VoroNet: A scalable object network based on Voronoi tessellations

Olivier Beaumont, Anne-Marie Kermarrec, Loris Marchal and Étienne Rivière

SCALAPPLIX project, LaBRI (Bordeaux)
PARIS project, IRISA (Rennes)
GRAAL project, LIP, ENS Lyon

March 2006
Outline

1. Introduction
2. Description of VoroNet
3. Evaluation (simulation)
4. Perspectives, conclusion
What is peer-to-peer?

- paradigm to organize distributed resources (peers)
- overlay network: logical organization
- core functionality: search objects in the system
- distributed hash tables (DHT) (Chord, Pastry,...)
- hash function gives identifiers for peers and objects
- choice of hash function to get uniform distribution

Main goals:

- scalability 😊
- fault tolerance 😊
- efficient search 😊

but restricted to exact search 😞
highly depends on the hash function 😞
Peer-to-peer overlay

What is peer-to-peer?

- paradigm to organize distributed resources (peers)
- overlay network: logical organization
- core functionality: search objects in the system
- distributed hash tables (DHT) (Chord, Pastry,...)
- hash function gives identifiers for peers and objects
- choice of hash function to get uniform distribution

Main goals:

- scalability 😊  but restricted to exact search 😞
- fault tolerance 😊  highly depends on the hash function 😞
- efficient search 😊
What is peer-to-peer?

- paradigm to organize distributed resources (peers)
- overlay network: logical organization
- core functionality: search objects in the system
- distributed hash tables (DHT) (Chord, Pastry, ...)
- hash function gives identifiers for peers and objects
- choice of hash function to get uniform distribution

Main goals:

- scalability 😊
- fault tolerance 😊
- efficient search 😊

but restricted to exact search 😞

highly depends on the hash function 😞
Content-based topologies:

- **CAN (Content Adressable Network)**
  - $d$-dimensional torus
  - degree $O(d)$
  - diameter $O(N^{1/d})$
  - not really "content addressed": location (of objects and peers) computed with hash function (to ensure homogeneous distribution)
Content-based topologies:

- **CAN (Content Adressable Network)**
  - $d$-dimensional torus
  - degree $O(d)$
  - diameter $O(N^{1/d})$
  - not really “content addressed”: location (of objects and peers) computed with hash function (to ensure homogeneous distribution)

- ?
Content-based topologies:

- **CAN (Content Adressable Network)**
  - $d$-dimensional torus
  - degree $O(d)$
  - diameter $O(N^{1/d})$
  - not really “content addressed”: location (of objects and peers) computed with hash function (to ensure homogeneous distribution)

- ?
Object-based peer-to-peer overlay
  - objects are linked rather than peers
  - an object is held by the node which published it

Content-based topology:
  - not based on a DHT
  - objects with “close” attributes will be neighbors

d-dimensional attribute space

VoroNet topology is inspired from:
  - Voronoi diagram in the attribute space
  - Kleinberg's small world routing algorithm designed for grids
Object-based peer-to-peer overlay
  - objects are linked rather than peers
  - an object is held by the node which published it

Content-based topology:
  - not based on a DHT
  - objects with “close” attributes will be neighbors

$d$-dimensional attribute space

VoroNet topology is inspired from:
  - Voronoi diagram in the attribute space
  - Kleinberg’s small world routing algorithm designed for grids
Object-based peer-to-peer overlay
  ▶ objects are linked rather than peers
  ▶ an object is held by the node which published it

Content-based topology:
  ▶ not based on a DHT
  ▶ objects with “close” attributes will be neighbors

\( d \)-dimensional attribute space

VoroNet topology is inspired from:
  ▶ Voronoi diagram in the attribute space
  ▶ Kleinberg’s small world routing algorithm designed for grids
Object-based peer-to-peer overlay
- objects are linked rather than peers
- an object is held by the node which published it

Content-based topology:
- not based on a DHT
- objects with “close” attributes will be neighbors

