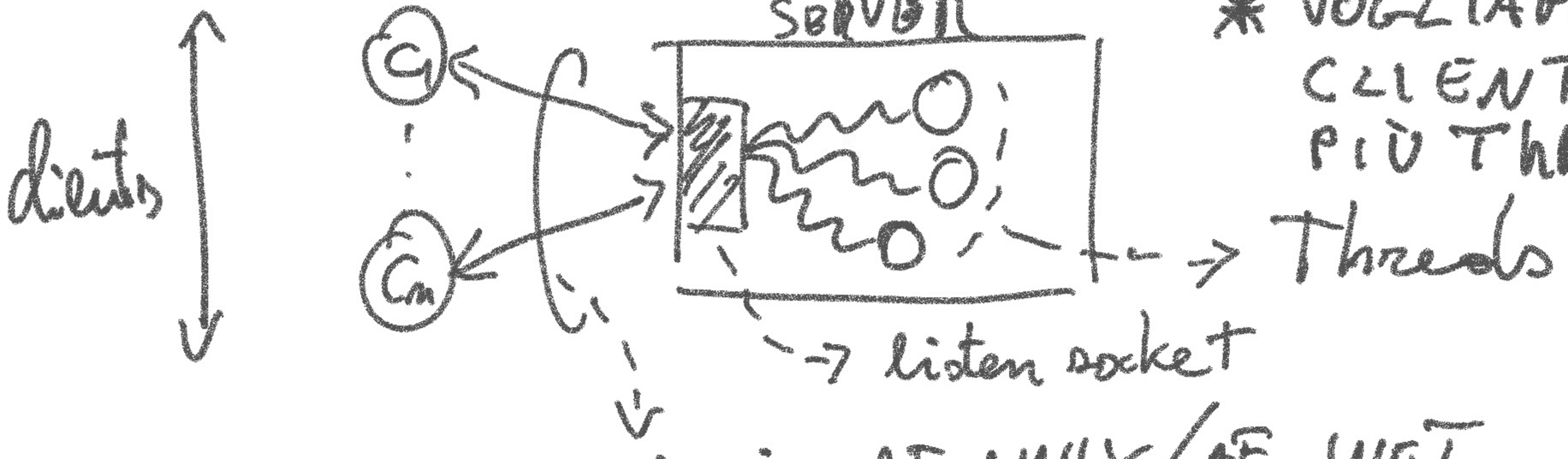
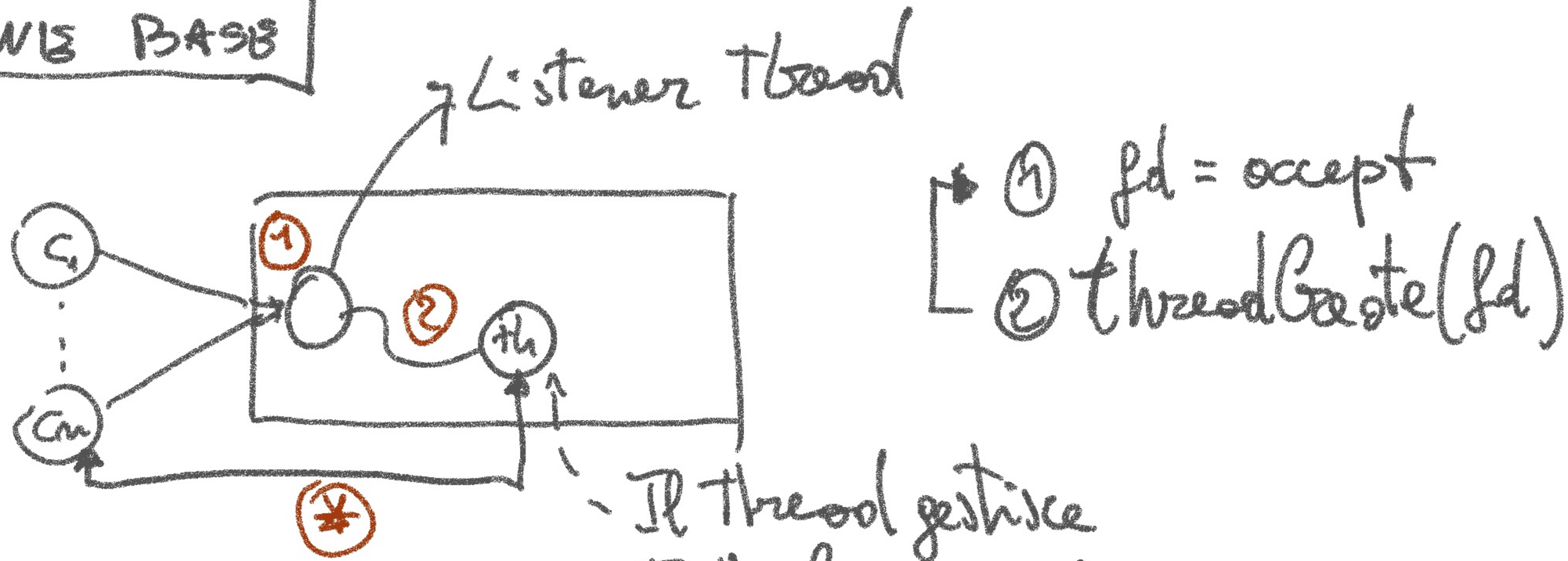
SERVER MULTITHREADED



