Alternative Approaches to Parallel GIS Processing

Nathan Kerr

Objective

Evaluate the Map Reduce and Message Passing paradigms of parallel processing for capability in parallel GIS processing.

Outline

- GIS Processing
- Alternative Approaches
- Evaluation
- Conclusion

GIS Processing

GIS Processing

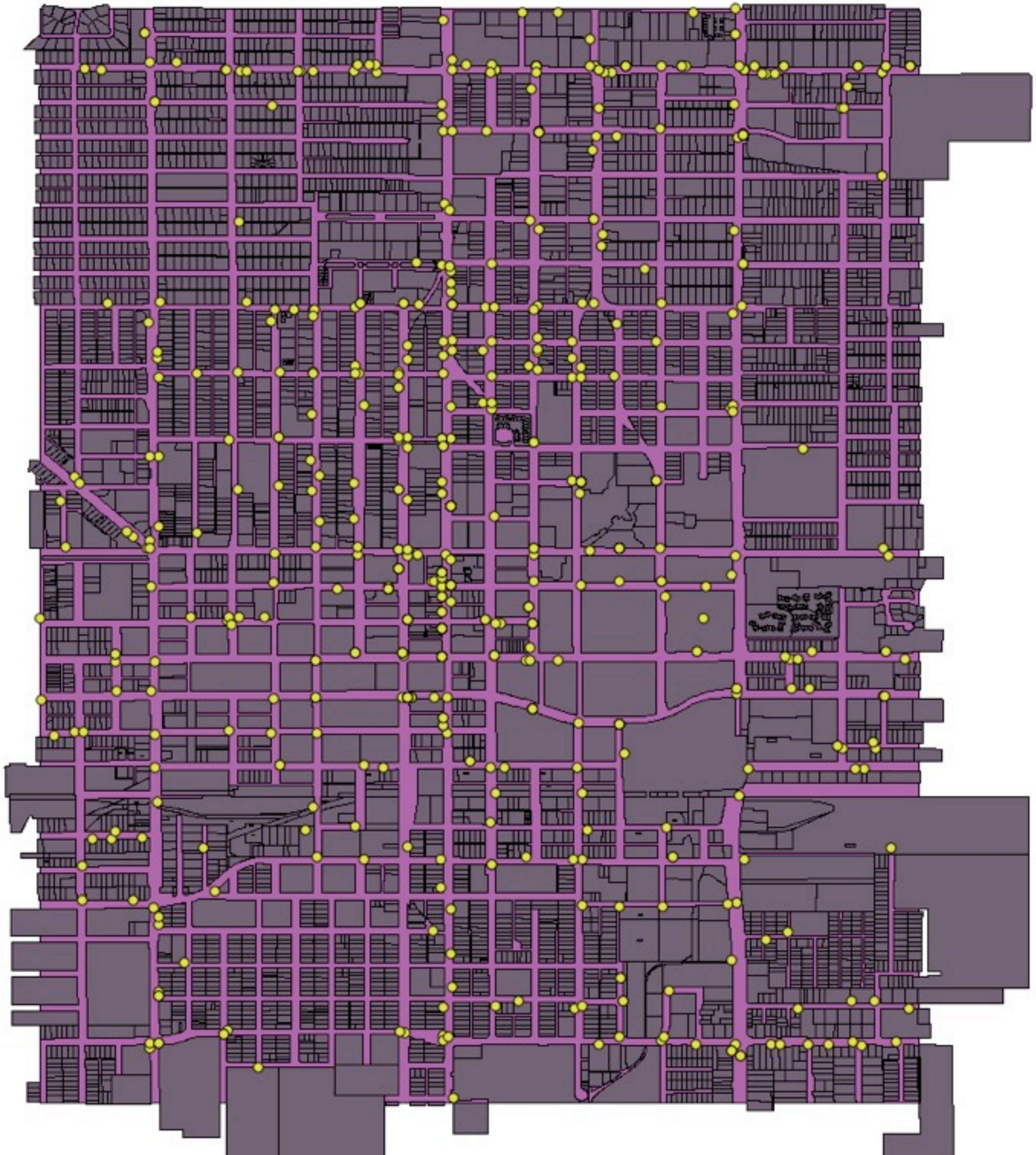
- Geographic Information Systems
- Used to represent real-world objects
- Enables sophisticated analysis of the world around us

GIS Data Growth

- Maricopa County in 2000
  - 9,224 square miles
  - 1.2 million parcels of land
  - 34 thousand employers
  - Other data sets available

Find the Nearest Parcel

- Employer locations stored using street address
- Parcels use lat/long
- Which parcel of land is that employer on?



