Introduction to FastFlow programming

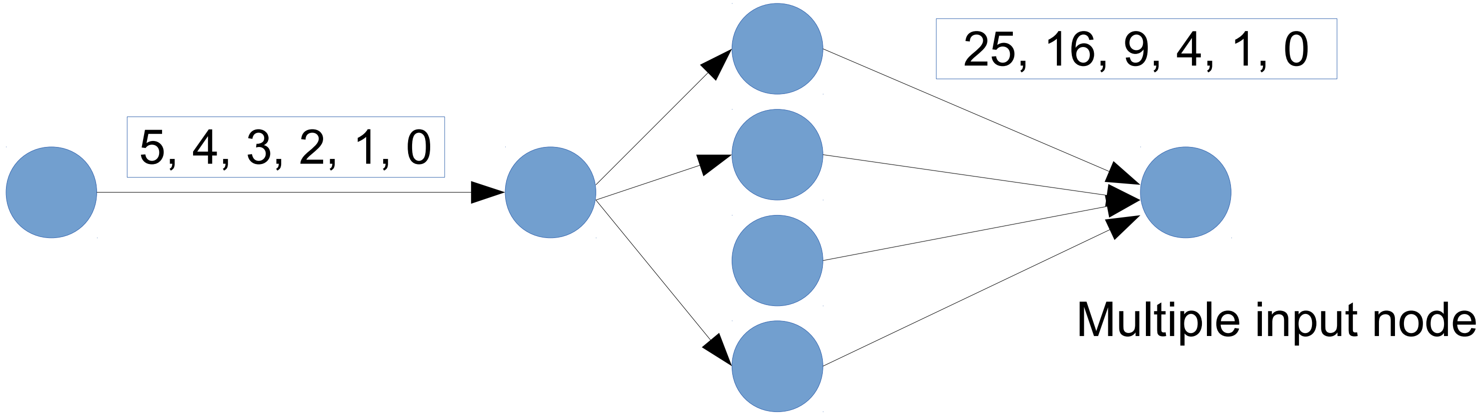
SPM lecture, November 2015

Massimo Torquati <torquati@di.unipi.it>

Computer Science Department, University of Pisa - Italy

ClassWork2: comments

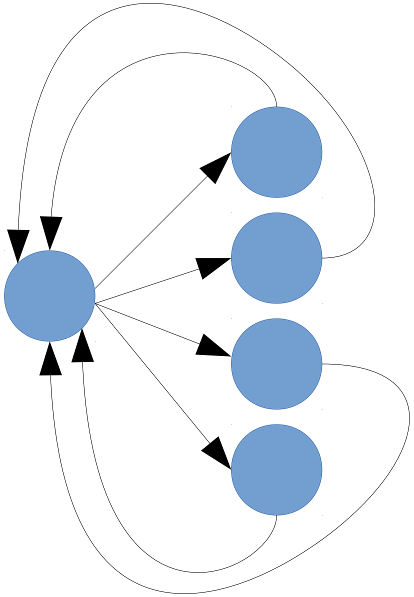
- 3-stage pipeline: pipe(seq, farm, seq)
- The farm does not have the collector node
- The third stage of the pipeline is a multi-input node (ff_minode_t)



- The Collector can be removed using:
  - myFarm.remove_collector();
  - If the next stage after the farm is a sequential node, it must be defined as *ff_minode_t* (multi-input node)

More on the *ff_farm*

- Emitter and Collector may be redefined by providing suitable ff_node objects
- Default task scheduling is (*pseudo) round-robin*
- Auto-scheduling:
  - myFarm.set_scheduling_ondemand()
- Possibility to implement user's specific scheduling strategies (ff_send_out_to)
  - *ff_send_out_to.cpp* example in the tutorial tests
- Master-Worker computation:
  - farm without the collector node
  - Workers send the results back to the Emitter
  - *feedback.cpp* example in the tutorial tests



master-worker skeleton

Ordered farm *ff_ofarm*

- Provides a total ordering between input and output
  - use case example: video streaming
- Limitations:
  - The number of tasks produced in output by the workers must be exactly the same of the number of tasks received in input
  - It is not possible to define your own scheduling and gathering policies
- If you don't need a strict input/output ordering then it is generally better to implement your own policy by re-defining the Emitter and the Collector

ClassWork3: comments

- Let's have a look at the proposed solution of the ClassWork3 assignment. You can find it under the folder **~smp1501/public/ClassWork3** of the course machine.

ClassWork4: finding prime numbers

- Problem: to find prime numbers in a given range of values.
  - es. primes between 200 and 250 are: 211, 223, 227, 229, 233, 239 and 241
- Starting from the provided primes.cpp sequential code that finds all prime numbers in a given range, write a (toy) program that computes the primes using the FF master-worker pattern by generating all numbers in the Emitter node. The Workers check if the number is a prime and if yes sends it back to the Emitter otherwise it discards the number.
- Then write a second version using the same pattern, but, instead of sending each single number to the workers, assigns a sub-range to each worker (map-like computation).