MapReduce Design Patterns

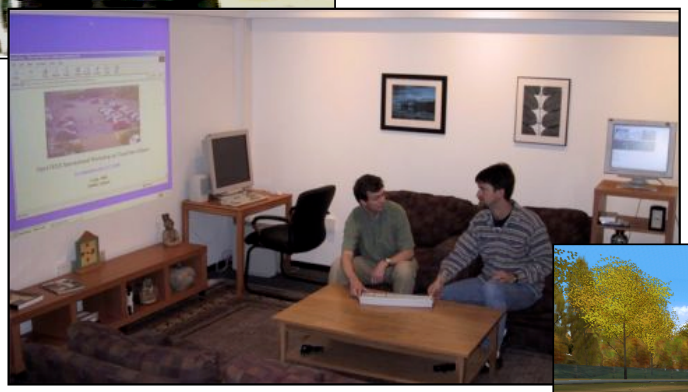
Barry Brumitt

barryb@google.com
Software Engineer

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 1
MapReduce Design Patterns
IPR2024-00303

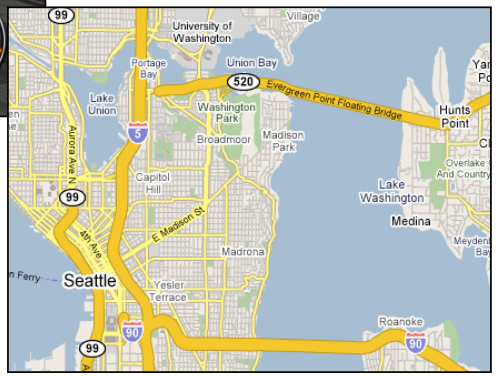
About your speaker…

Ph.D. Robotics, Carnegie Mellon, ‘91 - ’97
Path Planning for Multiple Mobile Robots



Researcher, Microsoft Research, ‘98 - ’02
Ubiquitous Computing

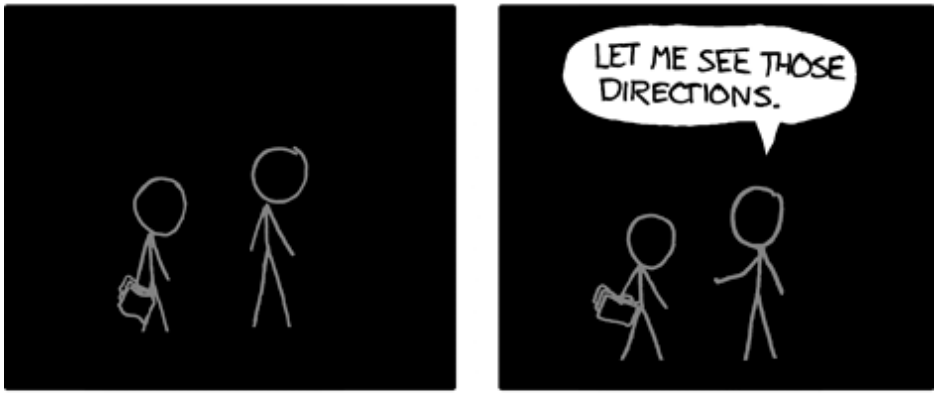
Software Eng., Microsoft Games, ‘03 - ’05
AI for Forza Motorsport



Software Engineer, Google, ‘05 - now
Maps: Pathfinder
Systems: Infrastructure

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 2
MapReduce Design Patterns
IPR2024-00303

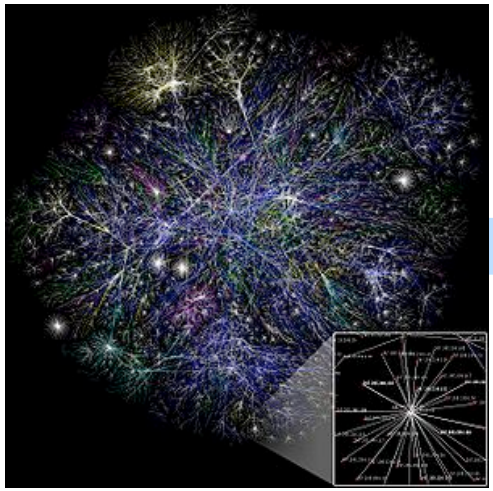
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 3
MapReduce Design Patterns
IPR2024-00303



74.	TAKE THE **FERRY** ACROSS THE **LAKE.**	GO 2.8 MI
75.	CLIMB THE **HILL** TOWARD **HANGMAN'S RIDGE,** AVOIDING ANY **MOUNTAIN LIONS.**	UP 1,172 FT
76.	WHEN YOU REACH AN **OLD BARN,** GO AROUND BACK, KNOCK ON THE **SECOND DOOR,** AND ASK FOR **CHARLIE.**	GO 52 FT
77.	TELL **CHARLIE** THE **DANCING STONES** ARE **RESTLESS.** HE WILL GIVE YOU HIS **VAN.**	CAREFUL
78.	TAKE **CHARLIE'S VAN** DOWN **OLD MINE ROAD.** DO NOT WAKE THE **STRAW MAN.**	GO 11 MI
79.	TURN LEFT ON **COMSTOCK.** WHEN YOU FEEL THE **BLOOD** CHILL IN YOUR **VEINS,** STOP THE VAN AND **GET OUT.**	GO 3.2 MI
80.	STAND VERY STILL. EXITS ARE **NORTH, SOUTH,** AND **EAST,** BUT ARE BLOCKED BY A **SPECTRAL WOLF.**	GO 0 FT
81.	THE **SPECTRAL WOLF** FEARS ONLY **FIRE.** THE **GOOGLE MAPS TEAM** CAN NO LONGER HELP YOU, BUT IF YOU MASTER THE **WOLF,** HE WILL GUIDE YOU. **GODSPEED.**	GO ?? MI

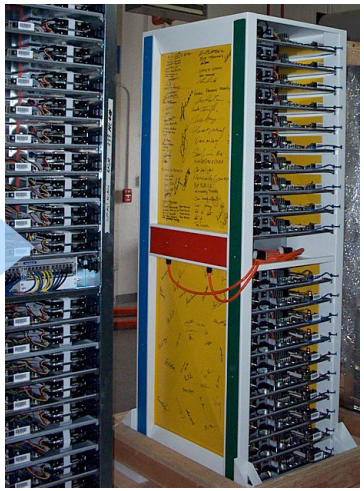
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 4
MapReduce Design Patterns
IPR2024-00303

Indexing Large Datasets



All web pages

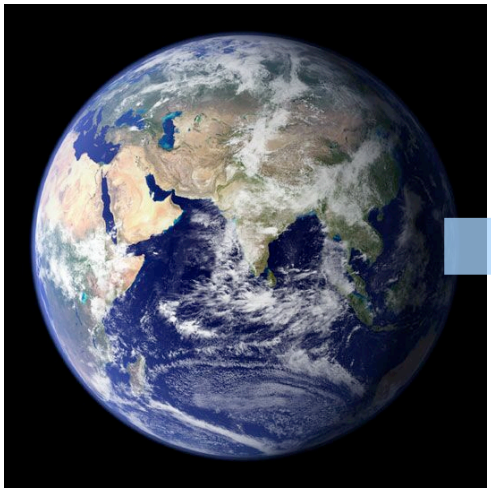
Index Files



Data Center

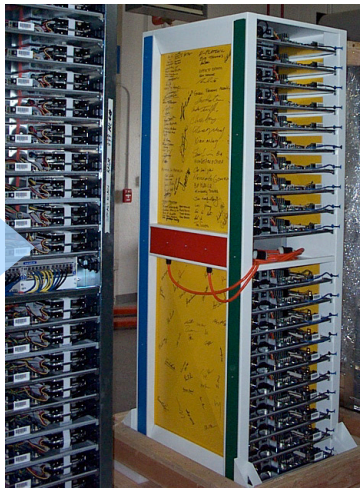
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 5
MapReduce Design Patterns
IPR2024-00303

