GeoTrellis

Adding Geospatial Capabilities to Spark

Rob Emanuele
lossyrob@gmail.com

@lossyrob

azavea

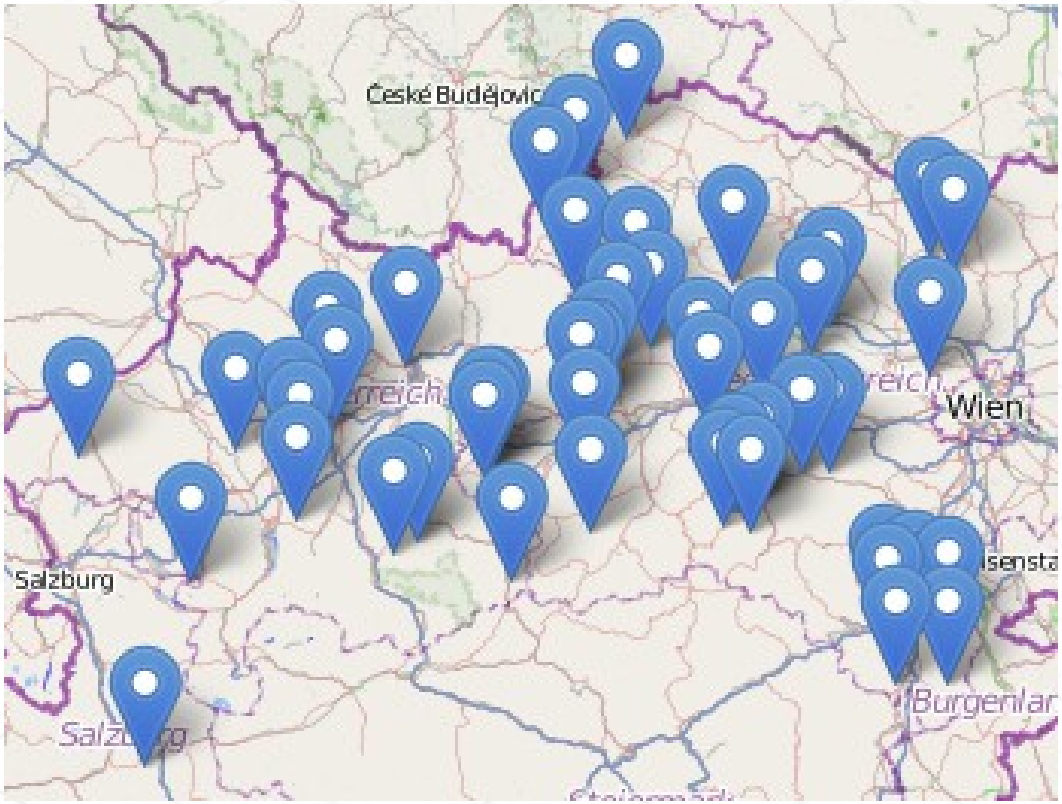
LocationTech

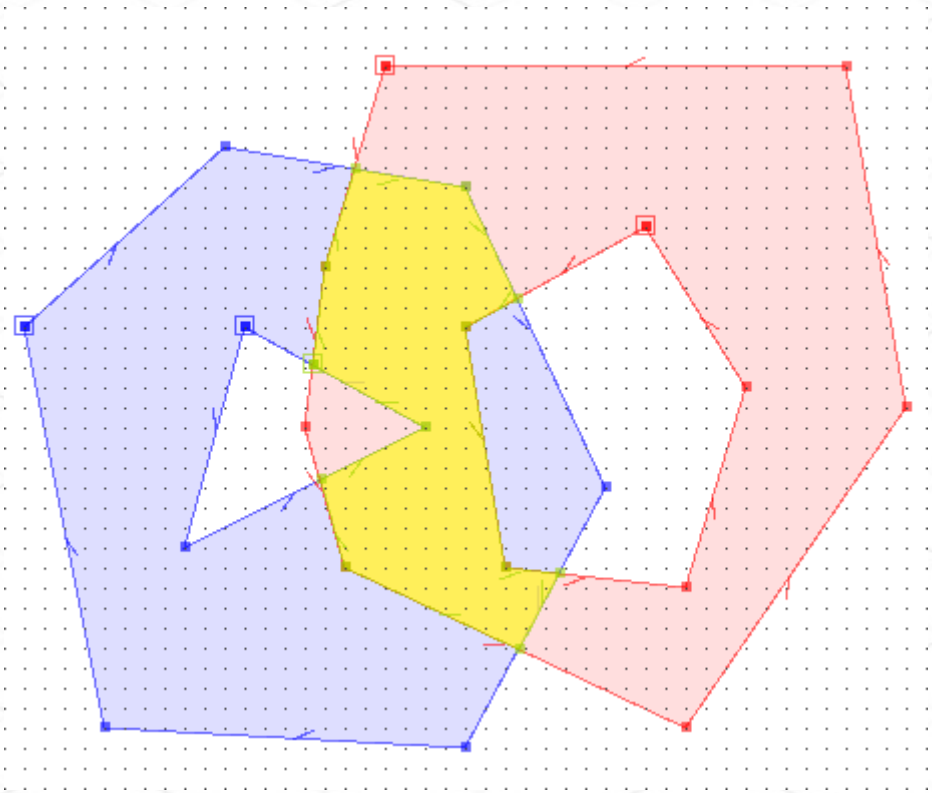
GeoTrellis

What is GeoTrellis?

