**Reason for using MAPREDUCE**

I have done HW4 using MapReduce, so I continued with it only. The output received is as below:

| **YEAR** | Avg TMAX | Avg TMIN | Min  TMIN | Max TMAX |
|---|---|---|---|---|
| 2000 | 17.558 | 4.430 | 57.8 | 52.2 |
| 2001 | 17.874 | 4.792 | -52.8 | 52.8 |
| 2002 | 17.722 | 4.654 | -47.2 | 53.3 |
| 2003 | 17.709 | 4.862 | -50.0 | 53.3 |
| 2004 | 17.450 | 4.909 | -53.3 | 51.7 |
| 2005 | 17.766 | 4.980 | -55.0 | 53.9 |
| 2006 | 18.077 | 5.085 | -52.8 | 52.8 |
| 2007 | 17.821 | 4.907 | -53.9 | 53.9 |
| 2008 | 17.009 | 4.073 | -57.8 | 52.8 |
| 2009 | 16.869 | 4.336 | -55.6 | 53.3 |
| 2010 | 17.151 | 4.726 | -53.3 | 51.7 |
| 2011 | 17.246 | 4.605 | -51.7 | 51.1 |
| 2012 | 18.39 | 5.348 | -54.4 | 53.7 |
| 2013 | 16.731 | 4.300 | -52.8 | 53.9 |
| 2014 | 16.843 | 4.384 | -50.0 | 52.2 |
| 2015 | 17.723 | 5.353 | -52.8 | 55.6 |
| 2016 | 17.907 | 5.510 | -46.9 | 53.9 |
| 2017 | 17.664 | 5.281 | -52.0 | 52.8 |
| 2018 | 17.132 | 4.945 | -47.8 | 52.8 |
| 2019 | 7.303 | -3.655 | -49.4 | 41.7 |

**5 COLDEST AND HOTTEST WEATHER STATIONS PER YEAR**

|  | **YEAR** |  |
|---|---|---|
| **HOTTEST STATIONS** | **2000** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** |  | ['USC00501684', 'USC00505644', 'USC00505644', 'USC00501684', 'USC00508140'] |

| | | |
|---|---|---|
| **HOTTEST STATIONS** | **2001** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USW00026508', 'USR0000ABCA', 'USW00026508', 'USS0051R01S', 'USW00026508'] |
| **HOTTEST STATIONS** | **2002** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USR0000AKAI', 'USS0051R01S', 'USS0050S01S', 'USR0000ABEV', 'USC00503212'] |
| **HOTTEST STATIONS** | **2003** | ['USC00042319', 'USR0000CMEA', 'USC00042319', 'USC00042319', 'USR0000AHAV'] |
| **COLDEST STATIONS** | | ['USC00501492', 'USC00501492', 'USS0051R01S', 'USW00026533', 'USS0050S01S'] |
| **HOTTEST STATIONS** | **2004** | ['USC00042319', 'USC00042319', 'USC00024761', 'USC00024761', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00501684', 'USC00501684', 'USC00502568', 'USS0045O10S', 'USS0045O10S'] |
| **HOTTEST STATIONS** | **2005** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00501684', 'USC00509313', 'USC00501684', 'USC00501684', 'USC00501684'] |
| **HOTTEST STATIONS** | **2006** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USR0000ASEL', 'USC00501492', 'USC00501492', 'USC00501492', 'USC00501492'] |
| **HOTTEST STATIONS** | **2007** | ['USC00042319', 'USC00042319', 'USW00053139', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00501684', 'USC00501684', 'USC00501684', 'USS0045R01S', 'USC00501684'] |
| **HOTTEST STATIONS** | **2008** | ['USC00044297', 'USC00042319', 'USC00042319', 'USC00024761', 'USC00044297'] |
| **COLDEST STATIONS** | | ['USC00501684', 'USC00501684', 'USC00501684', 'USC00501684', 'USC00501684'] |
| **HOTTEST STATIONS** | **2009** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00501684', 'USC00502101', 'USC00502101', 'USC00501684', 'USC00502101'] |
| **HOTTEST STATIONS** | **2010** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USR0000AHAV'] |
| **COLDEST STATIONS** | | ['USC00501684', 'USC00502101', 'USC00501684', 'USC00502101', 'USS0051R01S'] |
| **HOTTEST STATIONS** | **2011** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00509869', 'USS0045R01S', 'USS0045R01S', 'USS0051R01S', 'USS0051R01S'] |
| **HOTTEST STATIONS** | **2012** | ['USS0005N23S', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00503165', 'USC00503165', 'USC00503165', 'USC00503212', 'USS0051R01S'] |