$d$-dimensional attribute space we consider for now: $d = 2$

VoroNet topology is inspired from:
- Voronoi diagram in the attribute space
- Kleinberg’s *small world* routing algorithm designed for grids
VoroNet

- **Object-based peer-to-peer overlay**
  - objects are linked rather than peers
  - an object is held by the node which published it

- **Content-based topology:**
  - not based on a DHT
  - objects with “close” attributes will be neighbors

- **$d$-dimensional attribute space**
  - we consider for now: $d = 2$

- **VoroNet topology is inspired from:**
  - Voronoi diagram in the attribute space
  - Kleinberg’s *small world* routing algorithm designed for grids
Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than

- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Voronoi tessellations

- set of points in \( \mathbb{R}^2 \)
- consider object at point \( M \)
- region of points closer from \( M \) than
- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than
- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than from $A$

- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than from $A$
- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than from $A$ and $B$

- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Introduction

Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than from any other object: Voronoi cell of $M$ (or region)
- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Introduction

Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than from any other object: **Voronoi cell of $M$** (or region)
- do the same for all objects
- **Voronoi neighbors**: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than from any other object: Voronoi cell of $M$ (or region)
- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Voronoi tessellations

- set of points in $\mathbb{R}^2$
- consider object at point $M$
- region of points closer from $M$ than from any other object: Voronoi cell of $M$ (or region)
- do the same for all objects
- Voronoi neighbors: when cells share a border
- graph of Voronoi neighbors: Delaunay triangularization
Kleinberg’s small-world

- $N$ nodes in a 2D grid $(\sqrt{N} \times \sqrt{N})$
  - routing in $O(\sqrt{N})$
- add random long range links:
  - probability for a long link to be at distance $l$: $\propto \frac{1}{l^2}$
  - use greedy routing algorithm
  - then routing in $O(\ln^2 N)$
- can be extended to any dimension $d$, with proba $\propto \frac{1}{l^d}$
Kleinberg’s small-world

- $N$ nodes in a 2D grid \((\sqrt{N} \times \sqrt{N})\)
  - routing in $O(\sqrt{N})$
- add random long range links:
  - probability for a long link to be at distance $l$: $\propto \frac{1}{l^2}$
  - use greedy routing algorithm
  - then routing in $O(\ln^2 N)$
- can be extended to any dimension $d$, with proba $\propto \frac{1}{l^d}$
Kleinberg’s small-world

- $N$ nodes in a 2D grid $(\sqrt{N} \times \sqrt{N})$
  - routing in $O(\sqrt{N})$
- add random long range links:
  - probability for a long link to be at distance $l$: $\propto \frac{1}{l^2}$
  - use greedy routing algorithm
  - then routing in $O(\ln^2 N)$
- can be extended to any dimension $d$, with proba $\propto \frac{1}{l^d}$
Outline

1 Introduction

2 Description of VoroNet

3 Evaluation (simulation)

4 Perspectives, conclusion
Description of VoroNet

VoroNet neighborhood

- Voronoi neighbors
- long range neighbors:
  - randomly chose a target point $t$
  - the long range neighbors is the object “responsible” for point $t$
  - keep a back pointer for overlay maintenance
- close neighbors (within distance $d_{\text{min}}$) for convergence
VoroNet neighborhood

- Voronoi neighbors
- long range neighbors:
  - randomly chose a target point \( t \)
  - the long range neighbors is the object “responsible” for point \( t \)
  - keep a back pointer for overlay maintenance
- close neighbors (within distance \( d_{\text{min}} \)) for convergence
Voronoï neighbors

long range neighbors:
▶ randomly chose a target point \( t \)
▶ the long range neighbors is the object “responsible” for point \( t \)
▶ keep a back pointer for overlay maintenance

close neighbors (within distance \( d_{\text{min}} \)) for convergence
VoroNet neighborhood