Find the Nearest Parcel

- Criteria for matching:
  - Geographically closest parcel
  - Compatible land use code

Processing Time

Comparison Speed	Processing Time
1 million/second	11 hours 37 minutes
10 million/second	1 hour 10 minutes
100 billion/second	0.5 seconds

41,784,295,260 comparisons to make

PostGIS

- Reference for traditional GIS processing methods
- Built on PostgreSQL, an open source relation database (RDBMS)
- Provides GIS datatypes and processing functions

Alternative Approaches

Alternative Approaches

- Map Reduce
- Message Passing

Map Reduce

- Two phase approach
- Map applies a function to every record in the input dataset
- Reduce aggregates and processes the output of the map phase

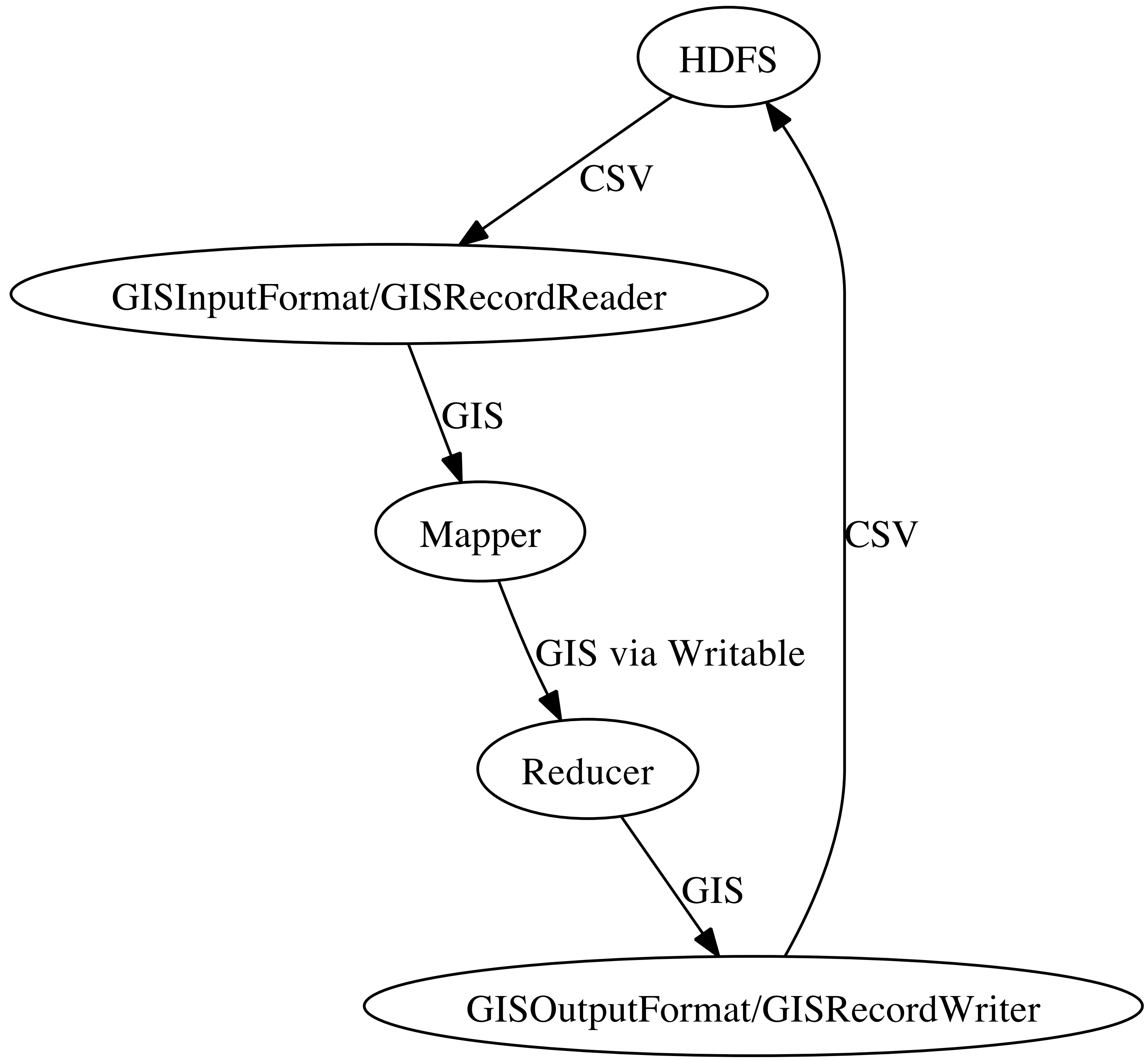
Hadoop

- Open source Map Reduce Framework
- Java
- Used by:
  - Yahoo!
  - Amazon
  - Facebook

