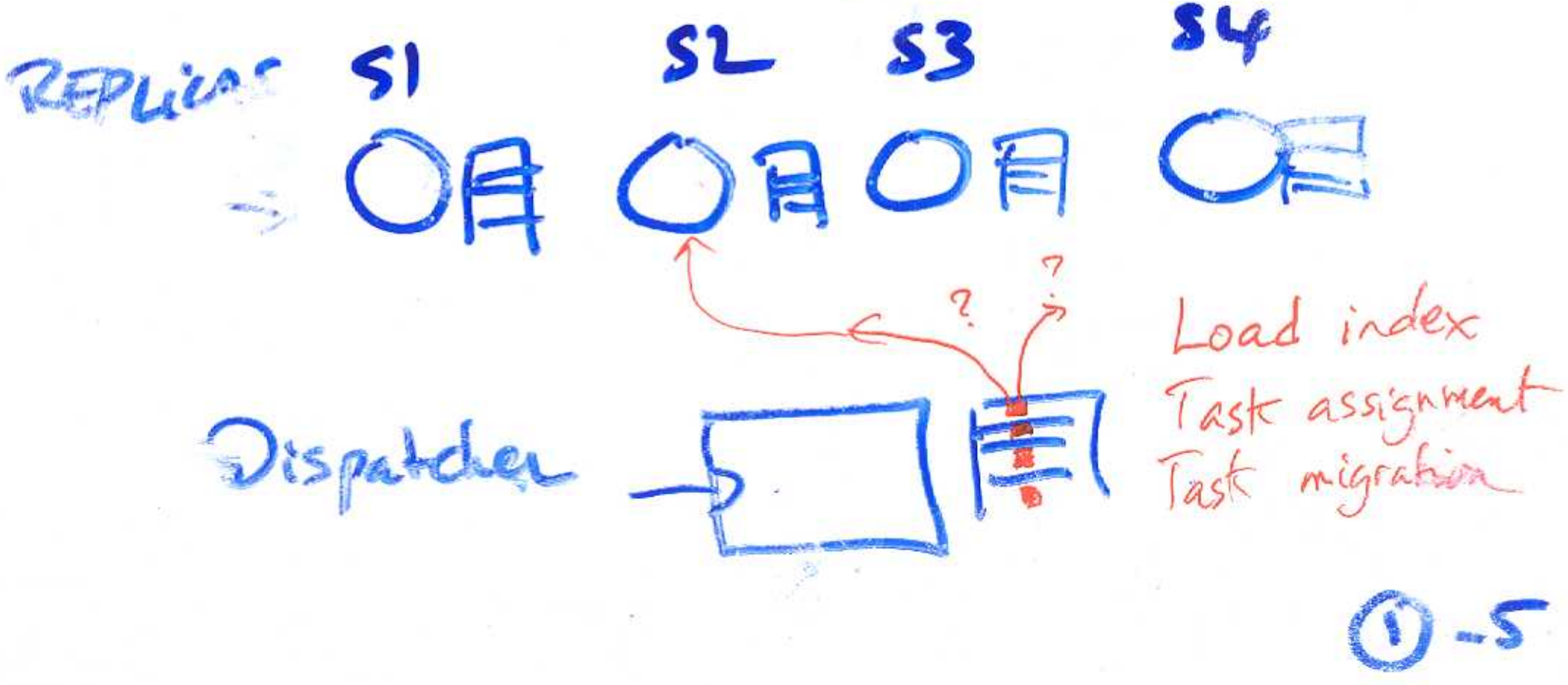
Load balancing

Server side: REPLICATE SERVERS TO PROVIDE BETTER PERFORMANCE



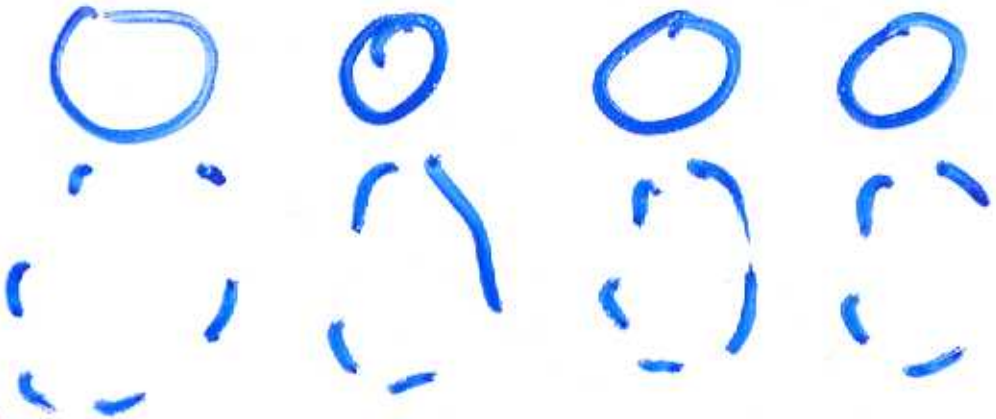
In general, if you have specific load balancing technique (e.g. RANDOM), this could be implemented in CORBA in 2 major ways:

1- Application level
(same like assignment 1)