Indexing Large Datasets



Geographic Data

Index Files



Data Center

...not so useful for user-facing applications…

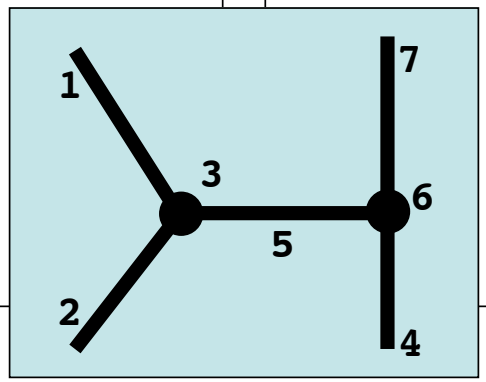
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 6
MapReduce Design Patterns
IPR2024-00303

Pointer Following (or) Joining

Input

Feature List

```
1: <type=Road>, <intersections=(3)>, <geom>, …
2: <type=Road>, <intersections=(3)>, <geom>, …
3: <type=Intersection>, stop_type, POI? …
4: <type=Road>, <intersections=(6)>, <geom>,
5: <type=Road>, <intersections=(3,6)>, <geom>, …
6: <type=Intersection>, stop_type, POI?,  …
7: <type=Road>, <intersections=(6)>, <geom>, …
8: <type=Town>, <name>, <geom>, …
                        .
                        .
                        .
```



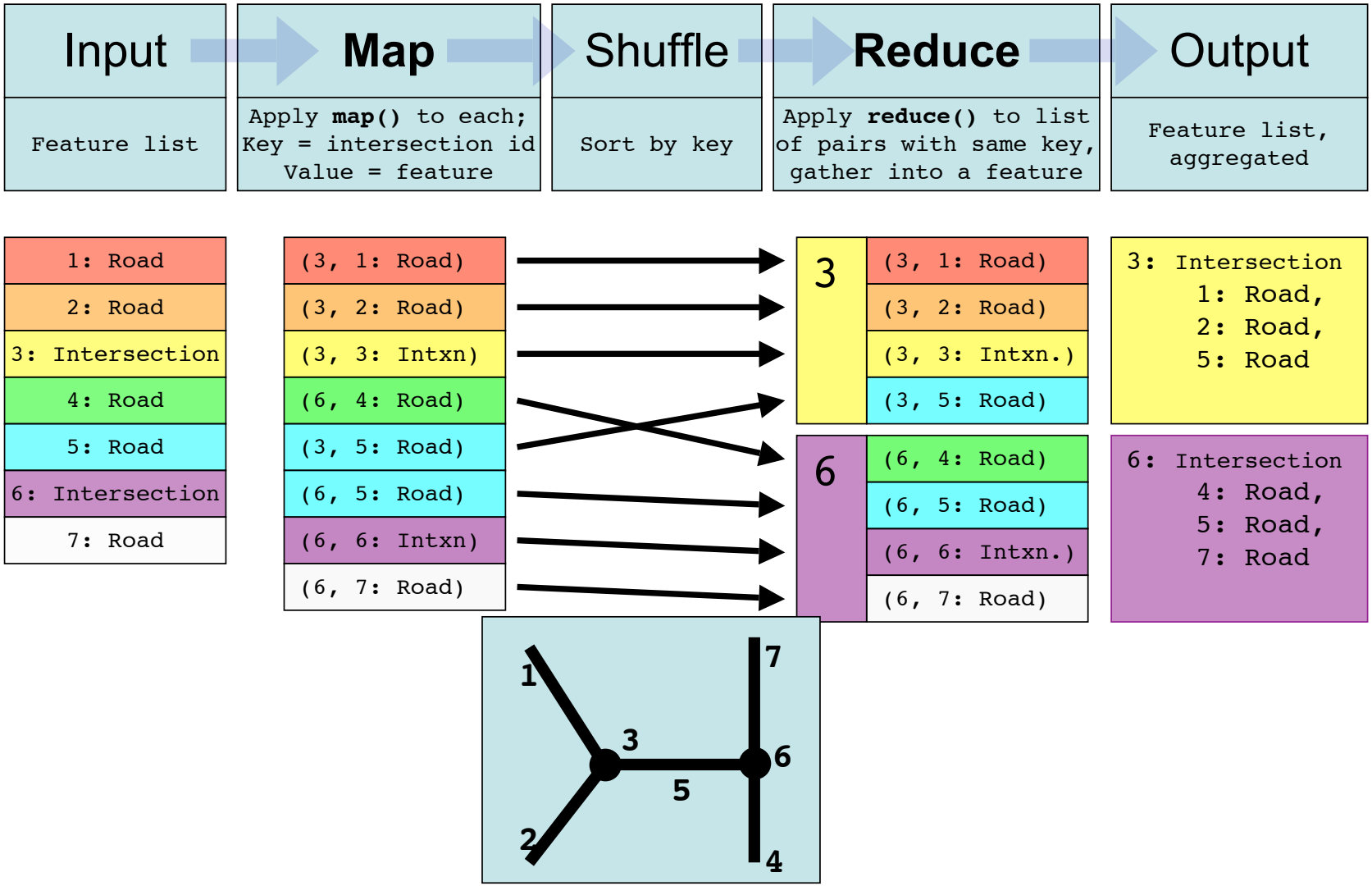
Output

Intersection List

```
3: <type=Intersection>, stop_type, <roads=(
     1: <type=Road>, <geom>, <name>, …
     2: <type=Road>, <geom>, <name>, …
     5: <type=Road>, <geom>, <name>, … )>, …
6: <type=Intersection>, stop_type, <roads=(
     4: <type=Road>, <geom>, <name>, … ,
     5: <type=Road>, <geom>, <name>, … ,
     7: <type=Road>, <geom>, <name>, … )>, …
                        .
                        .
                        .
```

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 7
MapReduce Design Patterns
IPR2024-00303

Inner Join Pattern



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 8
MapReduce Design Patterns
IPR2024-00303

Inner Join Pattern in SQL

roads

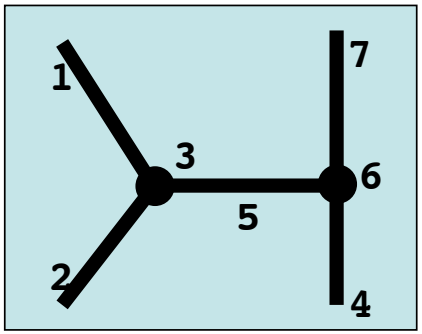
R	D	I
1	a	3
2	b	3
4	c	6
5	d	6
7	e	6

ints

I	D
3	x
6	y

```
SELECT roads.R, roads.D, ints.D
FROM roads INNER JOIN ints
ON roads.I = ints.I
```

r.R	r.D	i.D
1	a	x
2	b	x
4	c	y
5	d	y
7	e	x



r.R	r.D	r.I	i.I	i.D
1	a	3	3	x
2	b	3	3	x
4	c	6	3	x
5	d	6	3	x
7	e	6	3	x
1	a	3	6	y
2	b	3	6	y
4	c	6	6	y
5	d	6	6	y
7	e	6	6	y

“Cross Join”

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 9
MapReduce Design Patterns
IPR2024-00303

