Noter til "Concurrency - Aggregation, MapReduce"

1. Introduktion til Aggregation

- Aggregation er handlingen at samle elementer for at danne en samlet mængde.
- I sammenhæng med parallel programmering, samler aggregation delresultater til et samlet resultat.

2. Eksempel: Beregning af Pi med Dartboard Algoritmen

- Sequential beregning:

```
private static double SerialEstimationOfPi()
{
    double nInside = 0;
    double stepSize = 1 / (double)_nDarts; // nDarts - division in x and y
    for (int i = 0; i < _nDarts; i++)
    {
        var x = i * stepSize;
        for (int j = 0; j < _nDarts; j++)
        {
            var y = j * stepSize;
            if (Math.Sqrt(x * x + y * y) < 1.0) ++nInside;
        }
    }
    return 4 * nInside / (_nDarts * _nDarts);
}
```

3. Parallel Beregning af Pi

- Første forsøg:

```
private static double ParallelEstimationOfPi()
{
    var locker = new object();
    double nInside = 0;
    double stepSize = 1 / (double)_nDarts;
    Parallel.For(0, _nDarts, i =>
    {
        var x = i * stepSize;
        for (int j = 0; j < _nDarts; j++)
        {
            var y = j * stepSize;
            if (Math.Sqrt(x * x + y * y) < 1.0)
                lock (locker) ++nInside;
        }
    });
    return 4 * nInside / (_nDarts * _nDarts);
}
```

- Andet forsøg med lokal initialisering:

```
private static double ParallelEstimationOfPi()
{
    var locker = new object();
    double nInsideCircle = 0;
    double stepSize = 1 / (double)_nDarts;
    Parallel.For(0, _nDarts,
        () => 0, // localInit
        (i, state, nInside) =>
        {
            var x = i * stepSize;
            for (int j = 0; j < _nDarts; j++)
            {
                var y = j * stepSize;
                if (Math.Sqrt(x * x + y * y) < 1.0) ++nInside;
            }
            return nInside; // LocalFinally
        },
        inside => { lock (locker) nInsideCircle += inside; });
    return 4 * nInsideCircle / (_nDarts * _nDarts);
}
```

4. MapReduce

- MapReduce er et mønster, der tillader parallel beregning på store datasæt.
- Består af fire trin: Distribute, Map, Group, Reduce.

5. Eksempel på MapReduce Implementering i C#:

- Ordoptælling efter længde:

```
public static void Main(string[] args)
{
    var files = Directory.EnumerateFiles(@"C:\(…)\Books",
"*.txt").AsParallel();
    var wordCounts = files.MapReduce(
        path => Map(path),
        word => word.Length,
        group => Reduce(group));
    foreach (var pair in wordCounts)
    {
        Console.WriteLine($"{pair.Key}: {pair.Value}");
    }
}

public static IEnumerable<string> Map(string path)
{
    return File.ReadLines(path)
        .SelectMany(line => line.ToLower().Split(new char[] { ' ', ',',
'.', '-', '!', '?', ';' }));
}

public static IEnumerable<KeyValuePair<int, int>> Reduce(IGrouping<int,
string> group)
{
    yield return new KeyValuePair<int, int>(group.Key, group.Count());
}
```