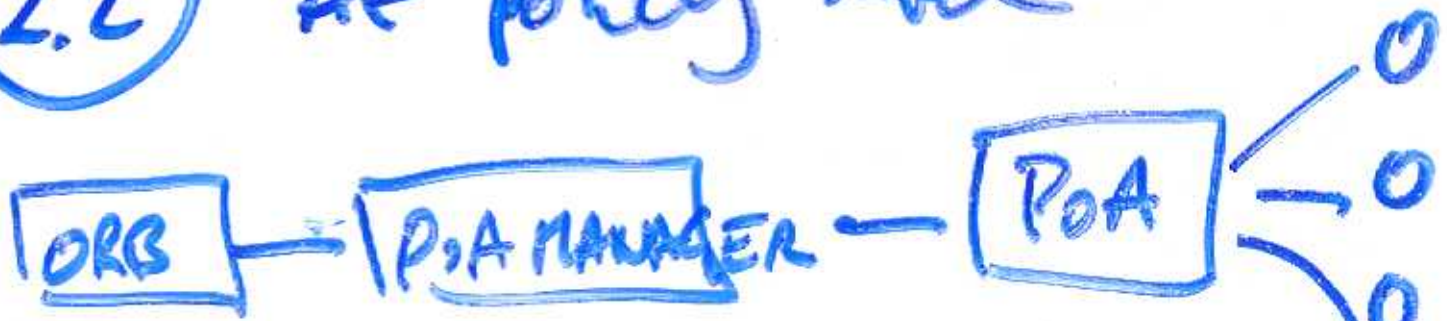
2- System level

(2.1) • POA MANAGER
= DISPATCHER
↳ control flow requests

Replicated POA

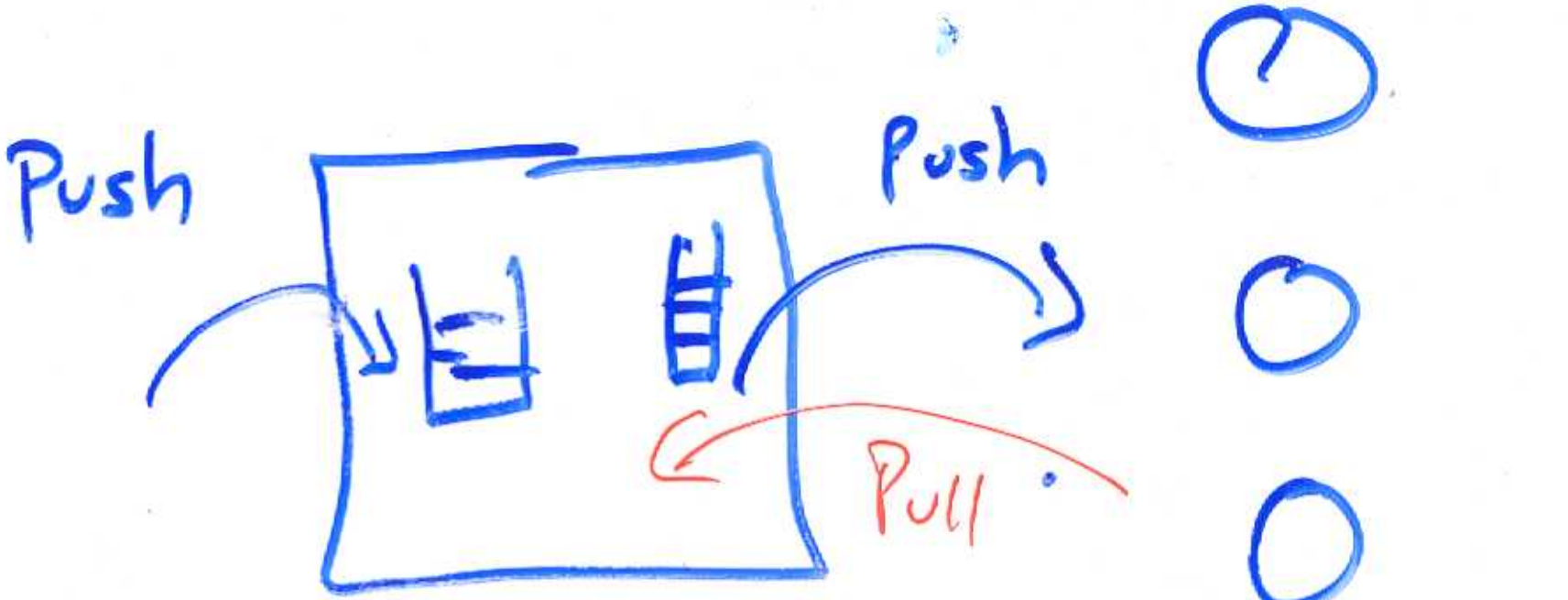


