

ARQUITECTURA CLIENTE SERVIDOR

Sistema de computadores REQ-REP

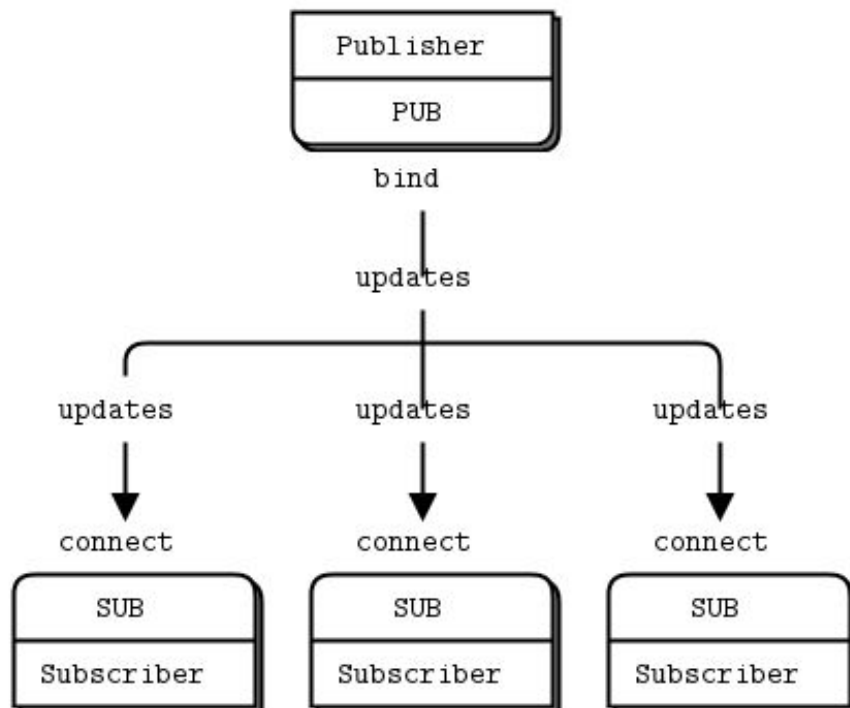
Patrón de comunicaciones síncrono

04/09/19

NUEVO PATRÓN (PUB-SUB)

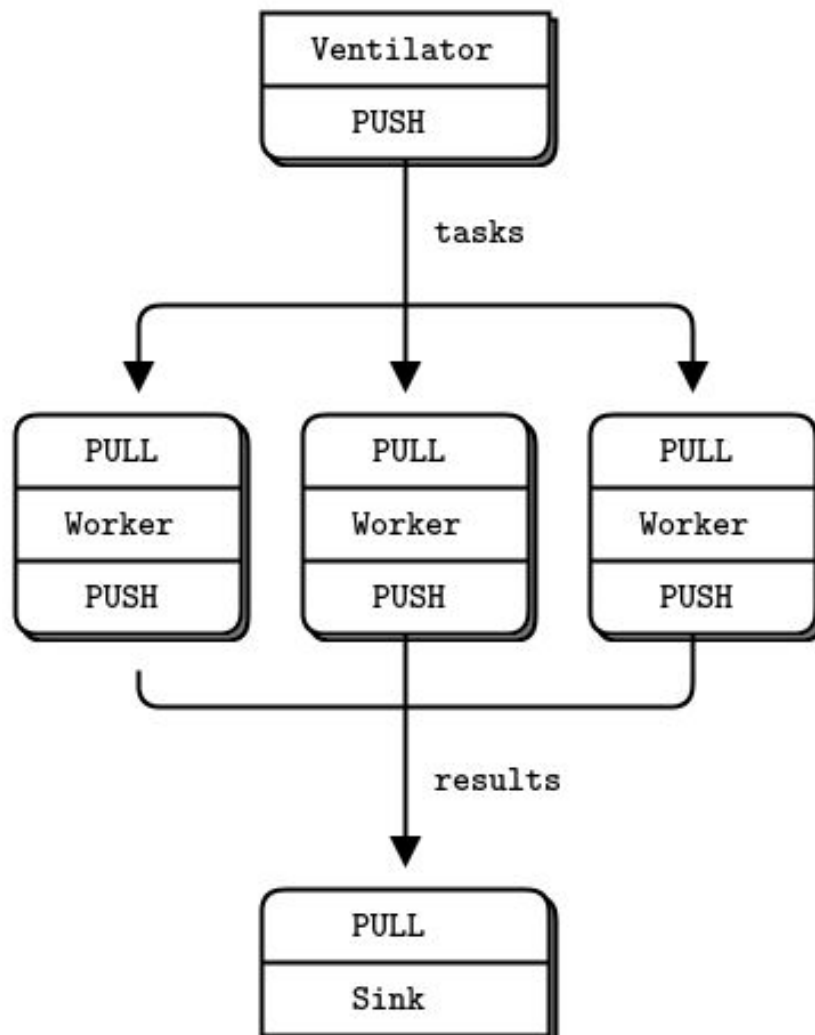
Resolver problemas como dar la hora de diferentes ciudades

Publish-Subscribe



Una forma más avanzada

- Parallel Pipeline



tasks = tareas

worker = nodos de cómputo llamados trabajadores

sink = resumidero, sifón donde todo se recoge

el sink toma todos los resultados y los va unir de acuerdo a la lógica del problema, ahora el sink le manda la solución al cliente

Arquitectura muy buena para resolver problemas en paralelo

worker puedo tener los que quiera entre más tenga mejor el paralelismo

ES UN PATRÓN DE PASO DE MENSAJES QUE NOS VA AYUDAR EN EL PARALELISMO

$T(P) = T(V) + T(S) + (n \cdot t) / W + 2nC$ -> Esta es la ecuación que nos dice cuánto tiempo
(SOLUCIÓN PARALELA DE NUESTRO PROBLEMA)

$2nC =$ > Si estoy trabajando en el mismo computador esta operación va a ser cero "0"

$T(P)$ = Tiempo que dura el problema

$T(V)$ = Tiempo que le toma al Ventilator enviar la solicitud (divirlo)

$T(S)$ = Tiempo que le toma a Sink reconstruir el mensaje (juntarlo)

t = tiempo que me toma resolver cada tarea (aproximadamente)

n = número de tareas

W = número de Trabajadores (Worker)

C = Es el tiempo que me demoro en transmitir una tarea o mensaje

Ecuación de manera secuencial

$T_s(P) = n \cdot t$

Debemos llegar a que:

$T_s(P) > T(P)$

TAREA

Próxima clase ejemplos de Parallel task worker in Python En ZQM

vert

work

skin