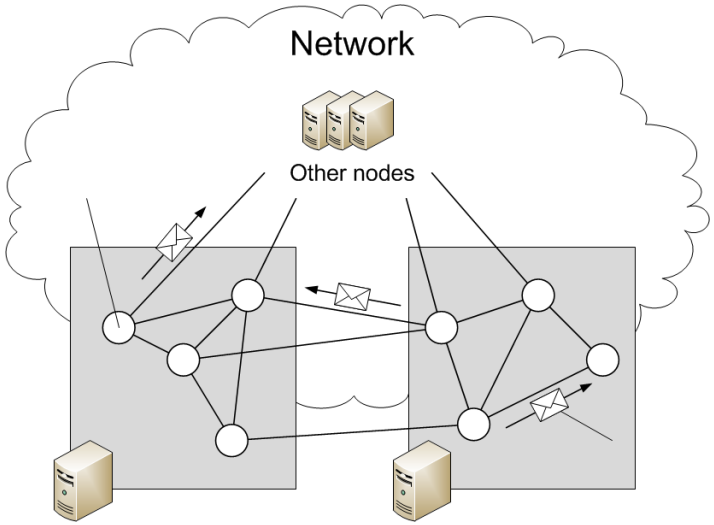
Scaling of MSW data structure



Summing up

- Algorithm is very simple
- Algorithm uses only distance values between the objects, making it suitable for arbitrary spaces.
- Proposed data structure has no root element.
- All operations (addition and search) use only local information and can be initiated from any element that was previously added to the structure.
- Accuracy of the approximate search can be tuned without rebuilding data structure
- Algorithm high scalable both in size and data dimensionality

Good base for building many real-world extreme dataset size high dimensionality similarity search applications



Source Code

https://github.com/searchivarius/NonMetricSpaceLib
https://github.com/aponom84/MetrizedSmallWorld

Questions?

Questions?

Why CERN doesn’t use DHT?