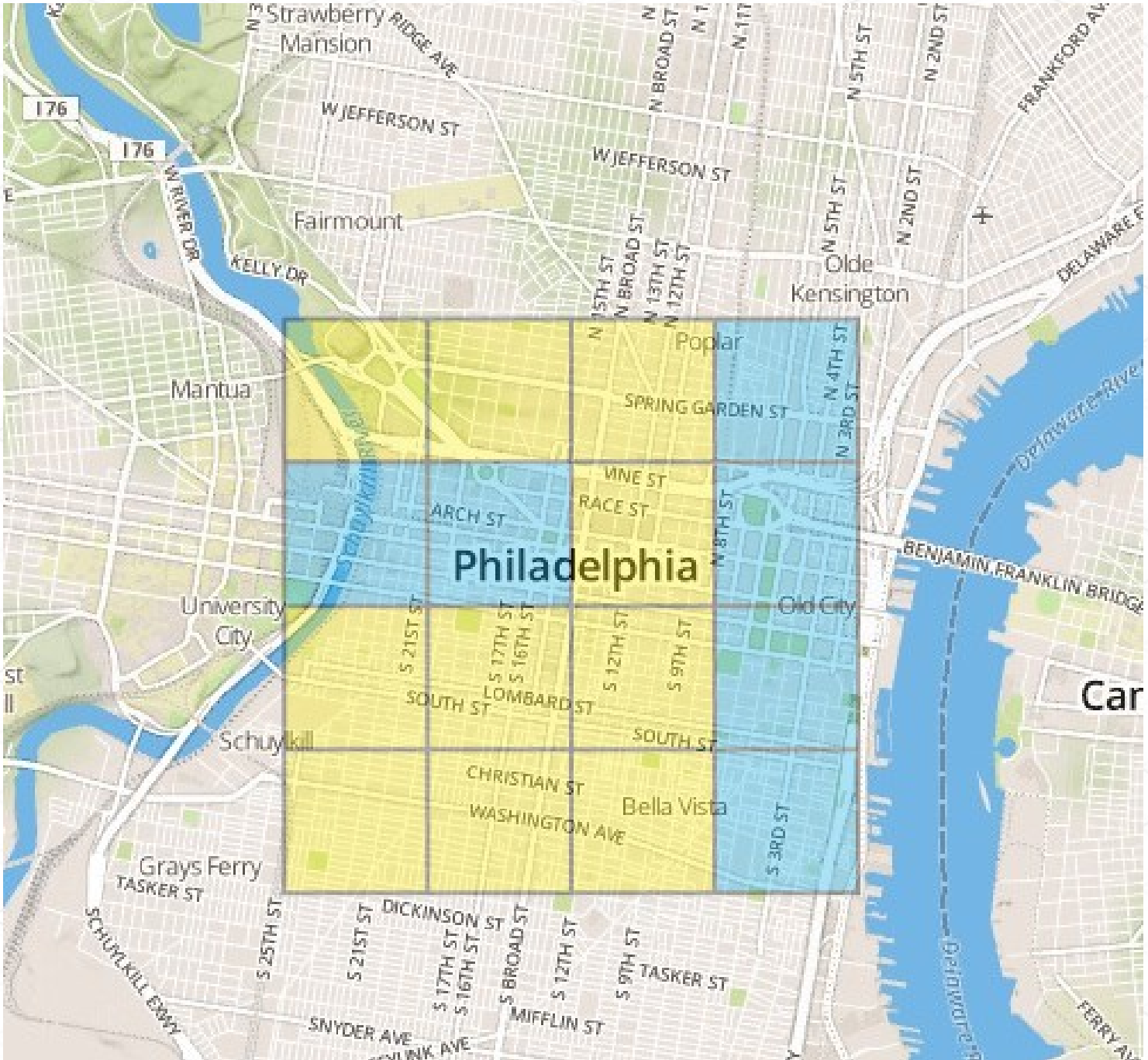
- Geospatial data types\IO (raster & vector)
- Raster operations (Map Algebra)
- Utilities for creating web services

Vector Data





GeoTrellis


Philadelphia
Strawberry Mansion
Fairmount
Mantua
Poplar
Olde Kensington
University City
Old City
Schuylkill
Grays Ferry
Bella Vista
Car
W JEFFERSON ST
N BROAD ST
N 5TH ST
N 2ND ST
FRANKFORD AVE
W RIVER DR
KELLY DR
N 15TH ST
N 13TH ST
N 12TH ST
N 4TH ST
N 3RD ST
SPRING GARDEN ST
VINE ST
RACE ST
ARCH ST
N 8TH ST
BENJAMIN FRANKLIN BRIDGE
S 21ST ST
S 17TH ST
S 16TH ST
S 12TH ST
S 9TH ST
LOMBARD ST
SOUTH ST
CHRISTIAN ST
WASHINGTON AVE
S 3RD ST
TASKER ST
DICKINSON ST
S 25TH ST
S BROAD ST
MIFFLIN ST
SNYDER AVE
SCHUYLKILL EXPY
FERRY AVE
176

NED (National Elevation Dataset)

GeoTrellis

What is GeoTrellis?