Inner Join Pattern in SQL

roads

R	D	I
1	a	3
2	b	3
4	c	6
5	d	6
7	e	5

ints

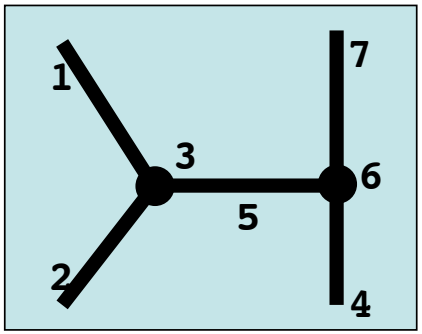
I	D
3	x
6	y

```
SELECT roads.R, roads.D, ints.D
FROM roads INNER JOIN ints
ON roads.I = ints.I
```



```
SELECT roads.R, roads.D, ints.D
FROM roads, ints
WHERE roads.I = ints.I
```

r.R	r.D	i.D
1	a	x
2	b	x
4	c	y
5	d	y
7	e	x



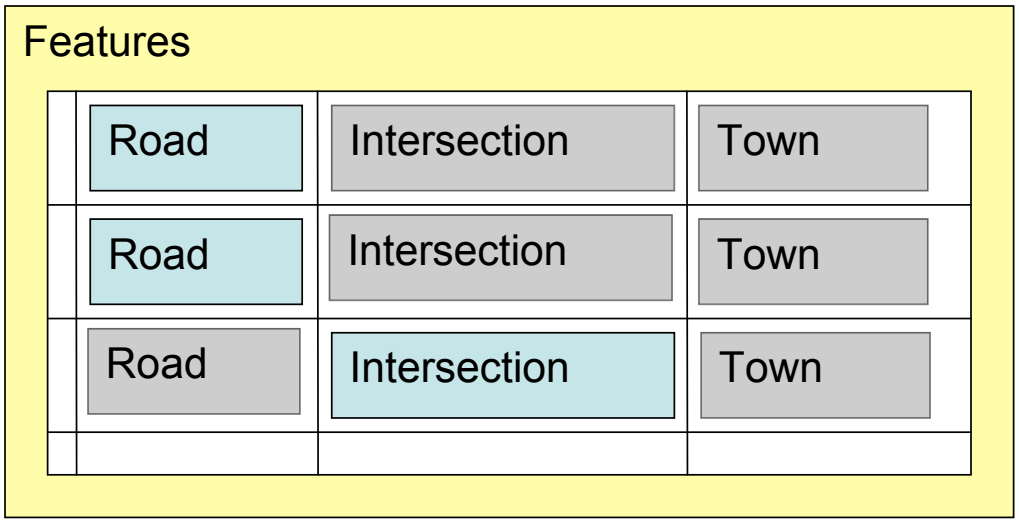
(aka “an Equi Join”)

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 10
MapReduce Design Patterns
IPR2024-00303

Tables vs. Flat File?

Tables

Flat File




```
Message GeoFeature {
  enum Type {
    ROAD = 1;
    INTERSECTION = 2;
    TOWN = 3;
  }
  required Type type = 0;
  optional Road road = 1;
  optional Intersection intersection = 2;
  optional Town town = 3 ;
}
```

“ Protocol Buffer ”

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 11
MapReduce Design Patterns
IPR2024-00303

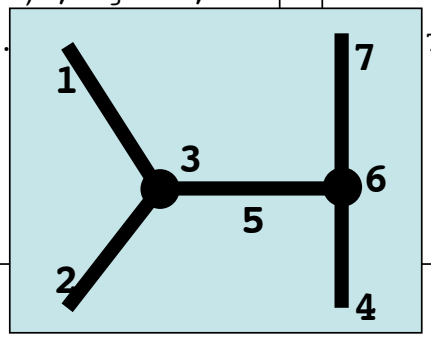
References vs. Duplication?

References

```
1: <type=Road>, <intersections=(3)>, <geom>, …
2: <type=Road>, <intersections=(3)>, <geom>, …
3: <type=Intersection>, <roads=(1,2,5)>, …
4: <type=Road>, <intersections=(6)>, <geom>,
5: <type=Road>, <intersections=(3,6)>, <geom>, …
6: <type=Intersection>, <roads=(5,6,7)>, …
7: <type=Road>, <intersections=(6)>, <geom>, …
8: <type=Town>, <name>, <geom>, .
                        .
                        .
                        .
```

Duplication

```
3: <type=Intersection>, <roads=(
     1: <type=Road>, <geom>, <name>, …
     2: <type=Road>, <geom>, <name>, …
     5: <type=Road>, <geom>, <name>, …)>, …
6: <type=Intersection>, <roads=(
     4: <type=Road>, <geom>, <name>, … >
     5: <type=Road>, <geom>, <name>, … >
     7: <type=Road>, <geom>, <name>, …)>, …
                        .
                        .
                        .
```

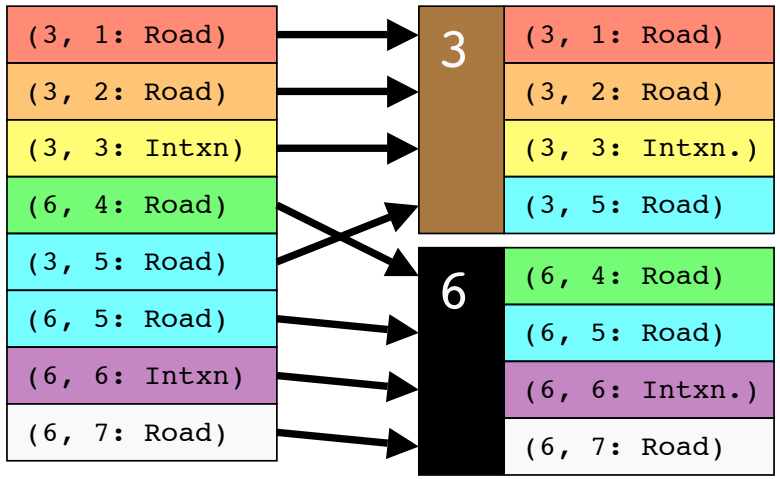


- References: Common primary key; easy restructuring
- Duplication: Avoids additional MR passes; denormalizes data
- …an engineering space / time / complexity tradeoff

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 12
MapReduce Design Patterns
IPR2024-00303

Code Example

```
class IntersectionAssemblerMapper : public
    Mapper {
  virtual void Map(MapInput* input) {
    GeoFeature feature;
    feature.FromMapInput(input);
    if (feature.type()==INTERSECTION) {
      Emit(feature.id(), input);
    } else if (feature.type() == ROAD) {
      Emit(feature.intersection_id(0), input);
      Emit(feature.intersection_id(1), input);
    }
  }
};
REGISTER_MAPPER(IntersectionAssemblerMapper);
```



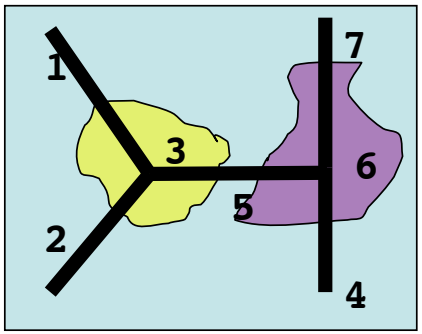

```
class IntersectionAssemblerReducer : public
    Reducer {
  virtual void Reduce(ReduceInput* input) {
    GeoFeature feature;
    GraphIntersection intersection;
    intersection.id = input->key();
    while(!input->done()) {
      feature.FromMapInput(input->value());
      if (feature.type()==INTERSECTION)
        intersection.SetIntersection(feature);
      else
        intersection.AddRoadFeature(feature);
      input->next();
    }
    Emit(intersection);
  }
};
REGISTER_REDUCER(IntersectionAssemblerReducer);
```

Join, but no pointers or keys?

Input	**Map**	Shuffle	**Reduce**	Output
List of items	Apply **map()** to each; emit (key,val) pairs	Sort by key	Apply **reduce()** to list of pairs with same key	New list of items

1: Road
2: Road
3: Town
4: Road
5: Road
6: Town
7: Road

3: 1,2,5
6: 4,5,7



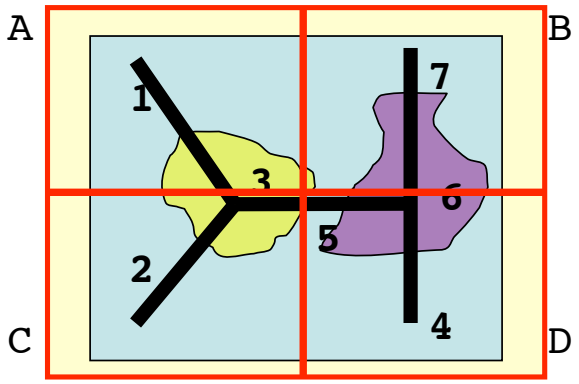
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 14
MapReduce Design Patterns
IPR2024-00303

Bucketing (or) Grace Hash Join

Input	Map	Shuffle	Reduce	Output
Feature List	Emit (key, item) pair Key = geometric hash Secondary key = Type	Sort by keys	Intersect all towns with all roads; emit intersecting pairs	(town, road) pair