(2.2) At policy level



↳ SERVANT LOCATOR (3-5)
= DISPATCHER

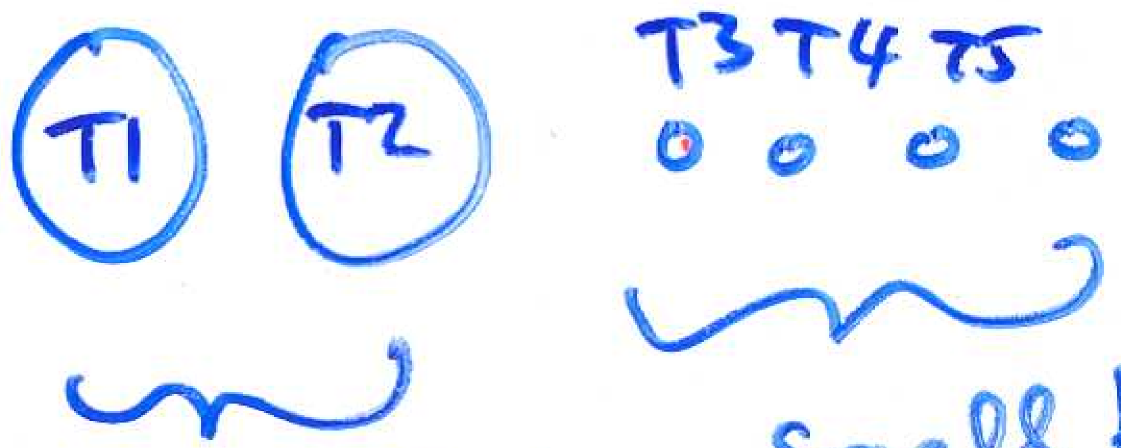
Service level (using CORBA)



DISPATCHER will use

Slow down "fairness"