- Geospatial data types\IO (raster & vector)
- Raster operations (Map Algebra)
- Utilities for creating web services

GeoTrellis

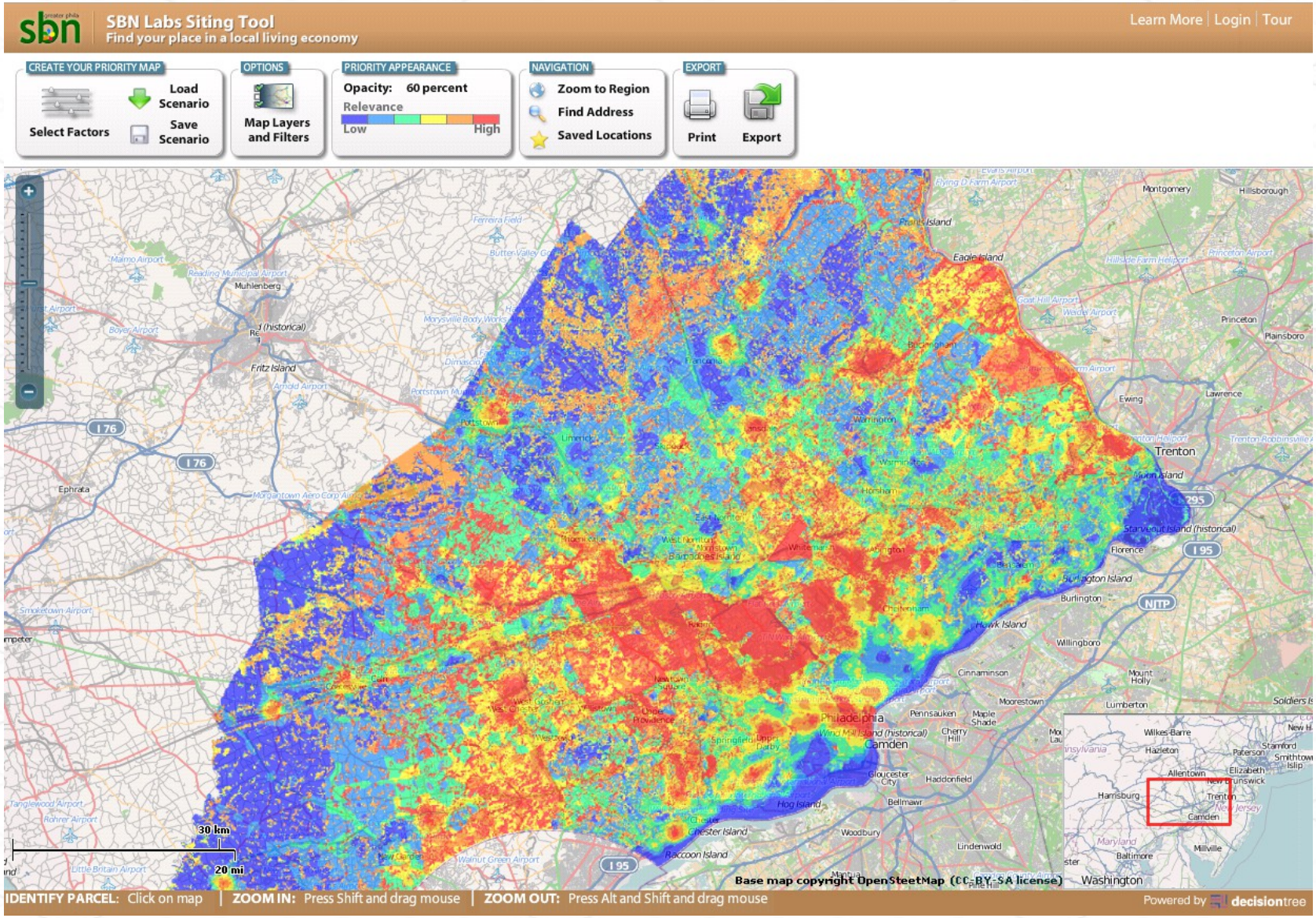
What is GeoTrellis?

- Geospatial data types\IO (raster & vector)
- Raster operations (Map Algebra)
- Utilities for creating web services

