Abstract

Apache Spark is used on NOAA weather data from 2008-2012. Highest temperature is recorded. Performance analysis indicates speedup of about 3x on multiple nodes.

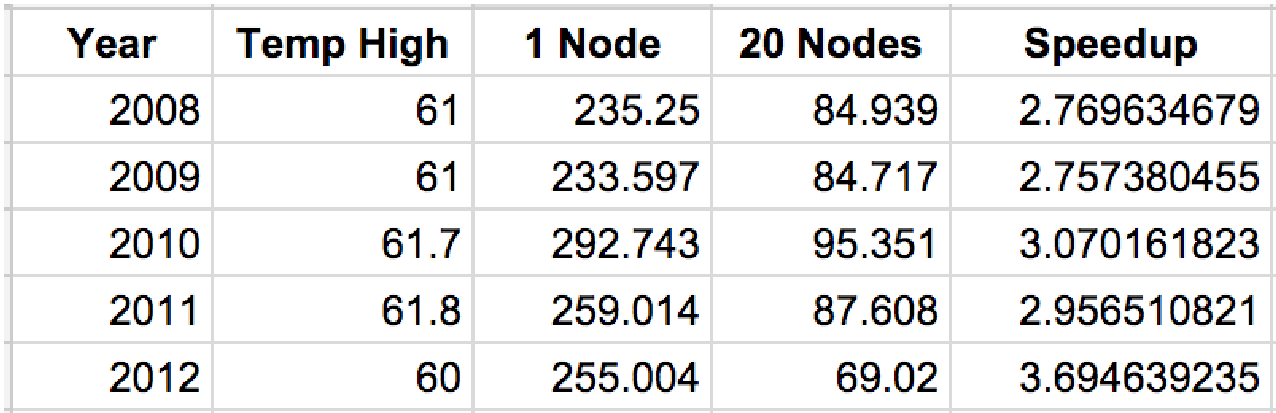
Implementation

PySpark is used to map the data by isolating the air temperature recording. Then reduce is used to compare values to find the highest.

Testing Methodology

Testing is run on a single EOS machine as a baseline. By default, the Spark output provides timing of program execution; these numbers are recorded and graphed. Tests were run on a single node and on a twenty node cluster.