* VOGLIAMO GESTIRE PIÙ CLIENTS UTILIZZANDO PIÙ THREADS.

+ Per aumentare la reattività del server
+ Per utilizzare al meglio le risorse della macchina.

SOLUZIONE BASE

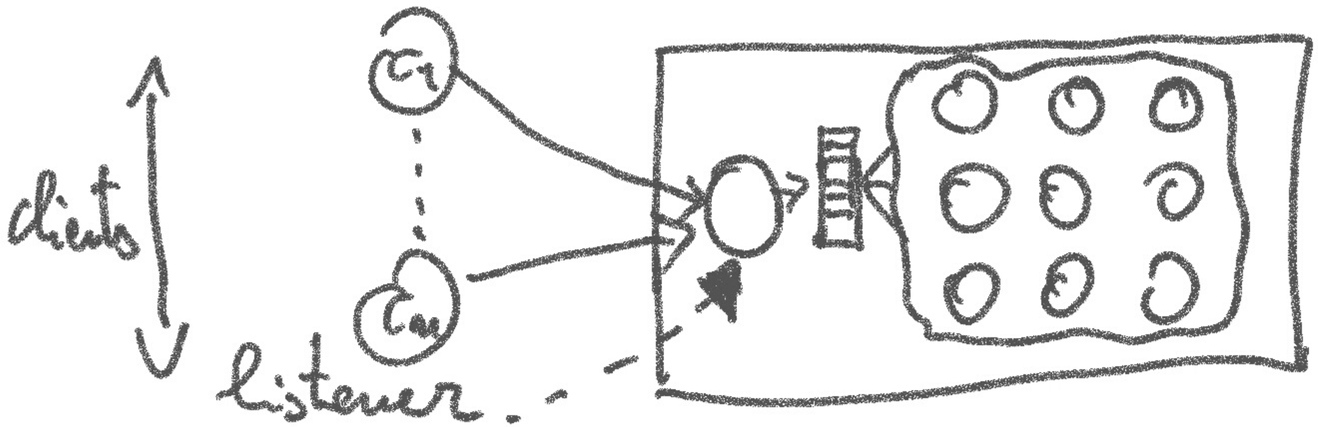


MA:

Fare ogni volta la spawn (in modalità detached) di un thread è costoso.

Perché non riciclare i threads non appena finiscono?

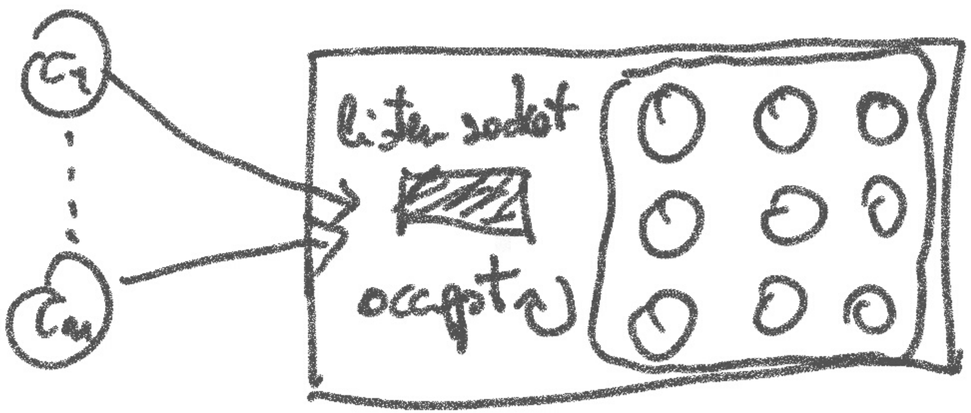
SOLUZIONE CON POOL (1):



- Soluzione con una CODA concorrente.
- Modello MASTER-WORKERS

- Il listener thread accette nuove connessioni ed inserisce il descrittore nella coda.
- I threads del pool (workers) estraggono un descrittore e gestiscono o l'intera connessione oppure il singolo comando.

SOLUZIONE CON POOL (2):



- Soluzione senza CODA, a turno i threads eseguono la accept in mutua esclusione.