- Voronoi neighbors
- long range neighbors:
  - randomly chose a target point \( t \)
  - the long range neighbors is the object “responsible” for point \( t \)
  - keep a back pointer for overlay maintenance
- close neighbors (within distance \( d_{\text{min}} \)) for convergence
VoroNet neighborhood

- Voronoi neighbors
- long range neighbors:
  - randomly chose a target point $t$
  - the long range neighbors is the object “responsible” for point $t$
  - keep a back pointer for overlay maintenance
- close neighbors (within distance $d_{\text{min}}$) for convergence
VoroNet neighborhood – details

- **Space**: 2-dimensional torus: \([0, 1] \times [0, 1]\)
- Long link target of object \(x\): distribution in \(1/d^2\)

\[
\Pr[\text{target}(x) \in B(y, dr)] = \alpha \frac{\pi r^2}{d(x, y)^2}
\]

- Link always points on the closest object from target.
- Close neighbors: within \(d_{\text{min}} = \frac{1}{\pi N_{\text{max}}}\)

\(N_{\text{max}} = \text{maximal number of nodes for which we have an efficient routing}\)
VoroNet neighborhood – details

- **Space**: 2-dimensional torus: $[0, 1] \times [0, 1]$
- **Long link target of object $x$**: distribution in $1/d^2$

\[
\Pr[\text{target}(x) \in B(y, dr)] = \alpha \frac{\pi r^2}{d(x, y)^2}
\]

- Link always points on the closest object from target.
- Close neighbors: within $d_{\min} = \frac{1}{\pi N_{\max}}$

$N_{\max}$ = maximal number of nodes for which we have an efficient routing
VoroNet neighborhood – details

- **Space:** 2-dimensional torus: \([0, 1] \times [0, 1]\)
- **Long link target of object** \(x\): distribution in \(1/d^2\)

\[
\Pr[\text{target}(x) \in B(y, dr)] = \alpha \frac{\pi r^2}{d(x, y)^2}
\]

- **Link always points on the closest object from target.**
- **Close neighbors:** within \(d_{\min} = \frac{1}{\pi N_{\max}}\)

\(N_{\max} = \text{maximal number of nodes for which we have an efficient routing}\)
VoroNet neighborhood – details

- Space: 2-dimensional torus: $[0, 1] \times [0, 1]$
- Long link target of object $x$: distribution in $1/d^2$

$$\Pr[\text{target}(x) \in B(y, dr)] = \alpha \frac{\pi r^2}{d(x, y)^2}$$

- Link always points on the closest object from target.
- Close neighbors: within $d_{\text{min}} = \frac{1}{\pi N_{\text{max}}}$

$N_{\text{max}} = \text{maximal number of nodes for which we have an efficient routing}$
Size of the neighborhood

- **Voronoi neighbors:**
  - graph of the Voronoi neighbors is planar $\Rightarrow$ average degree $\leq 6$
  - $O(1)$ size
  - mean value $\leq 6$ (max = $N$)

- Close neighbors: number of objects in $B(o, d_{\min})$
  - $O(1)$ size for a “reasonable” distribution
  - mean value $= 1$ (max = $N$)

- Long range neighbors:
  - one long range neighbor per object
  - $O(1)$ backward links for a “reasonable” distribution
  - mean value $= 1$ (max = $N$)

Size of data stored at each node: $O(1)$ mean value $\leq 9$
(we will check this property in the experiments)
Size of the neighborhood

- Voronoi neighbors:
  - graph of the Voronoi neighbors is planar $\Rightarrow$ average degree $\leq 6$
  - $O(1)$ size
  - mean value $\leq 6$ (max = $N$)

- Close neighbors: number of objects in $B(o, d_{\text{min}})$
  - $O(1)$ size for a “reasonable” distribution
  - mean value = 1 (max = $N$)

- Long range neighbors:
  - one long range neighbor per object
  - $O(1)$ backward links for a “reasonable” distribution
  - mean value = 1 (max = $N$)