| | | |
|---|---|---|
| **HOTTEST STATIONS** | **2013** | ['USC00042319', 'USW00004134', 'USC00042319', 'USC00042319', 'USC00044297'] |
| **COLDEST STATIONS** | | ['USC00502339', 'USC00501684', 'USC00501684', 'USC00501684', 'USC00502339'] |
| **HOTTEST STATIONS** | **2014** | ['USC00042319', 'USC00042319', 'USW00053139', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00501684', 'USC00501684', 'USC00501684', 'USC00501684', 'USC00501684'] |
| **HOTTEST STATIONS** | **2015** | ['USR0000HKAU', 'USR0000HKAU', 'USR0000HKAU', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00502339', 'USC00502339', 'USC00501684', 'USC00501684', 'USC00502339'] |
| **HOTTEST STATIONS** | **2016** | ['USR0000CBEV', 'USC00042319', 'USC00042319', 'USC00040924', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USS0051R01S', 'USR0000ACHL', 'USC00501684', 'USR0000ACHL', 'USR0000ACHL'] |
| **HOTTEST STATIONS** | **2017** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00021050'] |
| **COLDEST STATIONS** | | ['USS0051R01S', 'USR0000ASLC', 'USW00026529', 'USW00026529', 'USS0051R01S'] |
| **HOTTEST STATIONS** | **2018** | ['USC00042319', 'USC00042319', 'USC00042319', 'USC00042319', 'USC00042319'] |
| **COLDEST STATIONS** | | ['USC00501684', 'USC00501684', 'USR0000ANOR', 'USR0000AKAV', 'USW00096406'] |
| **HOTTEST STATIONS** | **2019** | ['USW00022010', 'USC00415048', 'USW00012907', 'USC00417624', 'USR0000TFAL'] |
| **COLDEST STATIONS** | | ['USC00509891', 'USC00501684', 'USC00211840', 'USC00218618', 'USC00211840'] |

**HOTTEST AND COLDEST DAYS WITH STATIONS**

| **TEMPERATURE** | **STATION CODE** |
|---|---|
| 55.6 | USR0000HKAU |
| -57.8 | USC00501684 |

**mapper.py**

```
#!/usr/bin/env python
import sys

def mapToDict(ls):
    return {'Id':ls[0],
            'Date':ls[1],
            'Type':ls[2],
            'Value':ls[3],
            'MFlag':ls[4],
            'QFlag':ls[5],
            'SFlag':ls[6],
            'OBSTime':ls[7]}

for line in sys.stdin:
    parse = line.strip().upper().split(',')
    row = mapToDict(parse)
    if 'TMAX' != row['TYPE'] and 'TMIN' != row['TYPE']:
        continue
    if row['Value'] == -9999:
        continue
    if row['SFlag'] == '':
        continue
    if row['QFlag'] != '':
        continue
    if row['MFlag'] == 'P':
        contiune
    print '%s,%s,%s,%s,' % (row['Date'],row['Id'],row['Type'],row['Value'])
```

**reducer.py**

```
#!/usr/bin/env python
import sys
import operator

max_cnt = 0
min_cnt = 0

avg_max = 0
avg_min = 0

hottest = []
coldest = []

curr_year = None

max = (-9999,'','')
min = (99999,'','')

for line in sys.stdin:
   line = line.strip().split(',')

   date = line[0]
   year = date[:4]
   id = line[1]
   metric = line[2]
   value = line[3]

   try:
```

```
   value = int(value)
except ValueError:
   continue

if curr_year is None:
   curr_year = year

if curr_year != year:
   print 'Year: %s' % curr_year
   print 'Average TMAX: %s' % (avg_max * 1.0 / max_cnt)
   print 'Average TMIN: %s' % (avg_min * 1.0 / min_cnt)

   print 'Hottest Day: day: %s | val: %s | loc: %s' %(hottest[0][2], hottest[0][0], hottest[0][1])
   print 'Coldest Day: day: %s | val: %s | loc: %s' %(coldest[0][2], coldest[0][0], coldest[0][1])

   print 'Hottest Stations %s' % ([x[1] for x in hottest])
   print 'Hottest Station Values %s' % ([x[0] for x in hottest])
   print 'Coldest Stations %s' % ([x[1] for x in coldest])
   print 'Coldest Station Values %s' % ([x[0] for x in coldest])

   print '-----------------------------------------------------------'


   curr_year = year

   max_cnt = 0
   min_cnt = 0

   avg_max = 0
```

```
    avg_min = 0

    hottest = []
    coldest = []

if metric == 'TMAX':
    avg_max += value
    max_cnt += 1
    if max[0] < value:
        max = (value,id,date)

    hottest.append((value,id,date))

    if len(hottest) > 5:
        hottest = sorted(hottest, key=operator.itemgetter(0), reverse=True)
        hottest.pop(len(hottest) - 1)

elif metric == 'TMIN':
    avg_min += value
    min_cnt += 1
    if min[0] > value:
        min = (value,id,date)

    coldest.append((value,id,date))

    if len(coldest) > 5:
        coldest = sorted(coldest, key=operator.itemgetter(0))
        coldest.pop(len(coldest) - 1)
```

```
print 'Year: %s' % curr_year
print 'Average TMAX: %s' % (avg_max * 1.0 / max_cnt)
print 'Average TMIN: %s' % (avg_min * 1.0 / min_cnt)

print 'Hottest Day: day: %s | val: %s | loc: %s' %(hottest[0][2], hottest[0][0], hottest[0][1])
print 'Coldest Day: day: %s | val: %s | loc: %s' %(coldest[0][2], coldest[0][0], coldest[0][1])

print 'Hottest Stations %s' % ([x[1] for x in hottest])
print 'Hottest Station Values %s' % ([x[0] for x in hottest])
print 'Coldest Stations %s' % ([x[1] for x in coldest])
print 'Coldest Station Values %s' % ([x[0] for x in coldest])

print '============================================================================'

print 'Max TMAX: %s | Station: %s' % (max[0],max[1])
print 'Min TMIN: %s | Station: %s' % (min[0],min[1])
```