1: Road
2: Road
3: Town
4: Road
5: Road
6: Town
7: Road

(A-Road, 1)
(C-Road, 1)
(C-Road, 2)
(A-Town, 3)
(B-Town, 3)
(C-Town, 3)
(D-Road, 4)
(C-Road, 5)
(D-Road, 5)
(B-Town, 6)
(D-Town, 6)
(B-Road, 7)
(D-Road, 7)



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 15
MapReduce Design Patterns
IPR2024-00303

Reduce on Key A

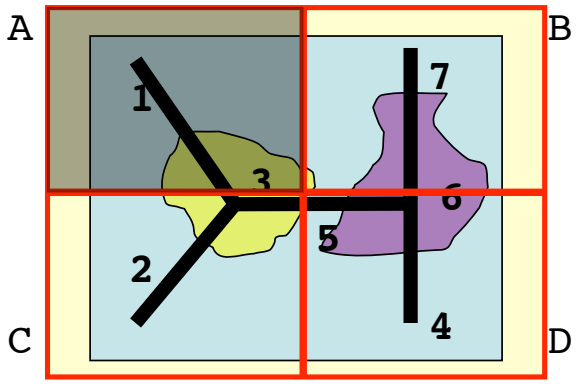
Input	Map	Shuffle	Reduce	Output
Feature List	Emit (key, item) pair Key = geometric hash Secondary key = Type	Sort by keys	Intersect all towns with all roads; emit intersecting pairs	(town, road) pair

1: Road
2: Road
3: Town
4: Road
5: Road
6: Town
7: Road

(A-Road, 1)
(C-Road, 1)
(C-Road, 2)
(A-Town, 3)
(B-Town, 3)
(C-Town, 3)
(D-Road, 4)
(C-Road, 5)
(D-Road, 5)
(B-Town, 6)
(D-Town, 6)
(B-Road, 7)
(D-Road, 7)

(A-Town, 3)
(A-Road, 1)

(3, 1)



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 16
MapReduce Design Patterns
IPR2024-00303

Reduce on Key B

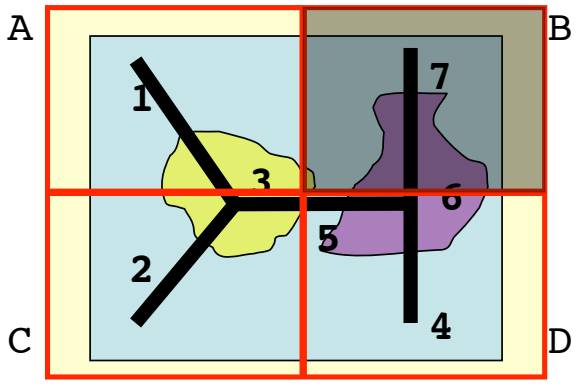
Input	Map	Shuffle	Reduce	Output
Feature List	Emit (key, item) pair Key = geometric hash Secondary key = Type	Sort by keys	Intersect all towns with all roads; emit intersecting pairs	(town, road) pair

1: Road
2: Road
3: Town
4: Road
5: Road
6: Town
7: Road

(A-Road, 1)
(C-Road, 1)
(C-Road, 2)
(A-Town, 3)
(B-Town, 3)
(C-Town, 3)
(D-Road, 4)
(C-Road, 5)
(D-Road, 5)
(B-Town, 6)
(D-Town, 6)
(B-Road, 7)
(D-Road, 7)

(B-Town, 3)
(B-Town, 6)
(B-Road, 7)

(6, 7)



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 17
MapReduce Design Patterns
IPR2024-00303

Reduce on Key C

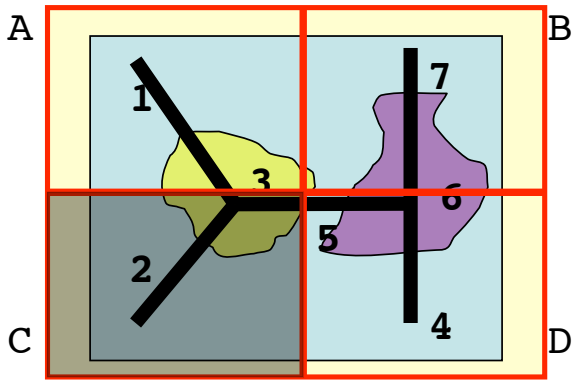
Input	Map	Shuffle	Reduce	Output
Feature List	Emit (key, item) pair Key = geometric hash Secondary key = Type	Sort by keys	Intersect all towns with all roads; emit intersecting pairs	(town, road) pair

1: Road
2: Road
3: Town
4: Road
5: Road
6: Town
7: Road

(A-Road, 1)
(C-Road, 1)
(C-Road, 2)
(A-Town, 3)
(B-Town, 3)
(C-Town, 3)
(D-Road, 4)
(C-Road, 5)
(D-Road, 5)
(B-Town, 6)
(D-Town, 6)
(B-Road, 7)
(D-Road, 7)

(C-Town, 3)
(C-Road, 1)
(C-Road, 5)
(C-Road, 2)

(3, 1)
(3, 2)
(3, 5)



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 18
MapReduce Design Patterns
IPR2024-00303

Reduce on Key D

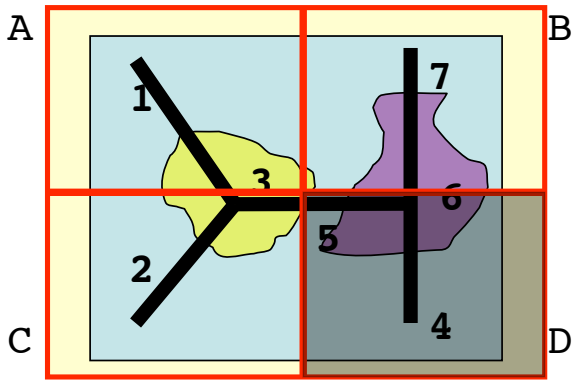
Input	**Map**	Shuffle	**Reduce**	Output
Feature List	Emit (key, item) pair Key = geometric hash Secondary key = Type	Sort by keys	Intersect all towns with all roads; emit intersecting pairs	(town, road) pair

1: Road
2: Road
3: Town
4: Road
5: Road
6: Town
7: Road

(A-Road, 1)
(C-Road, 1)
(C-Road, 2)
(A-Town, 3)
(B-Town, 3)
(C-Town, 3)
(D-Road, 4)
(C-Road, 5)
(D-Road, 5)
(B-Town, 6)
(D-Town, 6)
(B-Road, 7)
(D-Road, 7)

(D-Town, 6)
(D-Road, 4)
(D-Road, 5)
(D-Road, 7)

(6, 4)
(6, 5)
(6, 7)



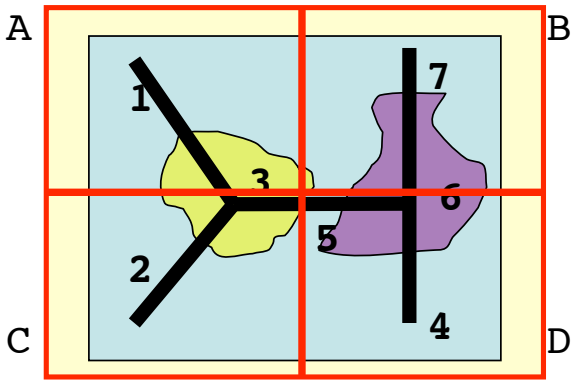
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 19
MapReduce Design Patterns
IPR2024-00303

Output… not quite...