≃ high task variation



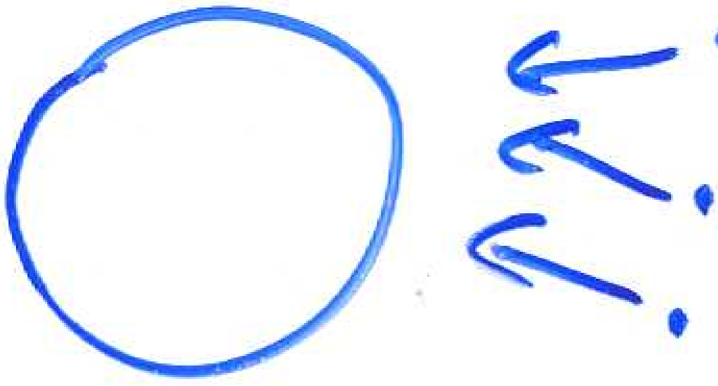
Large tasks

Small tasks

$$\frac{w(T_1)}{size(T_1)} \simeq \frac{w(T_2)}{size(T_2)} \simeq \frac{w(T_3)}{size(T_3)} \ldots$$

if T_3 is stuck behind T2, W(T3) will be very high

(5) - 6

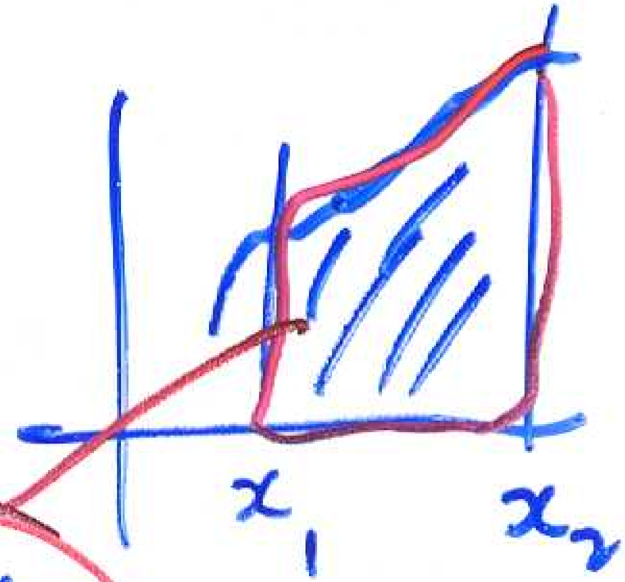


$\leftarrow E\{x\}$: average task size

λ

10 tasks/s

Server S1



$$\rho_{s_1} = \lambda \cdot E\{x\}$$

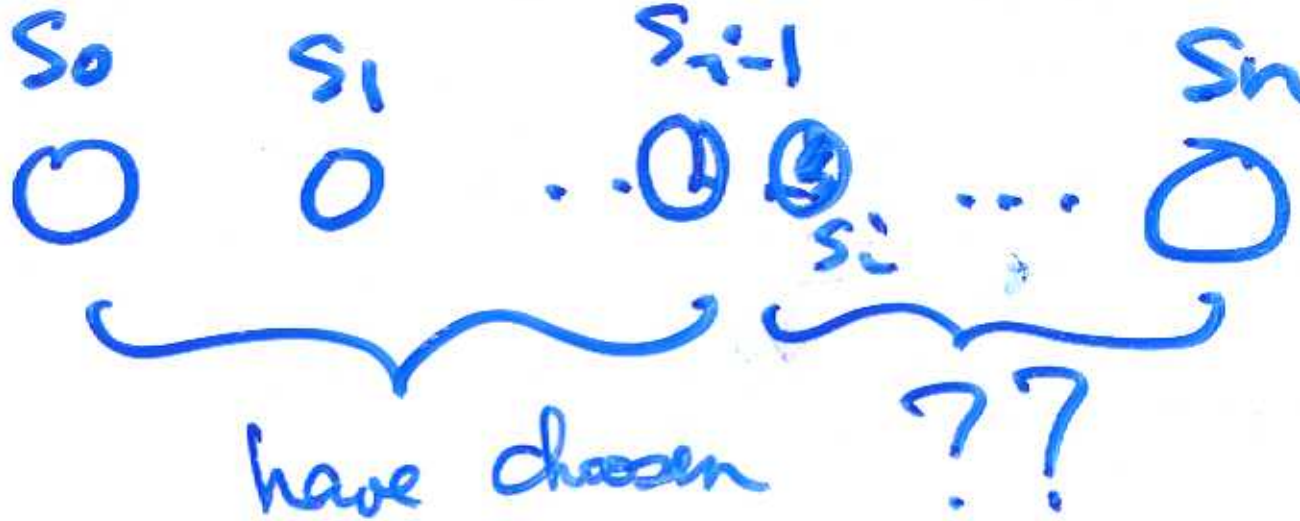
$$= \lambda \cdot \int_{x_1}^{x_2} f(x)\, dx$$