Size of data stored at each node: $O(1)$ mean value $\leq 9$
(we will check this property in the experiments)
Size of the neighborhood

- **Voronoi neighbors:**
  - graph of the Voronoi neighbors is planar \( \Rightarrow \) average degree \( \leq 6 \)
  - \( O(1) \) size
  - mean value \( \leq 6 \) (max = \( N \))

- **Close neighbors:** number of objects in \( B(o, d_{\text{min}}) \)
  - \( O(1) \) size for a “reasonable” distribution
  - mean value = 1 (max = \( N \))

- **Long range neighbors:**
  - one long range neighbor per object
  - \( O(1) \) backward links for a “reasonable” distribution
  - mean value = 1 (max = \( N \))

size of data stored at each node: \( O(1) \) mean value \( \leq 9 \)

(we will check this property in the experiments)
Voronoi neighbors:
- graph of the Voronoi neighbors is planar $\Rightarrow$ average degree $\leq 6$
- $O(1)$ size
- mean value $\leq 6$ \hspace{0.5cm} (max = $N$)

Close neighbors: number of objects in $\mathcal{B}(o, d_{\text{min}})$
- $O(1)$ size for a “reasonable” distribution
- mean value = 1 \hspace{0.5cm} (max = $N$)

Long range neighbors:
- one long range neighbor per object
- $O(1)$ backward links for a “reasonable” distribution
- mean value = 1 \hspace{0.5cm} (max = $N$)

size of data stored at each node: $O(1)$ mean value $\leq 9$
(we will check this property in the experiments)
Overlay maintenance

How to insert an object $x$?

- Update the Voronoi diagram:
  - Find the closest existing object (route to $x$)
  - Add a new Voronoi cell
  - Find and update the Voronoi neighbors

- Find and the close neighbors sufficient to consider close neighbors of Voronoi neighbors

- Create a long range target point $t$, find the corresponding object:
  - Route a JOIN message to $t$
How to insert an object $x$?

- Update the Voronoi diagram:
  - Find the closest existing object (route to $x$)
  - Add a new Voronoi cell
  - Find and update the Voronoi neighbors

- Find and the close neighbors
  sufficient to consider close neighbors of Voronoi neighbors

- create a long range target point $t$, find the corresponding object:
  $\Rightarrow$ route a JOIN message to $t$
Overlay maintenance

How to insert an object $x$?

- Update the Voronoi diagram:
  - Find the closest existing object (route to $x$)
  - Add a new Voronoi cell
  - Find and update the Voronoi neighbors

- Find and the close neighbors sufficient to consider close neighbors of Voronoi neighbors

- create a long range target point $t$, find the corresponding object:
  $\Rightarrow$ route a JOIN message to $t$
General routing scheme

**Route** *(Target)*:

find the object responsible for the Voronoi cell where *Target* is.

```
Route(Target)

z = DistanceToRegion(Target)
if d(z, Target) > \( \frac{1}{3} d(Target, CurrentObject) \) and 
d(Target, CurrentObject) > d_{\text{min}} then
    Spawn(Route, Target, GreedyNeighbor(Target))
else
    AddVoronoiRegion(z)
    AddVoronoiRegion(Target)
    Perform some local computations depending on the operation at z
    RemoveVoronoiRegion(z)
    (depending on the operation,
    RemoveVoronoiRegion(Target))
return
```
Lemma 1

The probability for the long link of $x$ to be chosen in a disk of center $y$ and radius $fr$, where $r = d(x, y)$ is lower bounded by $\pi f^2 \frac{1}{K(1+f)^2}$.

- For $f = 1/6$: probability lower bounded by: $\frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})}$

- $X$: number of hops necessary to reach the disk of center $Target$ and radius $\frac{d}{6}$.

$$E(X) = \sum_{i=1}^{+\infty} \Pr[X \geq i] \leq \sum_{i=1}^{+\infty} \left(1 - \frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})}\right)^{i-1}$$

$E(X) \leq 98 \ln(\sqrt{2\pi N_{\text{max}}})$.

but link target $\neq$ link destination

This accounts for a super-step.
Lemma 1

The probability for the long link of $x$ to be chosen in a disk of center $y$ and radius $fr$, where $r = d(x, y)$ is lower bounded by $\frac{\pi f^2}{K(1+f)^2}$.