Input	Map	Shuffle	Reduce	Output
Feature List	Apply **map()** to each; emit (key,val) pairs	Sort by key	Apply **reduce()** to list of pairs with same key	List of items

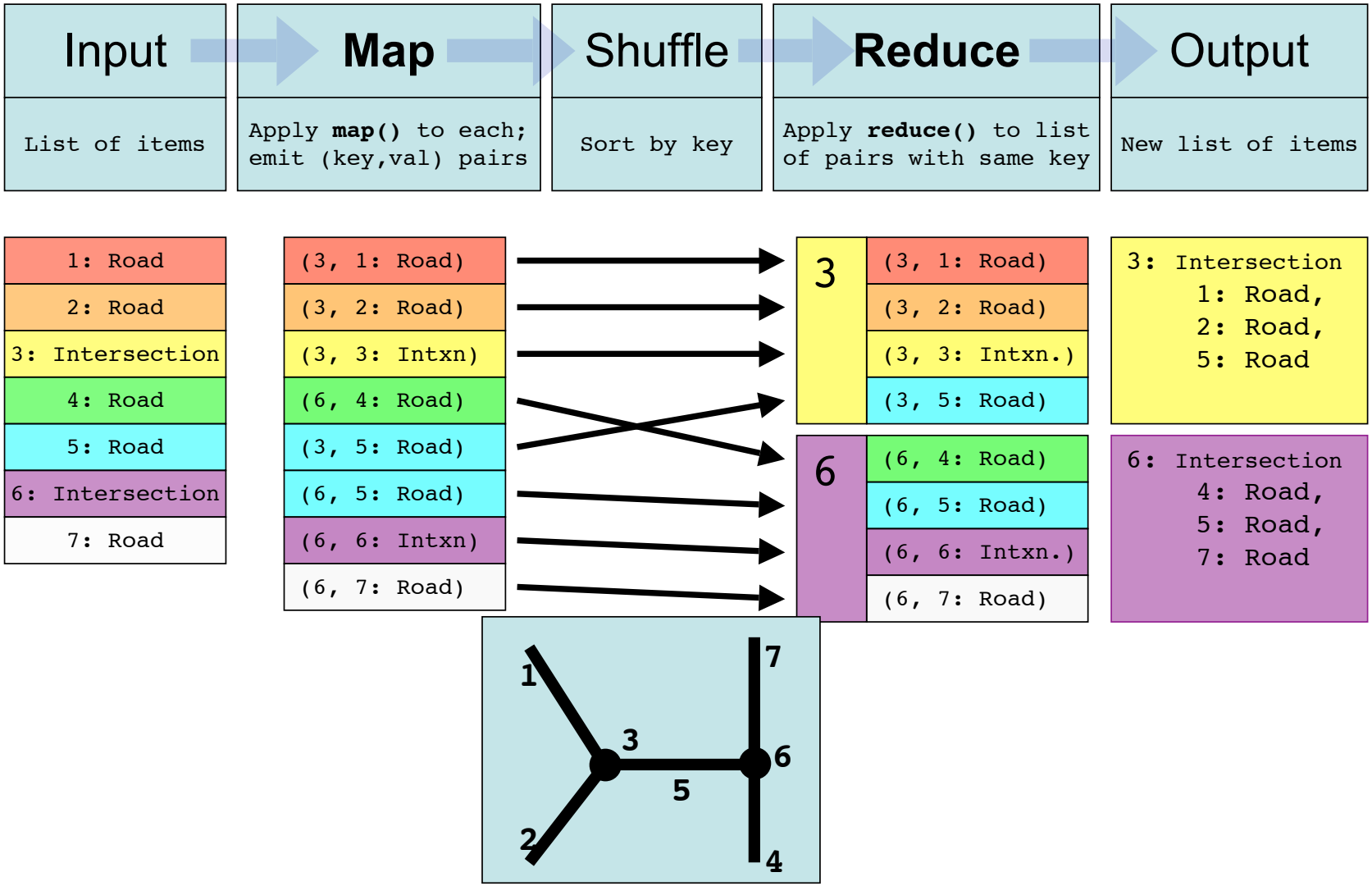
1: Road
2: Road
3: Town
4: Road
5: Road
6: Town
7: Road

(3, 1)
(6, 7)
(3, 1)
(3, 2)
(3, 5)
(6, 7)
(6, 4)
(6, 5)



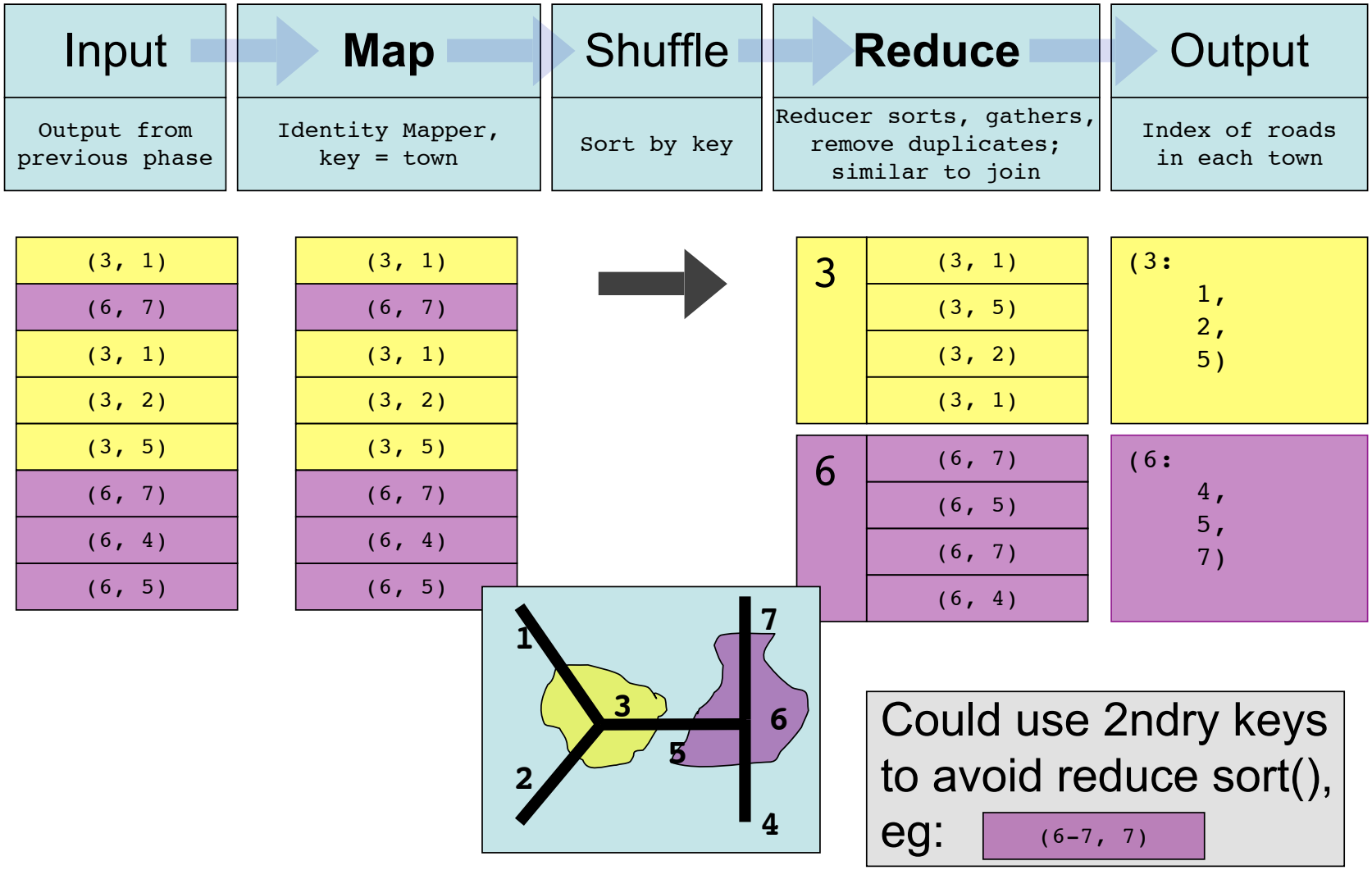
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 20
MapReduce Design Patterns
IPR2024-00303

