

Class Work 5:

map implementation with dynamic scheduling of tasks

- Consider a map computation on an array A of size N . The computation is in-place (i.e. $A[i]=F(A[i])$ for all i in $0..N-1$).
- You know that the time difference for computing different elements of the array A may be large, so a simple static partitioning of all elements of the array to a pool of workers would be inefficient. A dynamic scheduling policy has to be used instead. Furthermore, in order to preserve (as much as possible) space locality, the elements have to be assigned to workers in chunk of a given size.
- Write the implementation of the dynamic map described above using the FastFlow *task-farm* with a *feedback channel* between workers and the emitter in order to implement a request-reply protocol. Split the array elements in chunk of a given size (that is passed as a command line argument) and then assign the chunks dynamically to farm workers.

Example: `./dynmap #workers chunksize arraysizes`

Class Work 5: example

Suppose that $\text{chunksize}=2$ and num worker is 3, then the initial tasks table used for the scheduling is:

wid	#tasks	min-max
0	2	0-3
1	2	4-7
2	2	8-11