- For $f = 1/6$: probability lower bounded by: $\frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})}$

- $X$: number of hops necessary to reach the disk of center Target and radius $\frac{d}{6}$.

$$E(X) = \sum_{i=1}^{+\infty} \Pr[X \geq i] \leq \sum_{i=1}^{+\infty} \left(1 - \frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})}\right)^{i-1}$$

$$E(X) \leq 98 \ln(\sqrt{2\pi N_{\text{max}}}).$$

but link target $\neq$ link destination

This accounts for a super-step.
**Lemma 1**

The probability for the long link of $x$ to be chosen in a disk of center $y$ and radius $fr$, where $r = d(x, y)$ is lower bounded by $\frac{\pi f^2}{K(1+f)^2}$.

- For $f = 1/6$: probability lower bounded by: $\frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})}$
- $X$: number of hops necessary to reach the disk of center $\text{Target}$ and radius $\frac{d}{6}$.

$$E(X) = \sum_{i=1}^{+\infty} \Pr[X \geq i] \leq \sum_{i=1}^{+\infty} \left(1 - \frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})}\right)^{i-1}$$

$$E(X) \leq 98 \ln(\sqrt{2\pi N_{\text{max}}}).$$

but link target $\neq$ link destination

This accounts for a super-step.
Lemma 1

The probability for the long link of $x$ to be chosen in a disk of center $y$ and radius $fr$, where $r = d(x, y)$ is lower bounded by $\frac{\pi f^2}{K(1+f)^2}$.

- For $f = 1/6$: probability lower bounded by: $\frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})}$
- $X$: number of hops necessary to reach the disk of center Target and radius $\frac{d}{6}$.

$$E(X) = \sum_{i=1}^{+\infty} \Pr[X \geq i] \leq \sum_{i=1}^{+\infty} \left(1 - \frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})}\right)^{i-1}$$

$$E(X) \leq 98 \ln(\sqrt{2\pi N_{\text{max}}}).$$

but link target $\neq$ link destination

This accounts for a super-step.
Lemma 1

The probability for the long link of $x$ to be chosen in a disk of center $y$ and radius $fr$, where $r = d(x, y)$ is lower bounded by $\frac{\pi f^2}{K(1+f)^2}$.

- For $f = 1/6$: probability lower bounded by: $\frac{1}{98 \ln(\sqrt{2\pi N_{\max}})}$
- $X$: number of hops necessary to reach the disk of center $Target$ and radius $\frac{d}{6}$.

\[
E(X) = \sum_{i=1}^{+\infty} \Pr[X \geq i] \leq \sum_{i=1}^{+\infty} \left(1 - \frac{1}{98 \ln(\sqrt{2\pi N_{\max}})}\right)^{i-1}
\]

$E(X) \leq 98 \ln(\sqrt{2\pi N_{\max}})$.

but link target $\neq$ link destination

This accounts for a super-step.
**Lemma 1**

The probability for the long link target of $x$ to be chosen in a disk of center $y$ and radius $fr$, where $r = d(x, y)$ is lower bounded by $\frac{\pi f^2}{K(1+f)^2}$.