size distribution probability

⑥-5

$$E\{x^2\} = \int x f(x)\, dx$$

$$E\{x^j\} = \int x^{j+1} f(x)\, dx$$



$k \geq i-1$

$\binom{3}{2} = 3$ — a bc

ab
bc
ac

$k=1$

$$P_i = \frac{\binom{99-i}{0}}{\binom{99}{0}} \cdot \frac{1}{100} = \underline{\underline{CST}}$$

CONSTANT

$K \uparrow$ (is high) $\simeq 100$

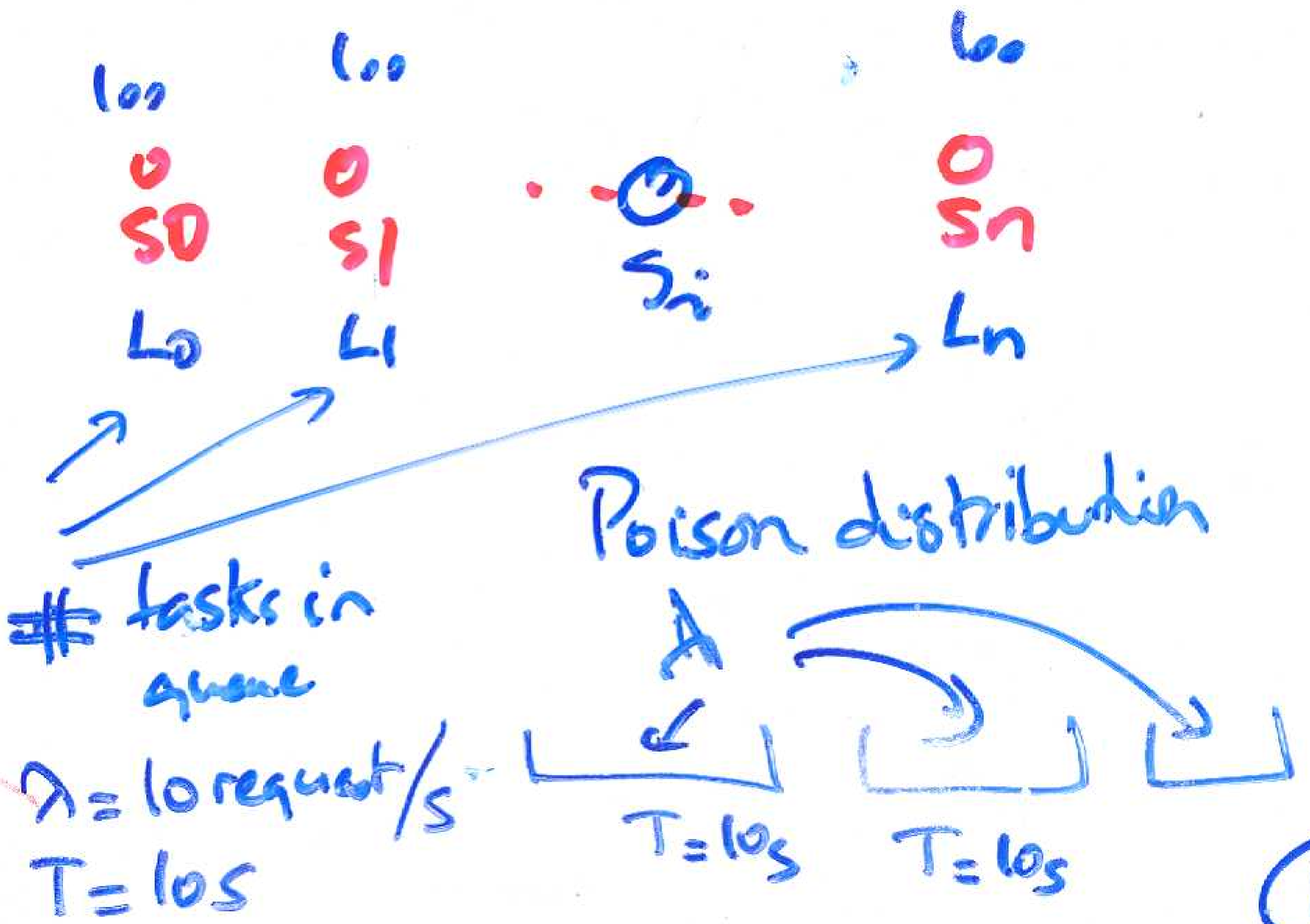
$-i \downarrow$ (small) $\simeq 1, 2, \ldots$