HadoopGIS

- Adds a GIS datatype to Hadoop
- Uses the Java Topology Suite (JTS) for geometric calculations

HadoopGIS Data Flow



Nearest Parcel

- map (input: a single employer)
  - Find the parcel with:
    - Minimum distance
    - Matching land use code
- reduce (input: a set of employers)
  - Output every input (identity)

Nearest Parcel

- Each mapper has a complete copy of the parcel dataset
- Memory limitations limit the size of the parcel dataset

Message Passing

- Creates a task for each participating processor
- Each task has a unique number
- Tasks can collaborate with each other by passing messages

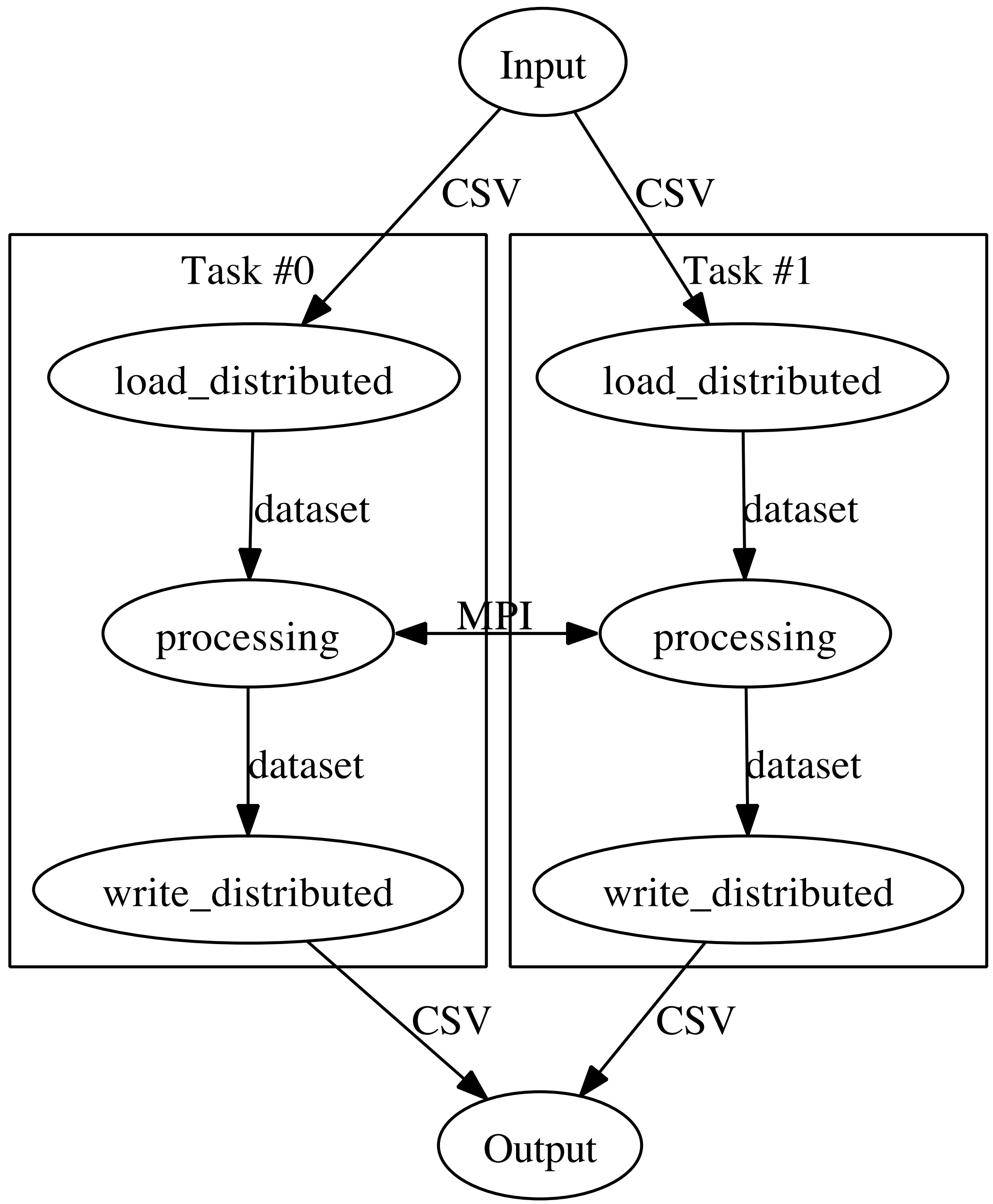
Message Passing Interface (MPI)

