Title: Implement Min, Max, Sum and Average operations using Parallel Reduction.

**Outcome:** At the end of this session students will be able to:

1) Understand REDUCE proress.

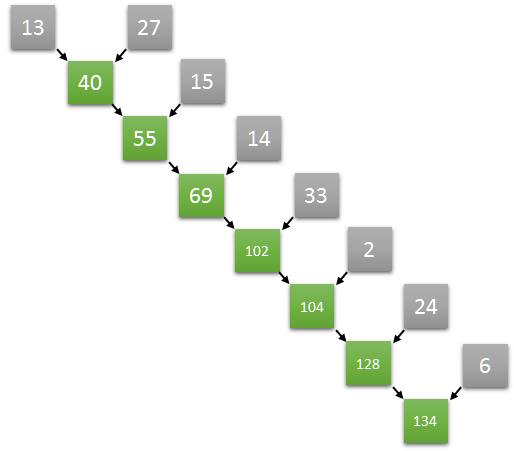
2) Understand parallel computing to find Min, Max and Average.

3) Able to write the code & test it for result.

**Reduce Operation**

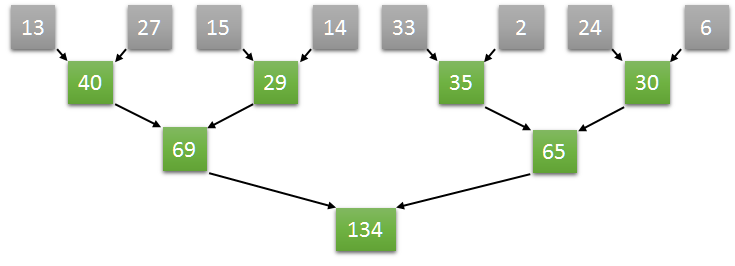
**Normal Sum operation**

The sum of an array whose values are 13, 27, 15, 14, 33, 2, 24, and 6 is 134doing something like this (((((((13+27)+15)+14)+33)+2)+24)+6).

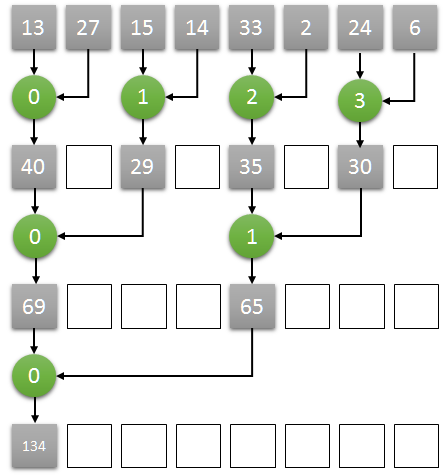
****

**Parallel Sum operation**

Adding values is an associative operation. So, we can try something like this ((13+27)+(15+14))+((33+2)+(24+6))



Let’s figure out how to do it using CUDA.



Here is the main idea:

* Assuming N as the number of the elements in an array, we start N/2 threads, one thread for every two elements
* Each thread computes the sum of the corresponding two elements, storing the result at the position of the first one.
* Iteratively, each step:
  + the number of threads halved (for example, starting with 4, then 2, then 1)
  + doubles the step size between the corresponding two elements (starting with 1, then 2, then 4)
* after some iterations, the reduction result will be stored in the first element of the array.

Procedure:

1) Write a program using text editor, name the source code with .cu or .c extension.

2) Compile the program using nvcc/OMP compiler.

3) Execute the program.

4) Verify the result.

Conclusion:

Min, Max and Average operation can be parallelized using reduce operation.