geospatial with scala

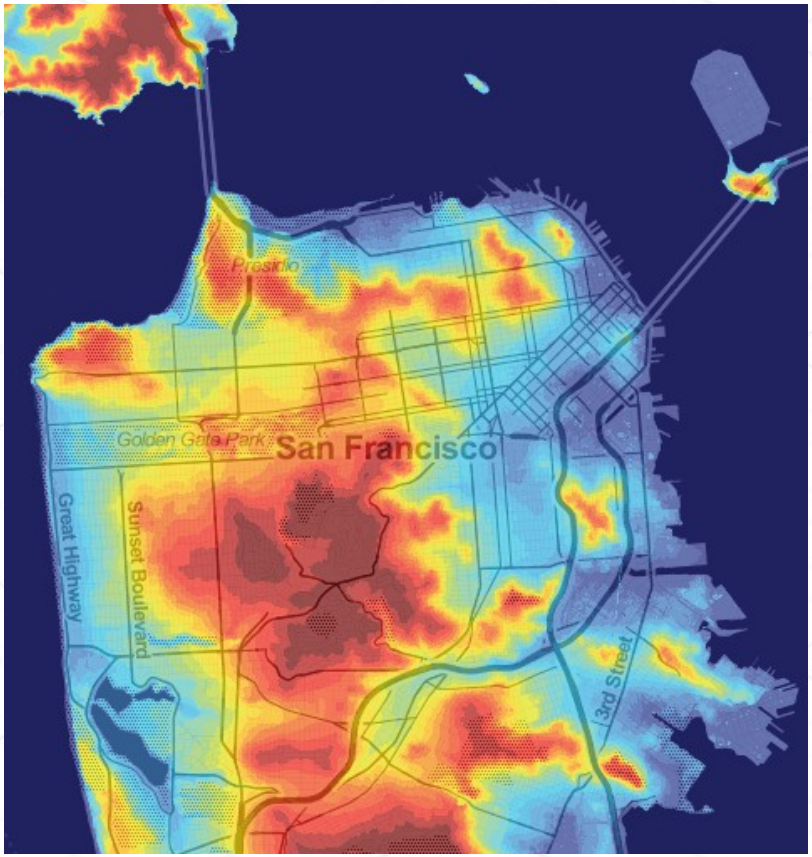
azavea
beyond dots on a map

GeoTrellis



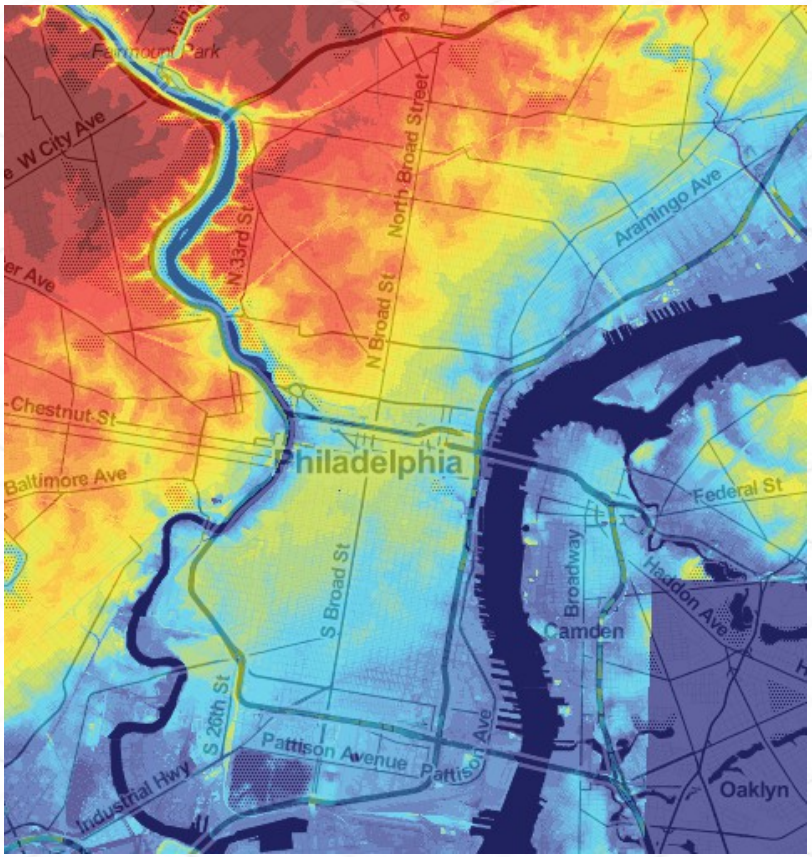
Business Siting

GeoTrellis



6183 x 4992

118 MB



4598 x 4867

86 MB

GeoTrellis

1770271 x 910139

5.8 TB

DigitalGlobe™

GeoTrellis + akka

GeoTrellis v0.9:

- Parallel operations across tiles
- Parallel execution of operations
- Basic cluster capabilities with Akka clustering

GeoTrellis + akka

What's missing?

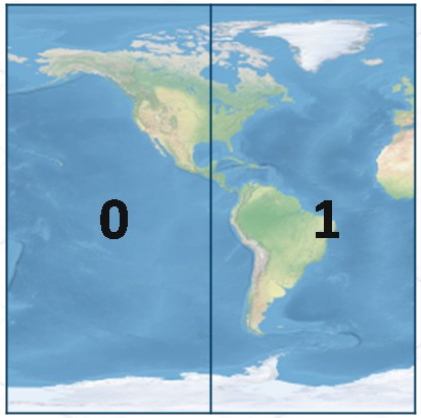
- Sharding raster data across the cluster
- Caching operation results across cluster
- HDFS support
- Advanced Fault tolerance
- Advanced Scheduling
- ...