- MPI is a standard, many implementations
- C, C++, Fortran
- Used by:
  - CERN (European Nuclear Research)
  - NASA
  - U.S. Airforce, Army, Navy

ClusterGIS

- Library of common operations
  - loading data
  - writing data
- Uses the GEOS library for geometric calculations

ClusterGIS Data Flow



Nearest Parcel

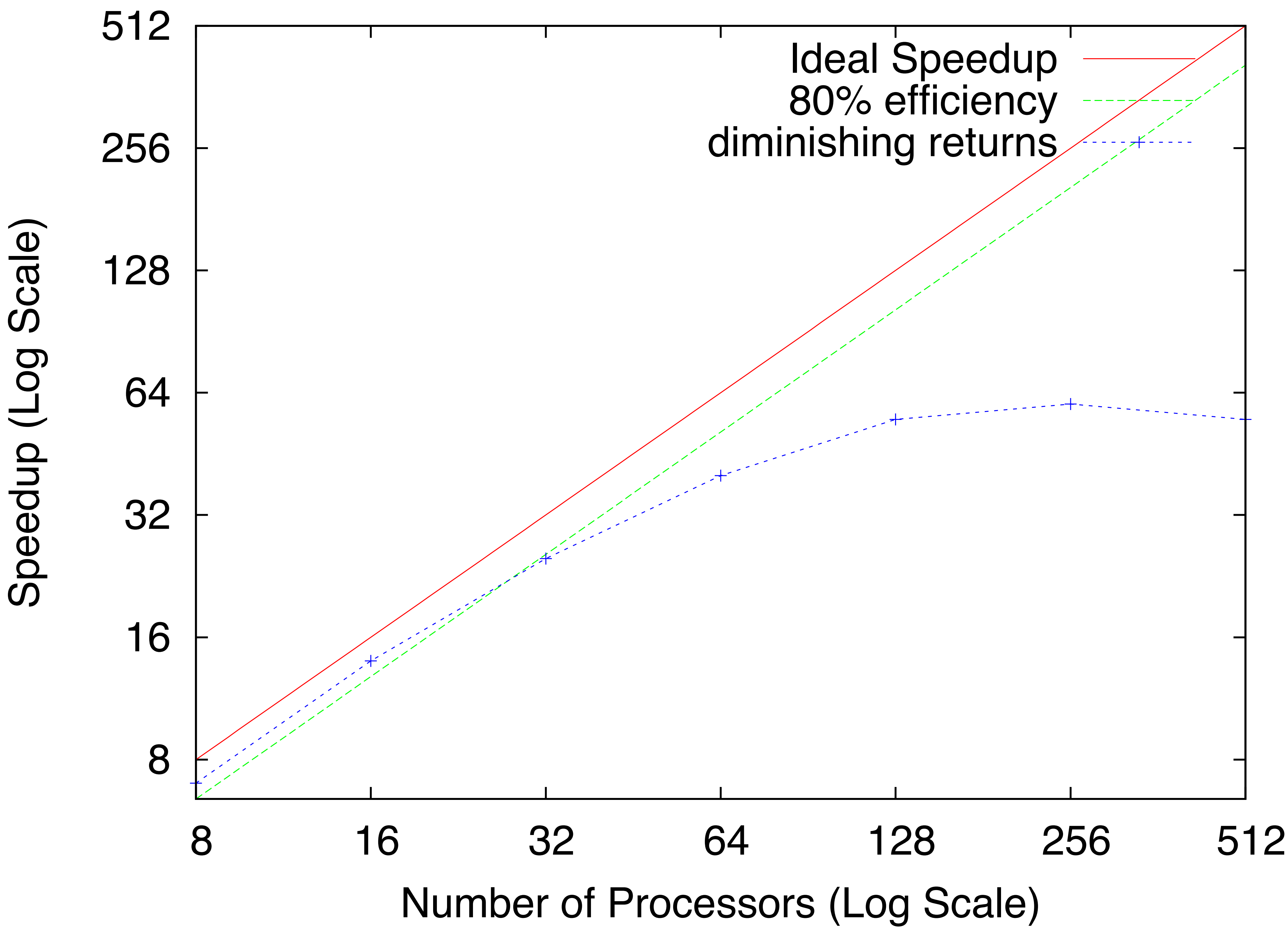
- Tasks are grouped by computer (8 tasks)
- Each computer has a full copy of the parcel dataset, each task has 1/8 of the parcels
- Every task on a computer has the same set of employers