Discussion

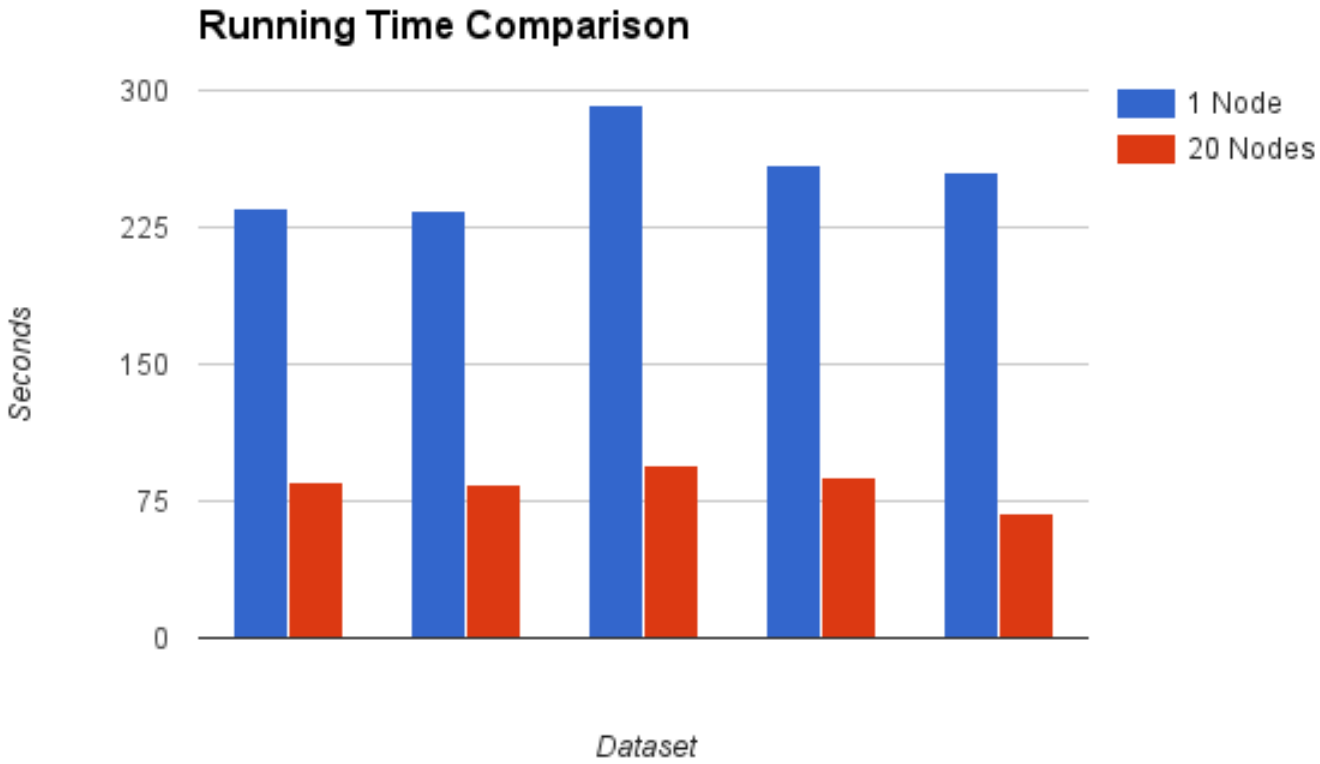


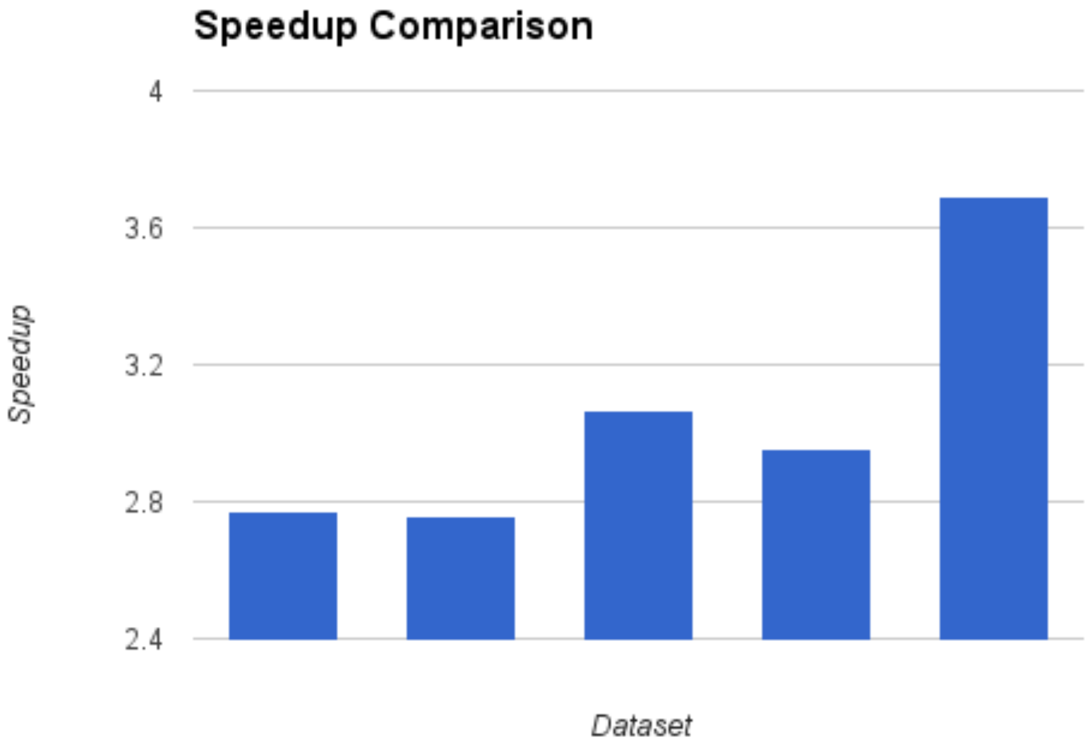
Year	Temp High	1 Node	20 Nodes	Speedup
2008	61	235.25	84.939	2.769634679
2009	61	233.597	84.717	2.757380455
2010	61.7	292.743	95.351	3.070161823
2011	61.8	259.014	87.608	2.956510821
2012	60	255.004	69.02	3.694639235

Significant speedup is observed when running on multiple nodes. Apache Spark’s ability to distribute the workload is very effective across large data sets.

Conclusion

Spark is an effective platform for both data analysis on large data sets - with the ability to scale calculation up by adding machines to a cluster.


Running Time Comparison
300
225
150
75
0
Seconds
Dataset
1 Node
20 Nodes


Speedup Comparison
4
3.6
3.2
2.8
2.4
Speedup
Dataset

```python
# Brett VanderHaar
from __future__ import print_function

import sys
import math
from operator import add
from pyspark import SparkContext

def mapper(line):
        # positive or negative
        sign = line[87:88]
        # before the decimal point, remove leading zeros
        before_decimal = line[88:91].lstrip("0")
        # combine into string that can be cast to decimal
        degrees = sign + before_decimal + "." + line[91]
        if (float(degrees) < 800):
            return float(degrees)
        else:
            return 0


def reducer(a, b):
    if a > b:
        return a
    else:
        return b

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: program <file>", file=sys.stderr)
        exit(-1)
    sc = SparkContext(appName="PySparkTemperature")
    lines = sc.textFile(sys.argv[1], 1)
    output = lines.map(mapper) \
                  .reduce(reducer)

    print ('Max = %.1f' % output)

    sc.stop()
```