Spark
Lightning-Fast Cluster Computing

+

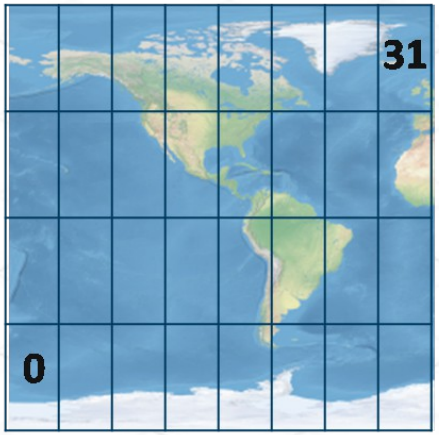
GeoTrellis

Level 1

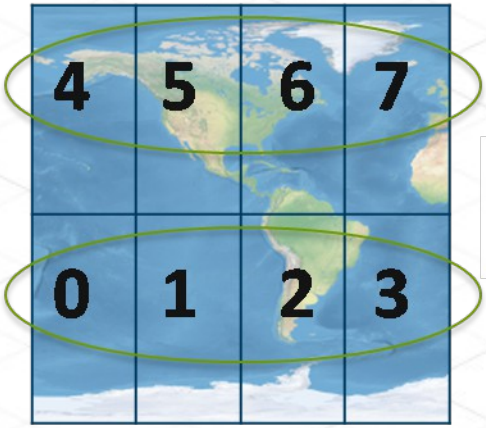


Level 2

Level 3



Level 2



Partitioned across two map files, one for each row

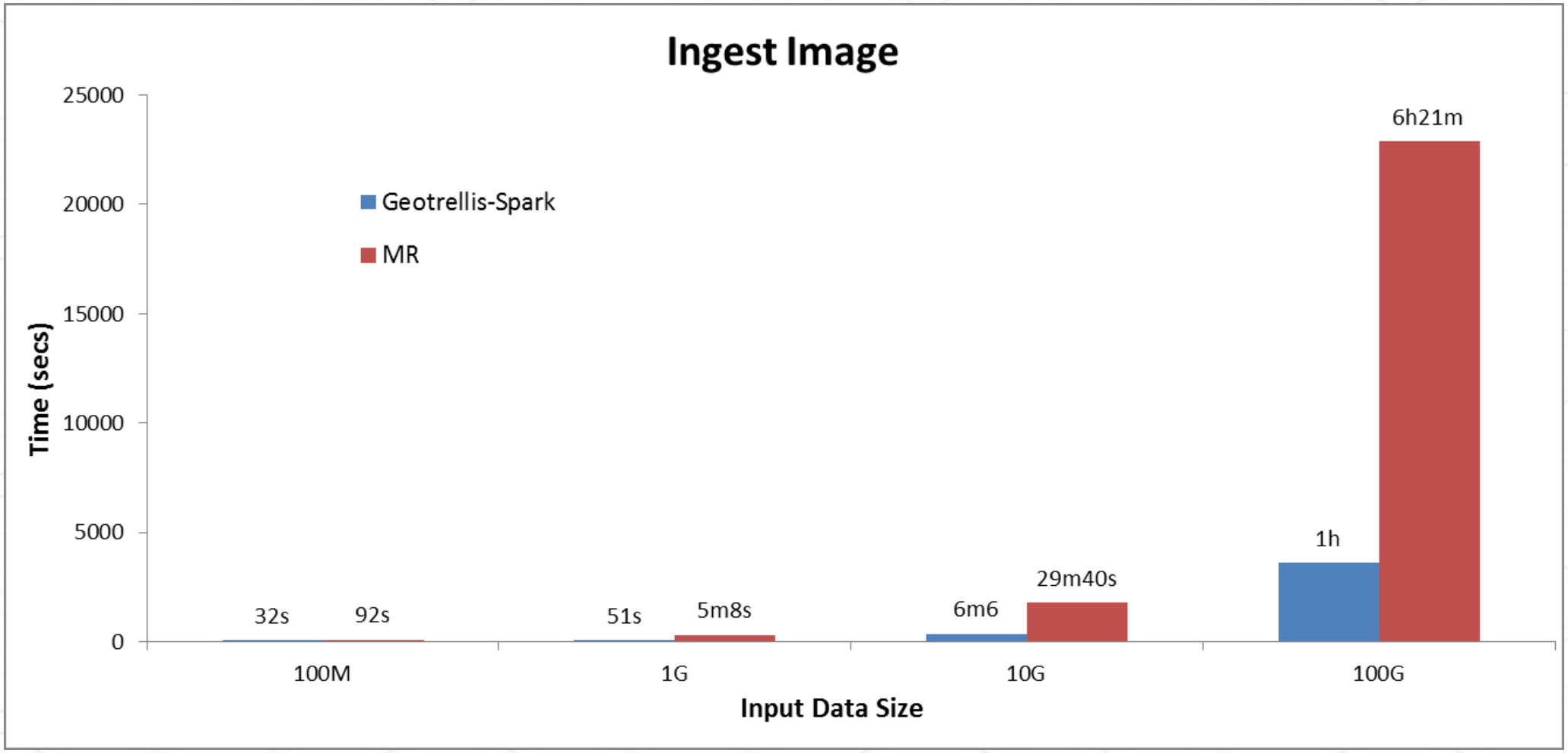
Key	Value
0	Tile 0
1	Tile 1
2	Tile 2
3	Tile 3