Nearest Parcel

- Each task on a computer finds the nearest parcel among the parcels it has (local min)
- All the tasks on a computer collaborate to find out which of the locally nearest parcels is the nearest parcel

Evaluation

Speedup = t(1)/t(n)

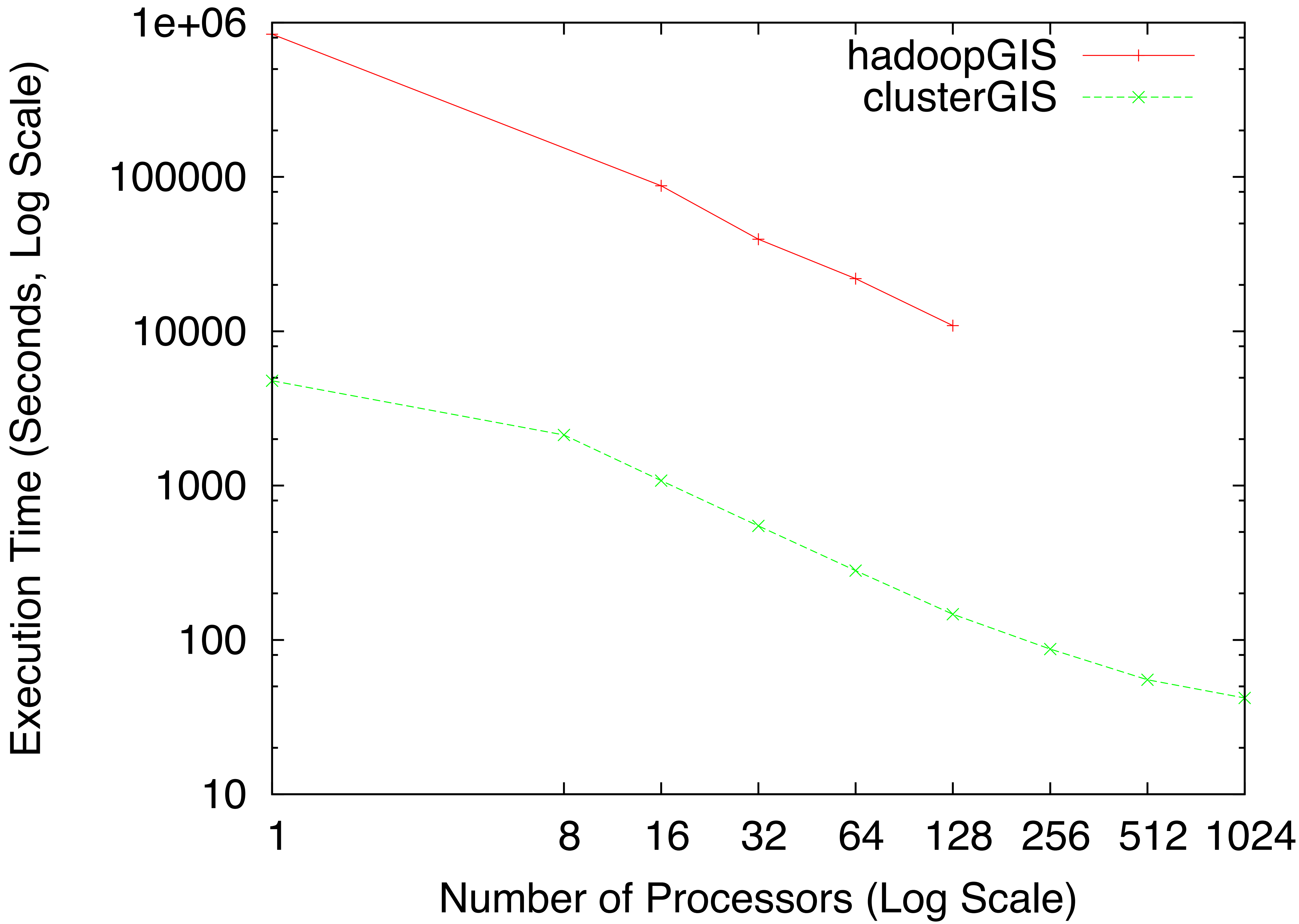


Execution Time

