# ZeroMQ based aggregation service

Grupo de Sistemas Distribuídos
Universidade do Minho

## 1 Objectives

Implement a request processing service in Java using ZeroMQ, aiming to perform an aggregation over a series of numbers. It delegates requests to several workers and makes use of pipeline processing. Use the REQ-REP and PUSH-PULL ZeroMQ socket types.

## 2 Tasks

Assume the existence of two computationally expensive functions $f$ and $g$, both from integers to integers, and a comutative and associative aggregation funtion $h$ from pairs of integers to integers, with 0 as neutral value. Given a stream of numbers $x_i$ arriving at a node $S$, we aim to know the current aggregated value of $h$ applied successively to the results of $g(f(x_i)$.

- Due to hardware characteristics, $g$ is to be made available at a server $G$ as a multi-threaded service using $W_g$ workers.
- $f$ should run using $W_f$ instances, possibly at different nodes, in a program $F$.
- $h$ should run in a single thread at node $S$, accumulating the current aggregation value.
- The current aggregation value should be replied to each client that provides a new $x_i$, but not necessarily including the supplied $x_i$. Consider the issue of ensuring that the aggregation accounts for $x_i$.

Write the programs for node $S$, nodes running $F$, node $G$, and possibly other auxiliar programs. Use concrete simple functions $f$, $g$, $h$ and number of instances $W_f$, $W_g$. Write also a program $C$ to be used as client of $S$, that supplies numbers. Test all of them in a single host, by running different programs configured using appropriate port numbers.