…recall earlierJoin Pattern



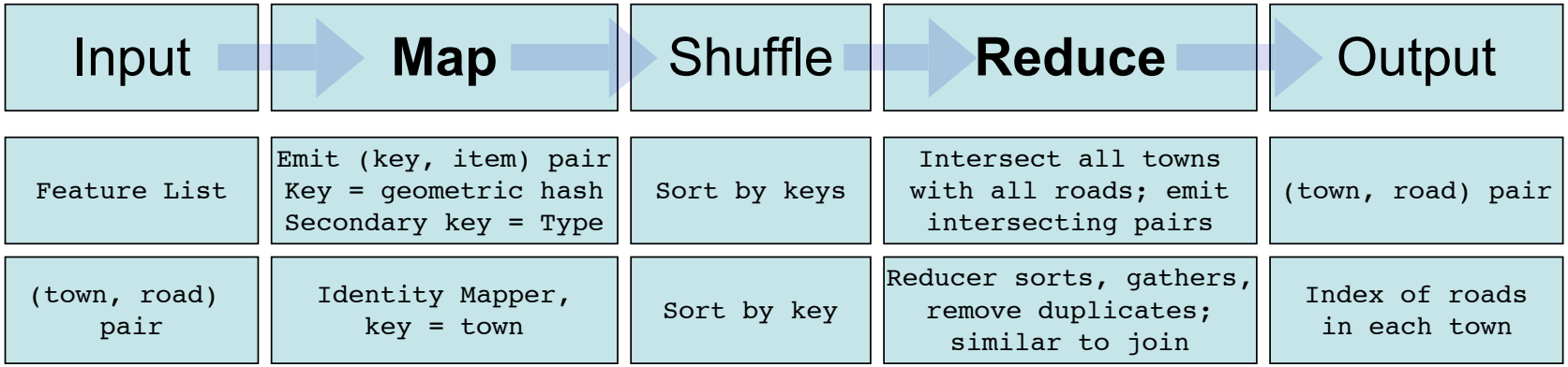
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 21
MapReduce Design Patterns
IPR2024-00303

Recursive Key Join Pattern



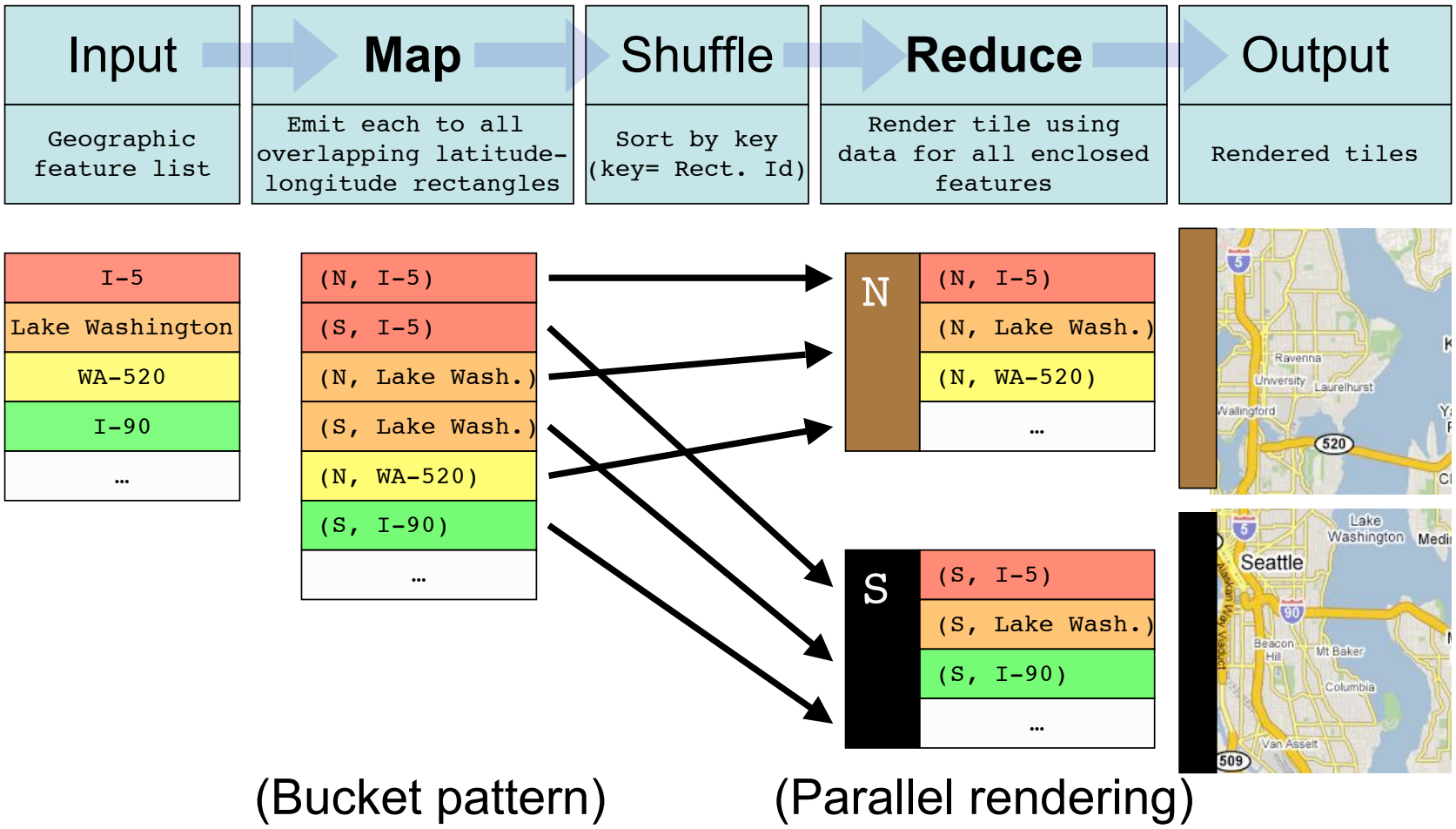
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 22
MapReduce Design Patterns
IPR2024-00303

Chained MapReduce's Pattern



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 23
MapReduce Design Patterns
IPR2024-00303

Distributing Costly Computation: e.g. Rendering Map Tiles



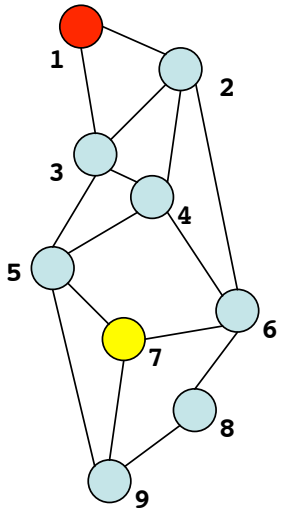
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 24
MapReduce Design Patterns
IPR2024-00303

Finding Nearest Points Of Interest (POIs)

Feature List

```
1, Type, Road, Intersection, …
2, Type, Road, Intersection, …
3, Type, Road, Intersection, …
 . . .
```

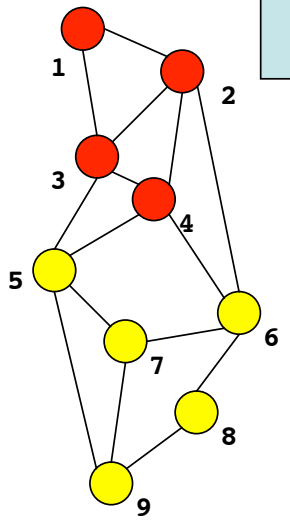
Input



Nearest POI within 5mi of Intersection

```
(1, 1)
(2, 1)
(3, 1)
(4, 1)
(5, 7)
(6, 7)
(7, 7)
(8, 7)
(9, 7)
```

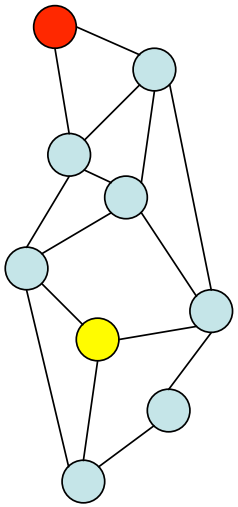
Output



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 25
MapReduce Design Patterns
IPR2024-00303