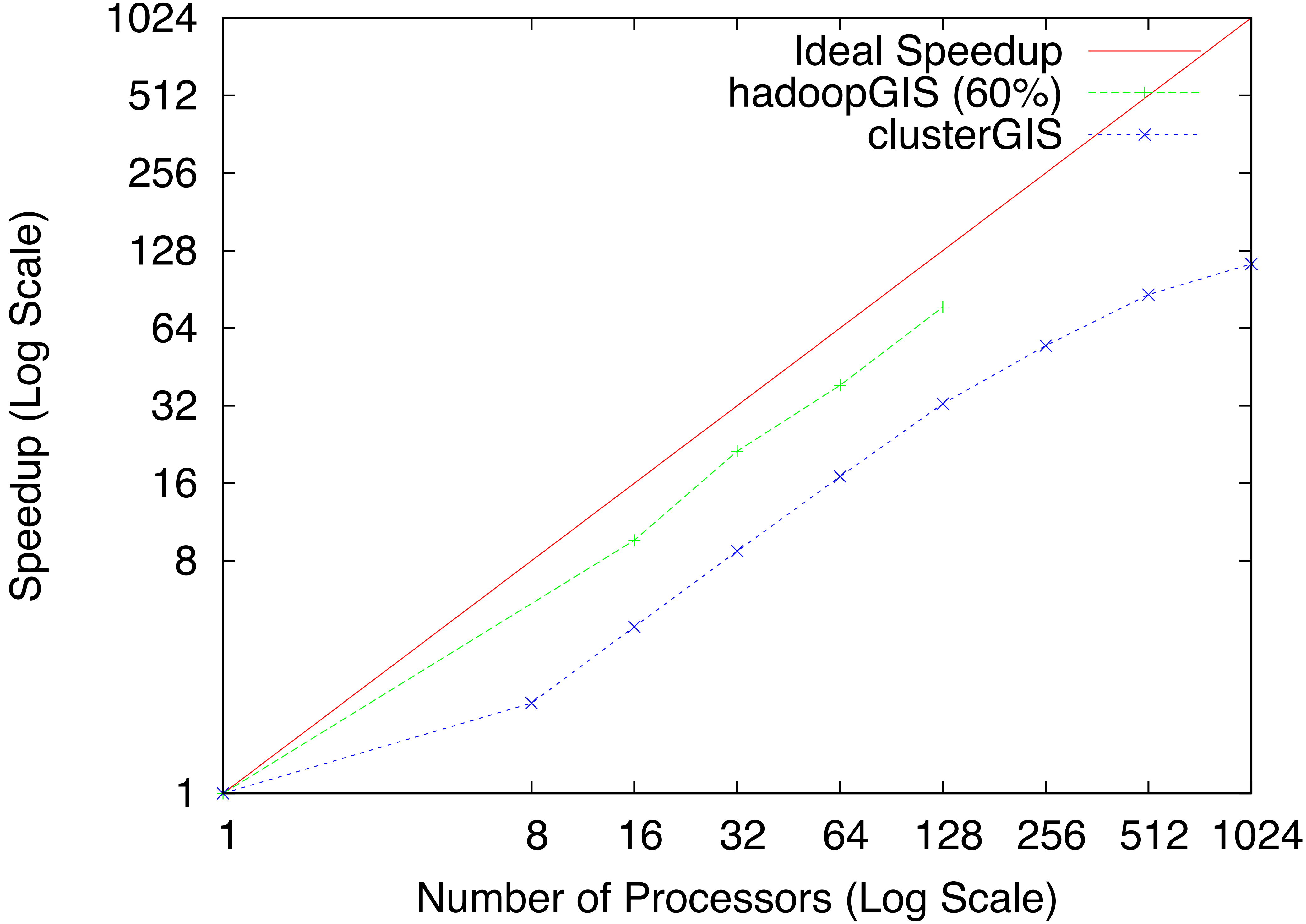
Cores	HadoopGIS	ClusterGIS
1	estimated 9d17h	1 hour 19 min
8	does not run	36 minutes
16	1 day 20 min	18 minutes
32	11 hours	9 minutes
64	6 hours	4 min 40 sec
128	3 hours	2 min 26 sec
256	does not run	1 min 27 sec
512	does not run	55 seconds
1024	does not run	42 seconds