+ Tutti i threads tranne 1 sono in attesa su una variabile di condizione - Quando l'accept viene sbloccata si fa subito una signal e si gestisce la connessione -
NOTA: l'accept può essere chiamata da tutti i threads contemporaneamente, ma è meno efficiente!

GESTIONE DELLE CONNESSIONI CON UN POOL

c1- Il thread deve gestire il descrittore gestisce tutta la connessione fino alla chiusura del socket e poi ne prende un altro.

c2- Il thread gestisce 1 richiesta (o in generale K richieste, cioè fino a quando non si blocca su una 'read'/'write'.) . Gestite le richieste estrae un nuovo descrittore e continua

NOTA: Per gestire K richieste devo usare la modalità non-bloccante (fcntl, O_NONBLOCK)

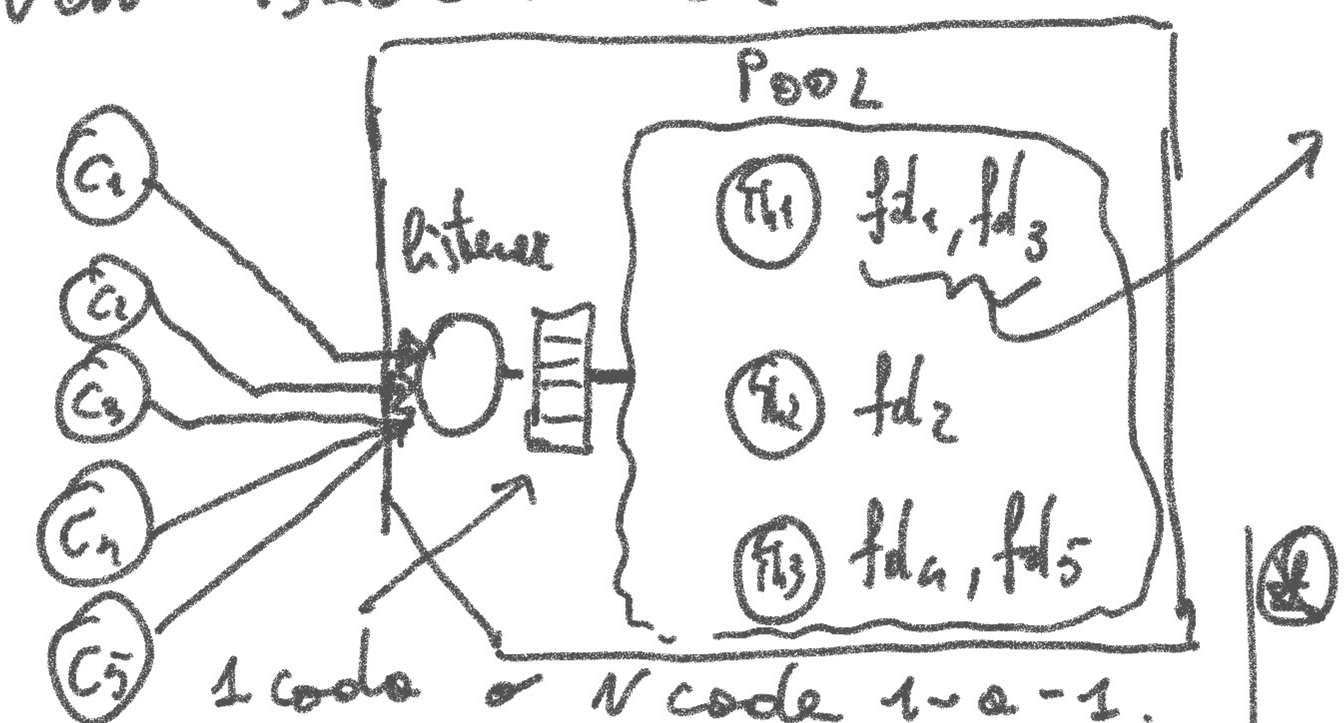
SUPPONIAMO C2, K=1 ED EVENTI SOLO DI LETTURA

- Il listener thread usa select per accettare nuove connessioni e ricevere eventi su socket già connessi.
- Se c'è un nuovo evento, toglie il descrittore dall'"fd_set" e la inserisce nella CODA.
- Un thread gestirà la richiesta ed al termine il descrittore sarà ri-oppinto all'fd_set affinché la select possa ascoltare nuovi eventi.

NOTA: La gestione dell'"fd_set" deve essere fatta in mutua esclusione!

GESTIONE DELLE CONNESSIONI CON POOL (cont.)

- Un'altra possibilità è quella di partizionare i descrittori delle connessioni attive sul pool di threads. Ogni threads gestisce il set di descrittori eseguiti dal listener in modo NON-BLOCCANTE.



Th1:
† select/pool (*) su fd_1, fd_3
† read e/o write non-bloccanti.

(*) Meglio usare libevent.org