Finding Nearest POI on a Graph

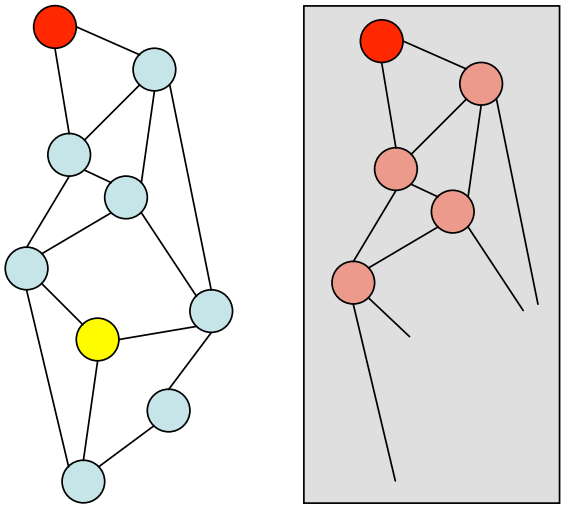
Input	**Map**	Shuffle	**Reduce**	Output
Graph	Perform Dijkstra from each POI node. Emit POI & dist. At each node in search.		For each node, emit closest POI	Nodes with nearest POIs



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 26
MapReduce Design Patterns
IPR2024-00303

Finding Nearest POI on a Graph

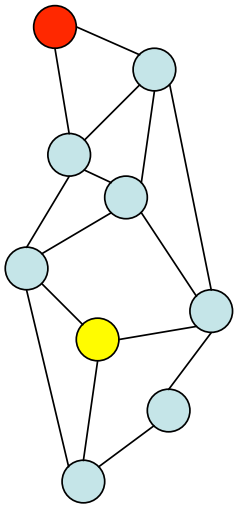
Input	**Map**	Shuffle	**Reduce**	Output
Graph	Perform Dijkstra from each POI node. Emit POI & dist. At each node in search.		For each node, emit closest POI	Nodes with nearest POIs

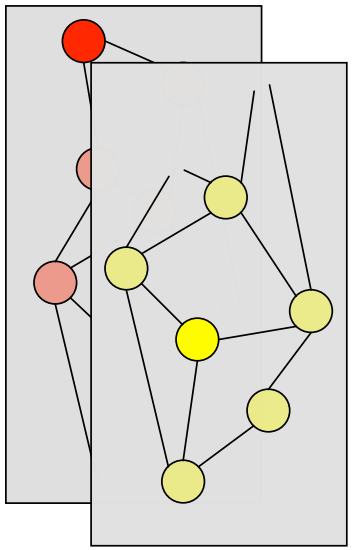


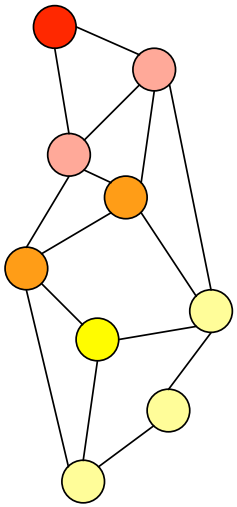
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 27
MapReduce Design Patterns
IPR2024-00303

Finding Nearest POI on a Graph

Input	**Map**	Shuffle	**Reduce**	Output
Graph	Perform Dijkstra from each POI node. Emit POI & dist. At each node in search.		For each node, emit closest POI	Nodes with nearest POIs