PostGIS: 55 hours

Execution Time



Nearest Parcel Speedup



Conclusion

Conclusions

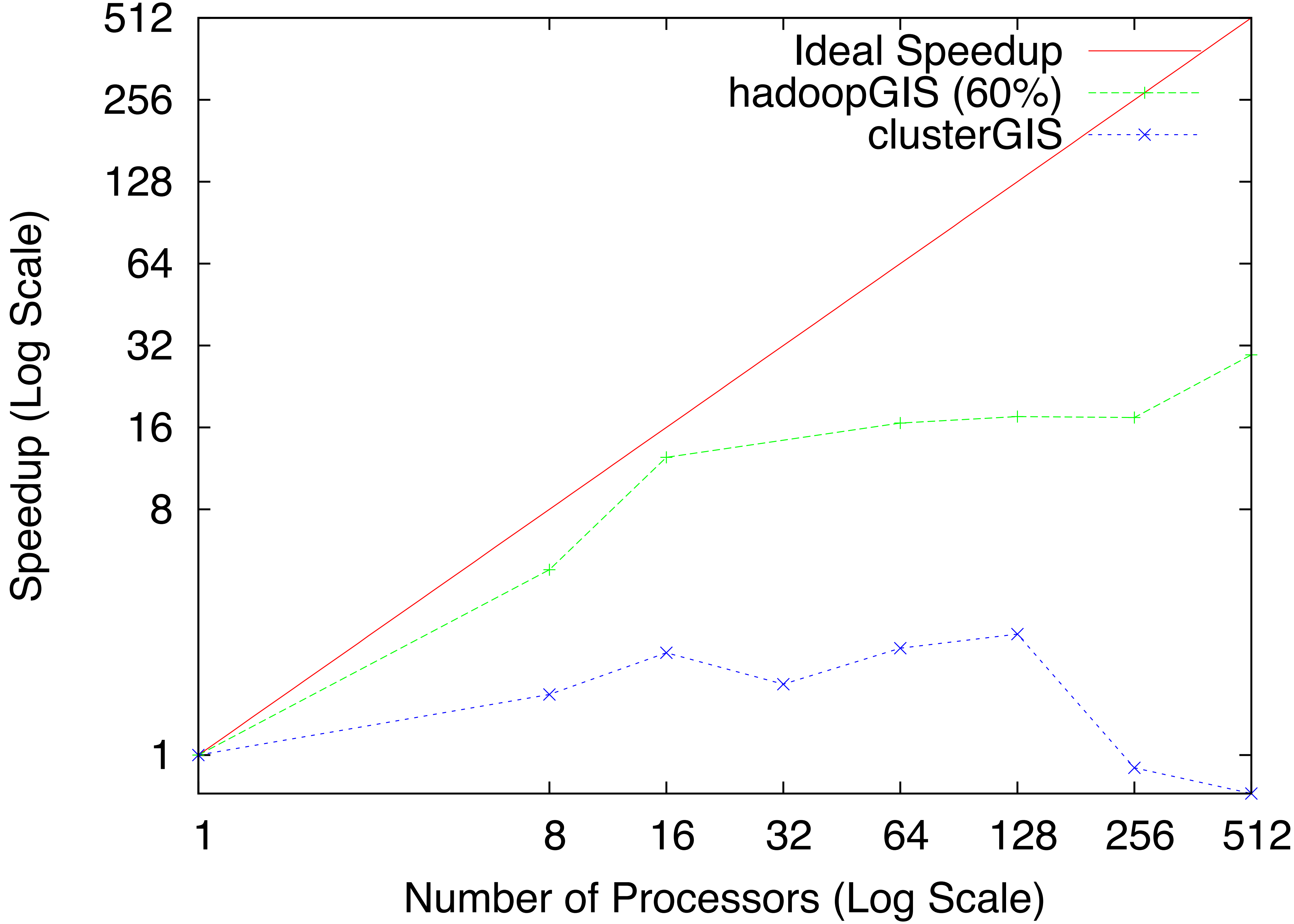
- Both Map Reduce and Message Passing are viable ways to perform GIS processing
- Map Reduce has problems scaling the secondary dataset
- Message Passing is flexible enough to handle alternative ways of scaling

Questions?