- For $f = 1/6$: probability lower bounded by: $\frac{1}{98 \ln(\sqrt{2\pi N_{\max}})}$

- $X$: number of hops necessary to reach an object whose long link target belongs to the disk of center Target and radius $\frac{d}{6}$.

$$E(X) = \sum_{i=1}^{+\infty} \Pr[X \geq i] \leq \sum_{i=1}^{+\infty} \left(1 - \frac{1}{98 \ln(\sqrt{2\pi N_{\max}})}\right)^{i-1}$$

$$E(X) \leq 98 \ln(\sqrt{2\pi N_{\max}}).$$

but link target $\neq$ link destination

This accounts for a super-step.
Lemma 1

The probability for the long link **target** of \( x \) to be chosen in a disk of center \( y \) and radius \( fr \), where \( r = d(x, y) \) is lower bounded by \( \frac{\pi f^2}{K(1+f)^2} \).

- For \( f = 1/6 \): probability lower bounded by: \( \frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})} \)

- \( X \): number of hops necessary to reach an object whose long link **target** belongs to the disk of center \( \text{Target} \) and radius \( \frac{d}{6} \).

\[
E(X) = \sum_{i=1}^{+\infty} \Pr[X \geq i] \leq \sum_{i=1}^{+\infty} \left( 1 - \frac{1}{98 \ln(\sqrt{2\pi N_{\text{max}}})} \right)^{i-1}
\]

\[
E(X) \leq 98 \ln(\sqrt{2\pi N_{\text{max}}}).
\]

but **link target \( \neq \) link destination**

This accounts for a **super-step**.
During a super-step, the distance to the target is divided by \(5/6\).

Number of super-steps bounded by

\[
\frac{\ln\left(\frac{\sqrt{2}}{d_{\text{min}}}\right)}{\ln\left(\frac{6}{5}\right)} = \frac{\ln(\sqrt{2\pi N_{\text{max}}})}{\ln\left(\frac{6}{5}\right)}
\]

Expectation of number of steps:

\[E(N) \leq \alpha \ln^2(N_{\text{max}})\]
Outline

1. Introduction
2. Description of VoroNet
3. Evaluation (simulation)
4. Perspectives, conclusion
Simulations
  ▶ 300,000 objects
  ▶ objects are not leaving the overlay (for now)

Distribution of object
  ▶ uniform
  ▶ skewed (powerlaw with parameter $\alpha = 1, 2, 5$)

We observe:
  ▶ number of neighbors
  ▶ polylogarithmic routing
  ▶ what happens if we add several long range links
Evaluation (simulation)

Outgoing degree

Uniform distribution

Skewed distribution ($\alpha = 5$)
Polylogarithmic routing

\[ h = (\ln n)^\gamma \iff \ln(h) = \ln\left((\ln n)^\gamma\right) = \gamma \ln(\ln n) \]
Polylogarithmic routing

\[ h = (\ln n)^\gamma \iff \ln(h) = \ln \left( (\ln n)^\gamma \right) = \gamma \ln(\ln n) \]
Using several long links to improve routing
1 Introduction
2 Description of VoroNet
3 Evaluation (simulation)
4 Perspectives, conclusion
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d + 1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d + 1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d + 1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood

- Use other techniques: lifting in dimension $d + 1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d+1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d + 1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d + 1$ and LP

$$z = x^2 + y^2$$
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d + 1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d + 1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d+1$ and LP
Extension to higher dimensional cases

- No bound on the number of neighbors, $O(1)$ data size?
- Extend small world property seems possible
- Compute Voronoi diagram: geometric algorithms costly and sensitive to computation errors
- No need to have complete description of Voronoi cells, only compute neighborhood
- Use other techniques: lifting in dimension $d+1$ and LP
Conclusion

Perspectives:

- Range queries
- Queries by proximity: all objects within $d$ from a given object
- Fault tolerance?

VoroNet in a nutshell:

- Object-to-object overlay
- Efficient routing
- Distributed construction and management
- Reasonable size of neighborhood
-Insensitive to object distribution
- Base for complex requests
Perspectives:

- Range queries
- Queries by proximity: all objects within \( d \) from a given object
- Fault tolerance?

VoroNet in a nutshell:

- Object-to-object overlay
- Efficient routing
- Distributed construction and management
- Reasonable size of neighborhood
- Insensitive to object distribution
- Base for complex requests