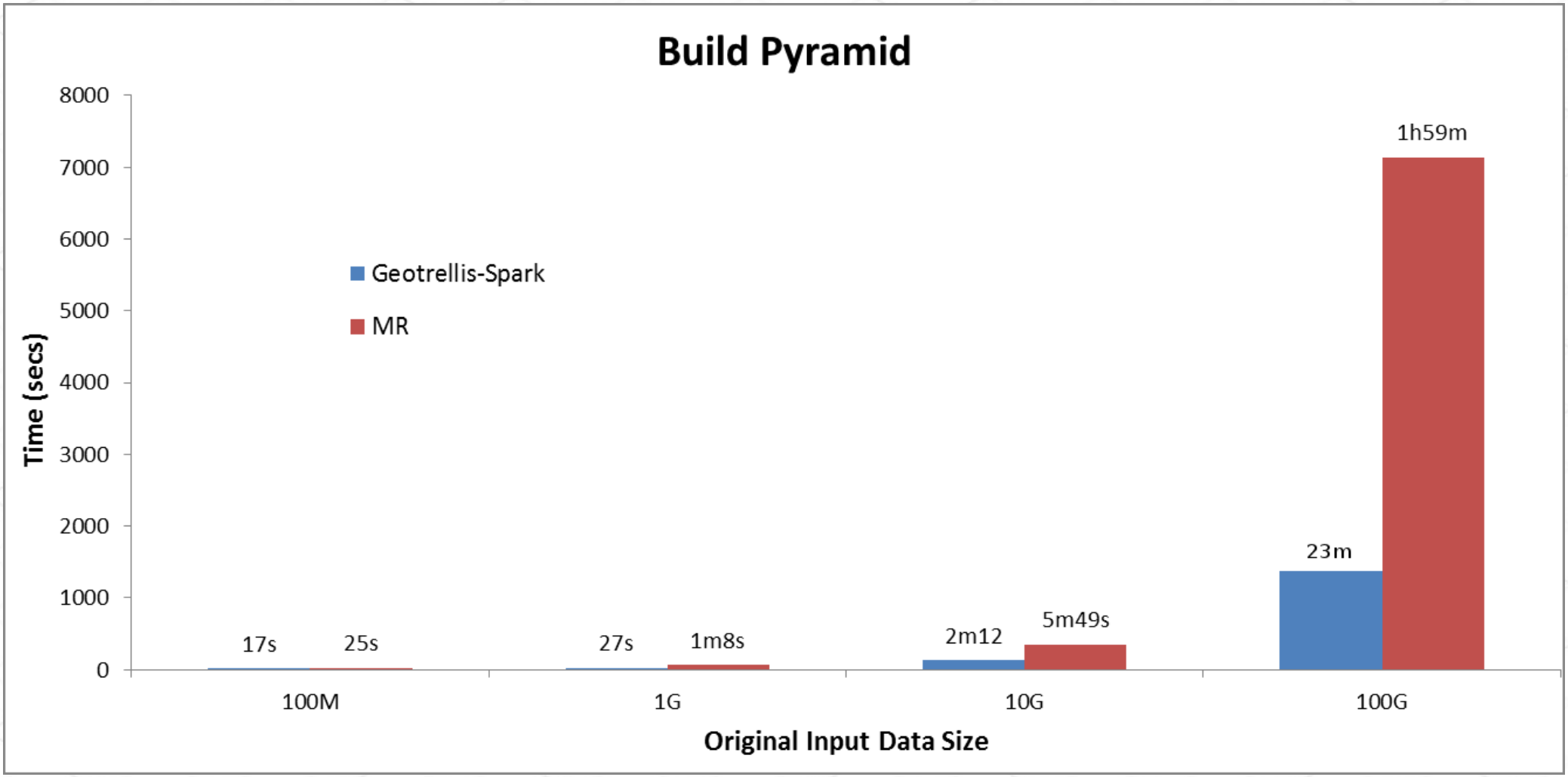
Key	Value
4	Tile 4
5	Tile 5
6	Tile 6
7	Tile 7