Create Times

Implementation	Time (seconds)	Cores
PostGIS	0.012	1
HadoopGIS	77.453	512
ClusterGIS	6.160	128

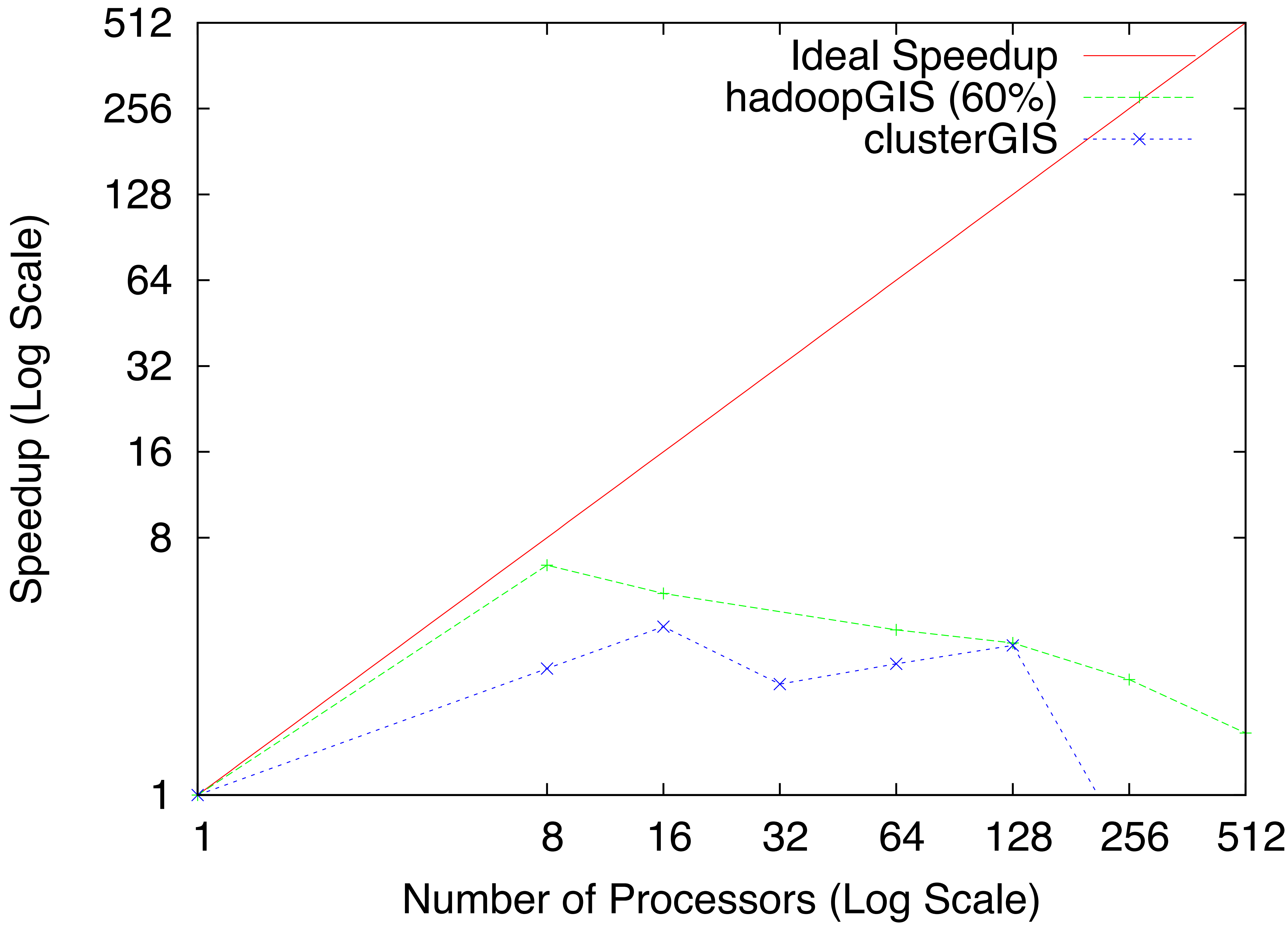
Create Speedup



Read Times

Implementation	Time (seconds)	Cores
PostGIS	0.023	1
HadoopGIS	49.324	8
ClusterGIS	2.015	16

Read Speedup



Filter Times

Implementation	Time (seconds)	Cores
PostGIS	1.123	1
HadoopGIS	50.408	8
ClusterGIS	2.957	128

Filter Speedup