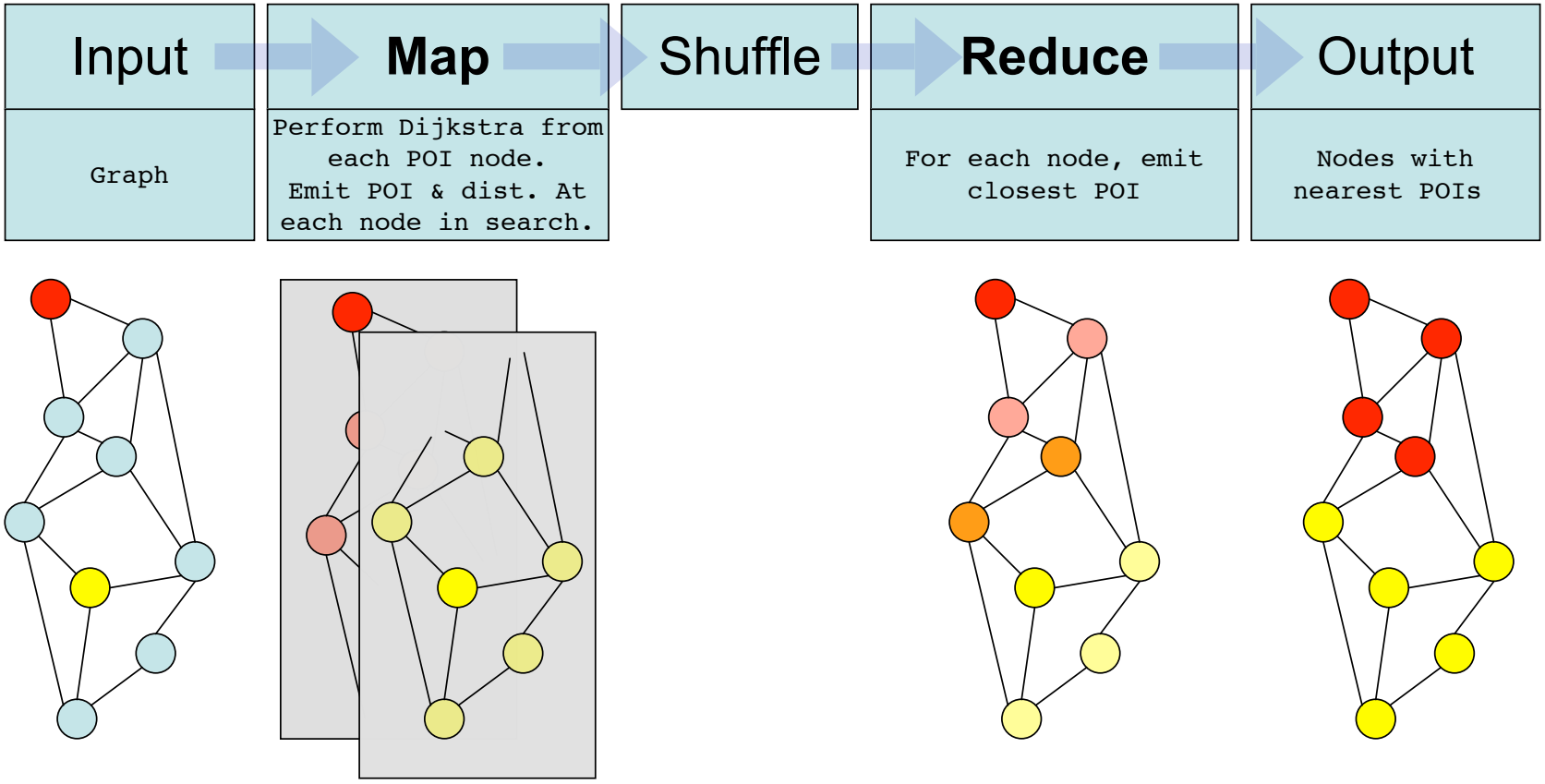






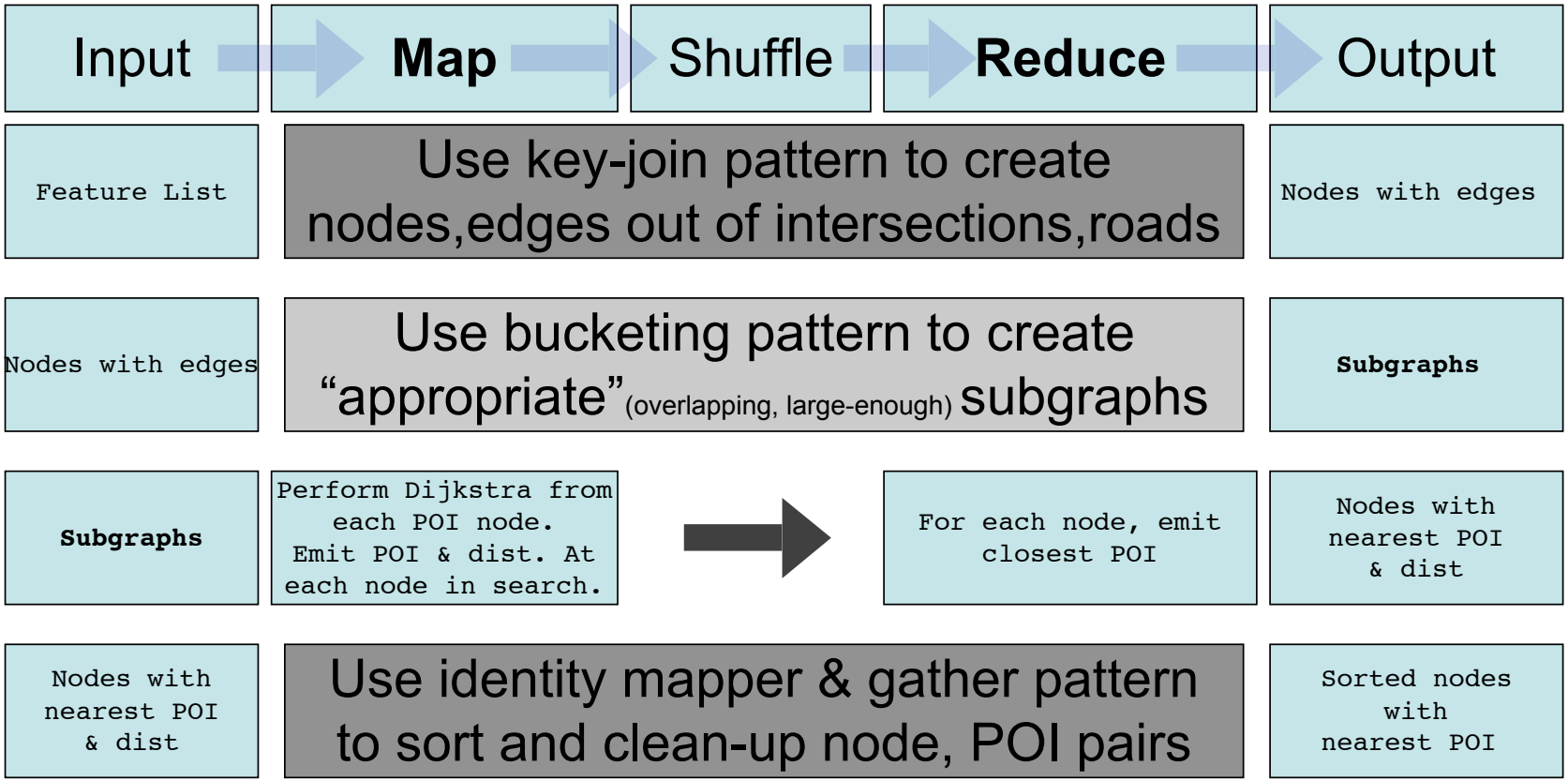
Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 28
MapReduce Design Patterns
IPR2024-00303

Finding Nearest POI on a Graph



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 29
MapReduce Design Patterns
IPR2024-00303

Putting it all together: Nearest POI



Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 30
MapReduce Design Patterns
IPR2024-00303

Hard Problems for MapReduce

- Following multiple pointer hops
- Iterative algorithms
- Algorithms with global state
- Operations on graphs without good embeddings

- [insert your favorite challenge here]

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 31
MapReduce Design Patterns
IPR2024-00303

Summary

MapReduce eases:

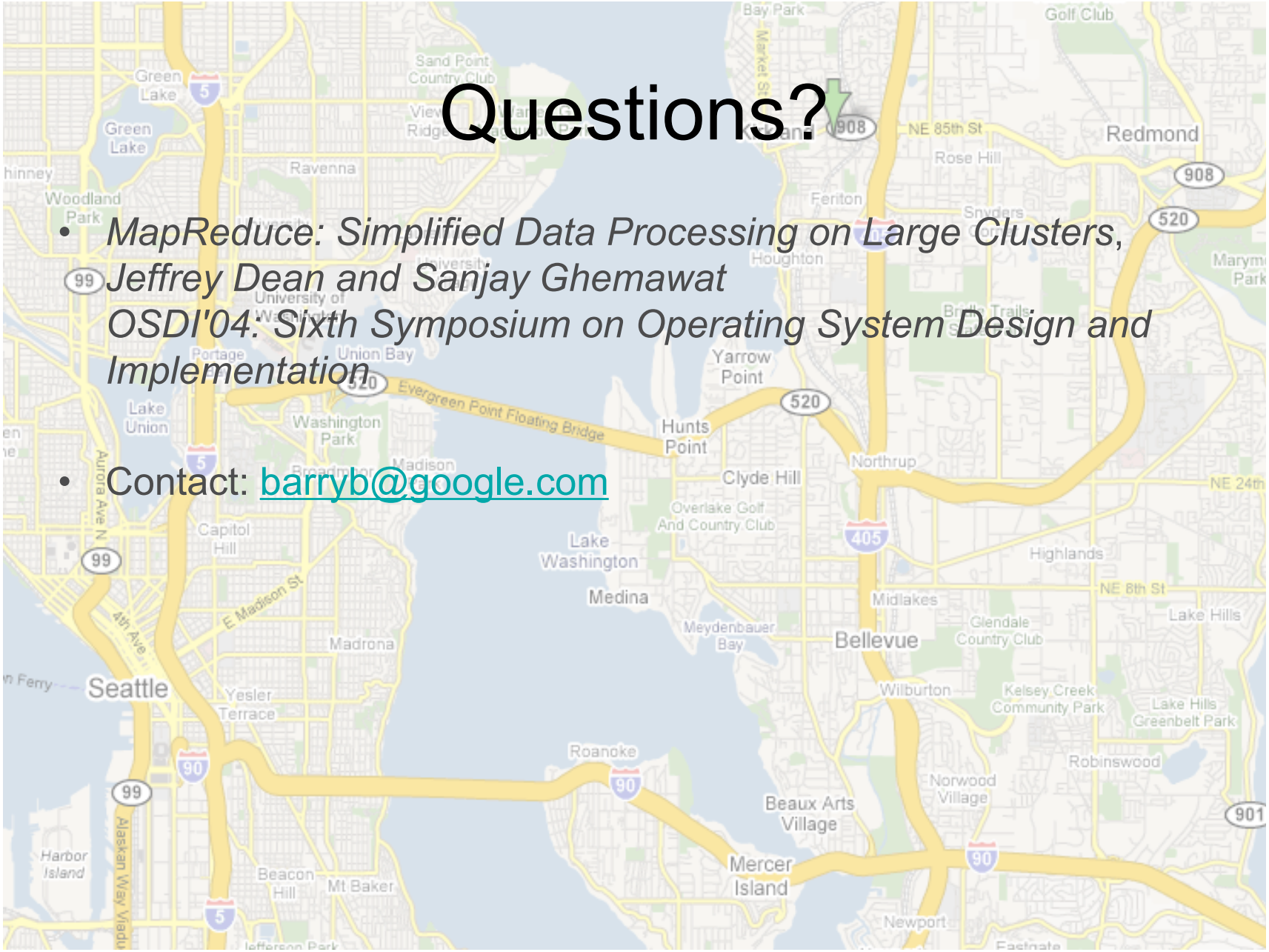
- Machine coordination
- Network communication
- Fault tolerance
- Scaling
- Productivity

MapReduce patterns:

- “Flat” data structures
- Foreign / Recursive Key Joins (aka pointer following)
- Hash Joins (aka bucketing)
- Distribute $$ computation
- Chain MapReduce phases
- Simplify Reduce() by using secondary keys
- [insert your pattern here]

Cloudera, Inc. v. R2 Solutions LLC
R2 Solutions LLC Ex. 2003 - 32
MapReduce Design Patterns
IPR2024-00303

Questions?



- *MapReduce: Simplified Data Processing on Large Clusters*, *Jeffrey Dean and Sanjay Ghemawat*
  OSDI'04: Sixth Symposium on Operating System Design and Implementation
- Contact: barryb@google.com