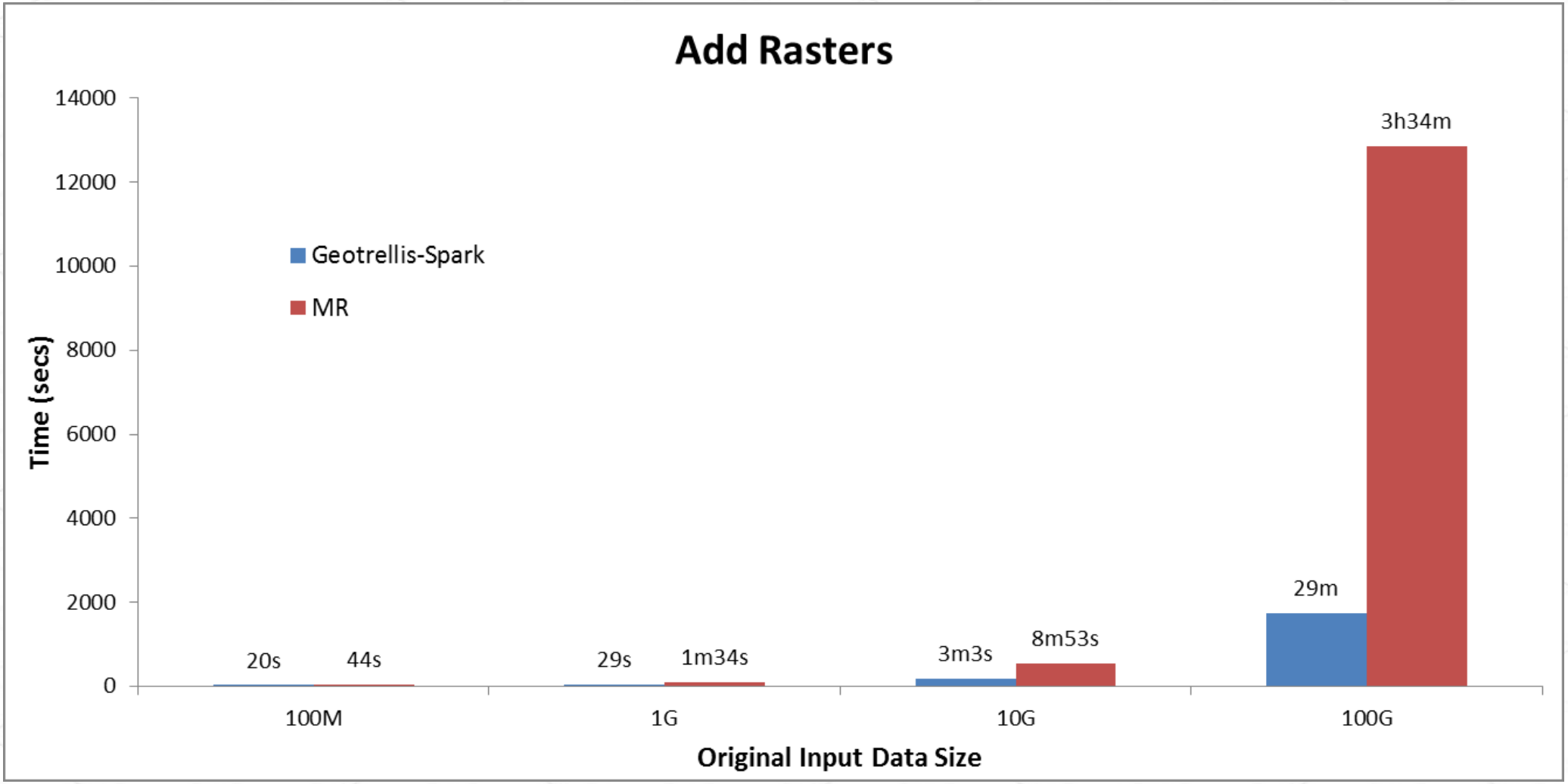
Data Types
Key: Long
Value: Raster

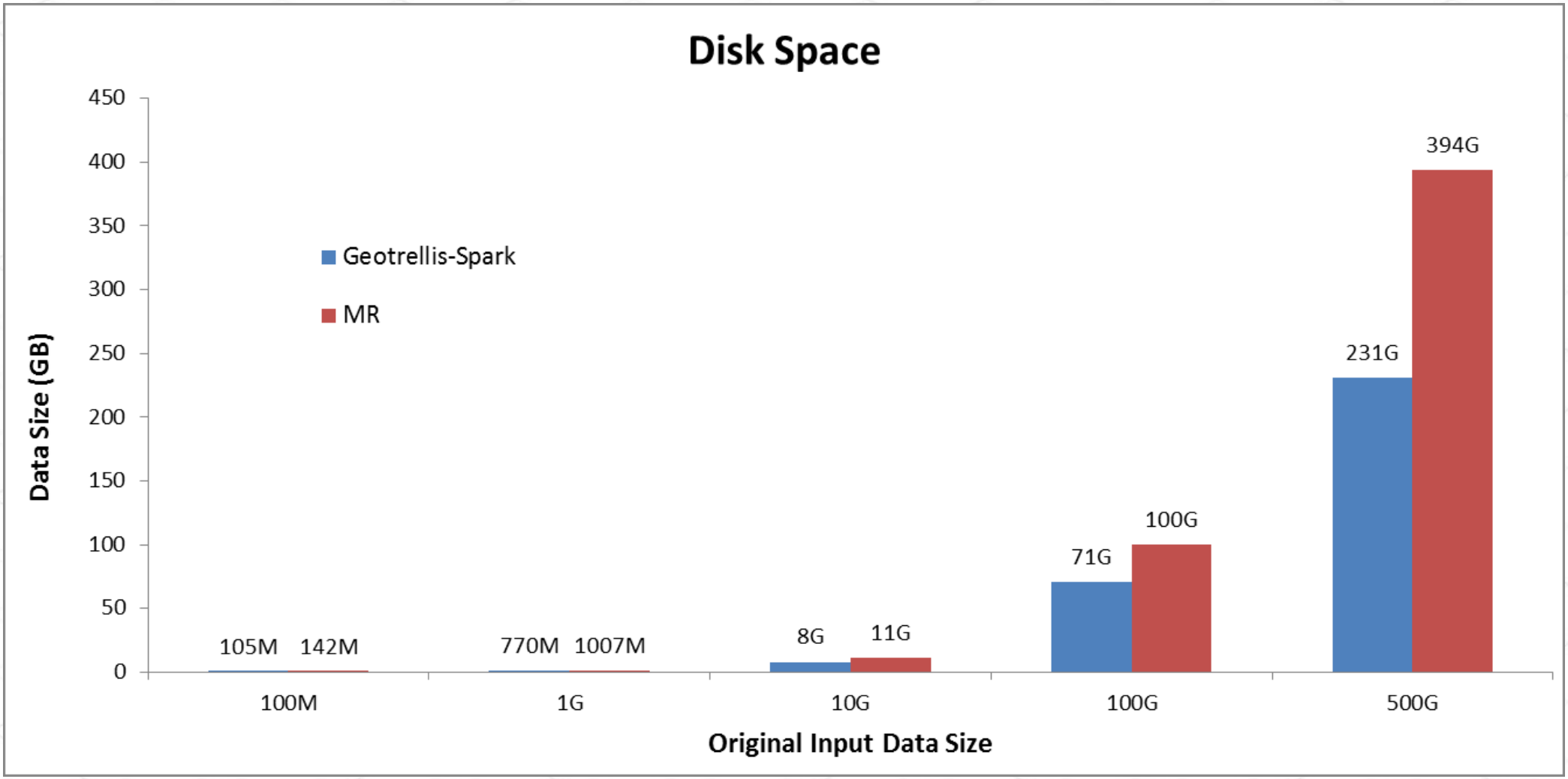
RasterRDD

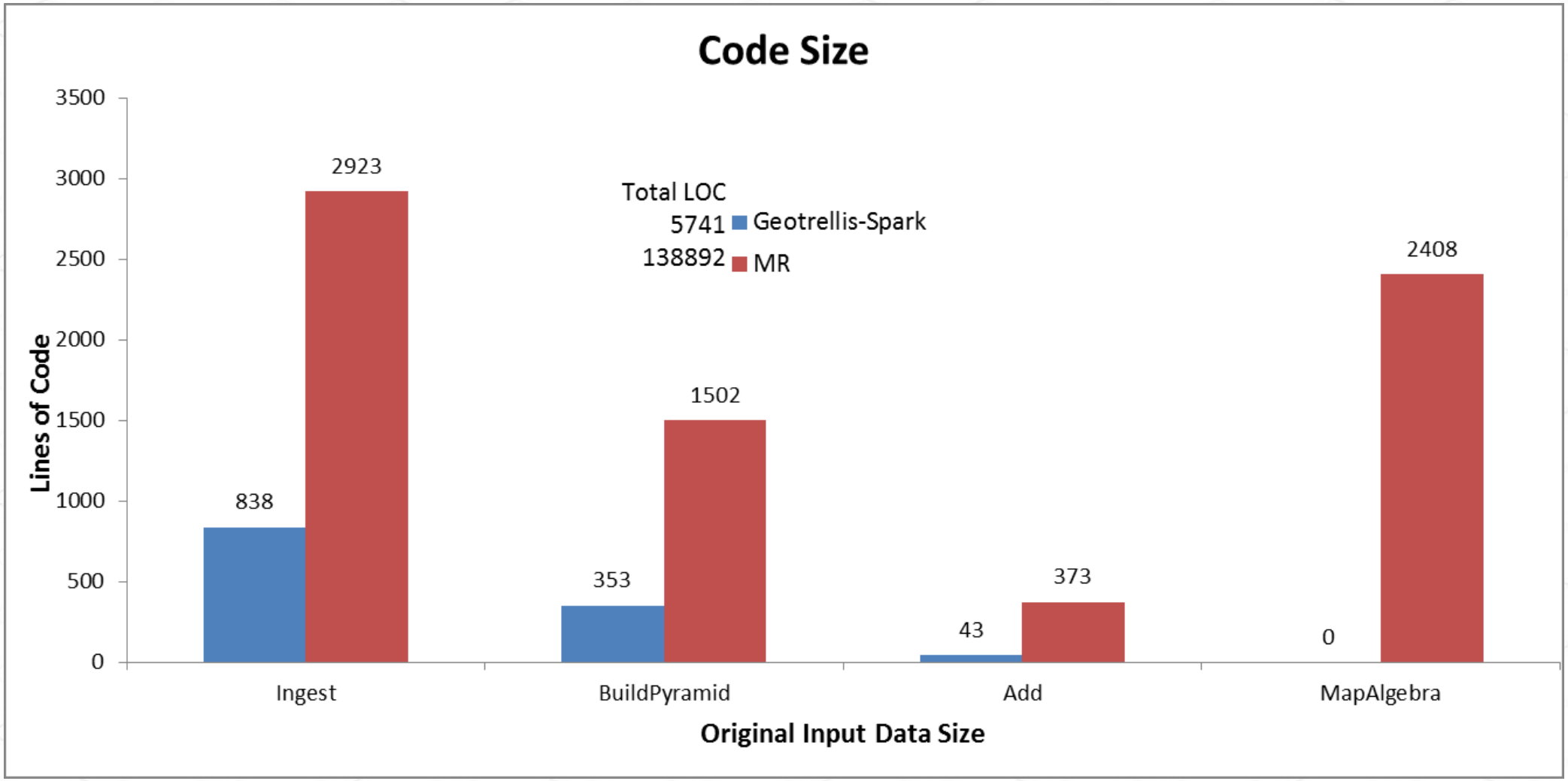
- Custom Partitioner based on TileID
  so that each worker works with one
  MapFile in HDFS.
- Defines raster operations in terms of Spark
  transformation and actions like mapPartitions,
  zipPartition, etc.
- Operations use logic already written and tested in
  core GeoTrellis, but use Spark for distributing those
  operations over tiles.


Ingest Image
25000
20000
15000
10000
5000
0
Time (secs)
Geotrellis-Spark
MR
32s
92s
51s
5m8s
6m6
29m40s
1h
6h21m
100M
1G
10G
100G
Input Data Size


Build Pyramid
8000
7000
6000
5000
4000
3000
2000
1000
0
Time (secs)
Geotrellis-Spark
MR
17s
25s
27s
1m8s
2m12
5m49s
23m
1h59m
100M
1G
10G
100G
Original Input Data Size


Add Rasters
14000
12000
10000
8000
6000
4000
2000
0
Time (secs)
Geotrellis-Spark
MR
20s
44s
29s
1m34s
3m3s
8m53s
29m
3h34m
100M
1G
10G
100G
Original Input Data Size


Disk Space
Data Size (GB)
450
400
350
300
250
200
150
100
50
0
Geotrellis-Spark
MR
105M
142M
770M
1007M
8G
11G
71G
100G
231G
394G
100M
1G
10G
100G
500G
Original Input Data Size


Code Size
Lines of Code
3500
3000
2500
2000
1500
1000
500
0
Total LOC
5741 Geotrellis-Spark
138892 MR
2923
838
1502
353
373
43
2408
0
Ingest
BuildPyramid
Add
MapAlgebra
Original Input Data Size

What's next?

- More operations: Focal operations, zonal operations
- Creating tile serving mechanisms for web services.
- Vector operations on spark?

+

=

geospatial at scale

GeoTrellis

geotrellis.io
github.com/geotrellis/geotrellis

IRC: #geotrellis on freenode