P_i is big $(99 - i)$

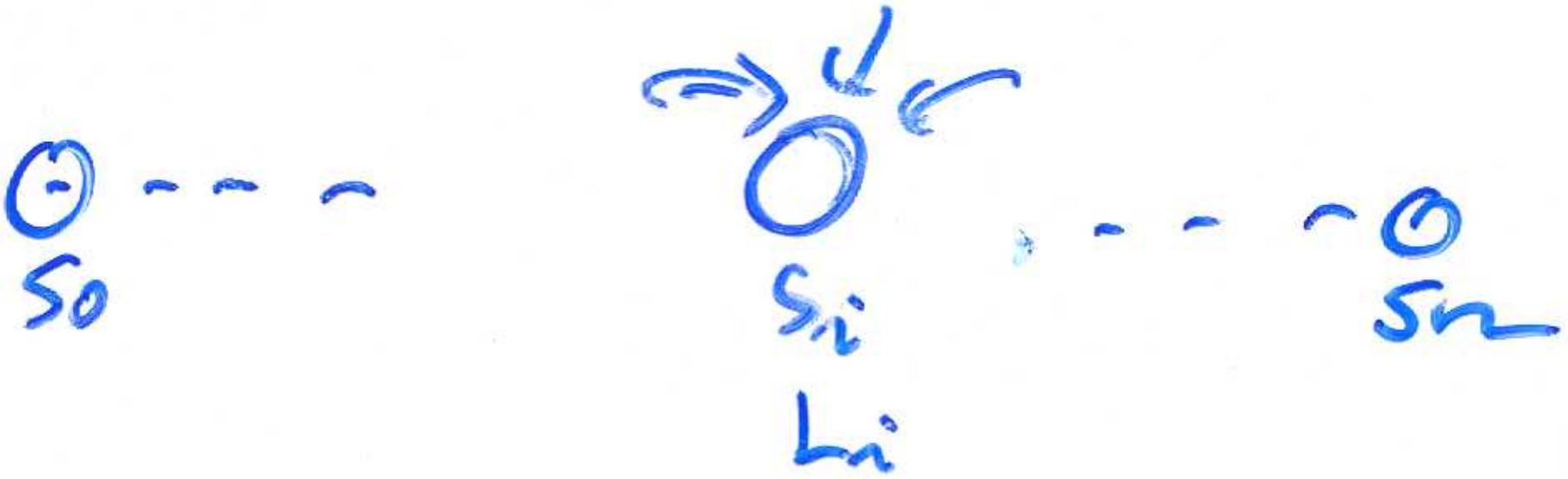
$-i \uparrow$ (big) $\simeq 100, 99, \ldots$

P_i is small $(99 - i)$

Stale information



10 -5



T

T

average per server = (TOTAL + arrivet) / n - Li

(4) - 5