

FASTFLOW - IDEE PRINCIPALI

→ Comunicazione tra i nodi:

Message passing Model: chi inizia ha un certo oggetto che deve essere inviato ad un altro nodo di FF.

Non vogliamo usare le lock quindi possiamo usare il message passing model. L'oggetto è salvato in memoria e io ho un puntatore all'oggetto. Quello che faccio è mettere nella coda che è collegata a chi deve ricevere il puntatore e non tutto l'oggetto.

L'altro quindi riceve il puntatore ed evita copie inutili.

Questo mi evita le lock.

Quando il producer mette il puntatore nella coda vuol dire che non toccherà più l'oggetto.

Building Block:

I building block sono come mattoncini Lego che possiamo comporre per creare codici più complessi.

Ci sono già i pattern ma in alcuni casi non vanno bene per le nostre necessità e quindi si deve usare un building block che scrivo per essere adatto alle mie esigenze.

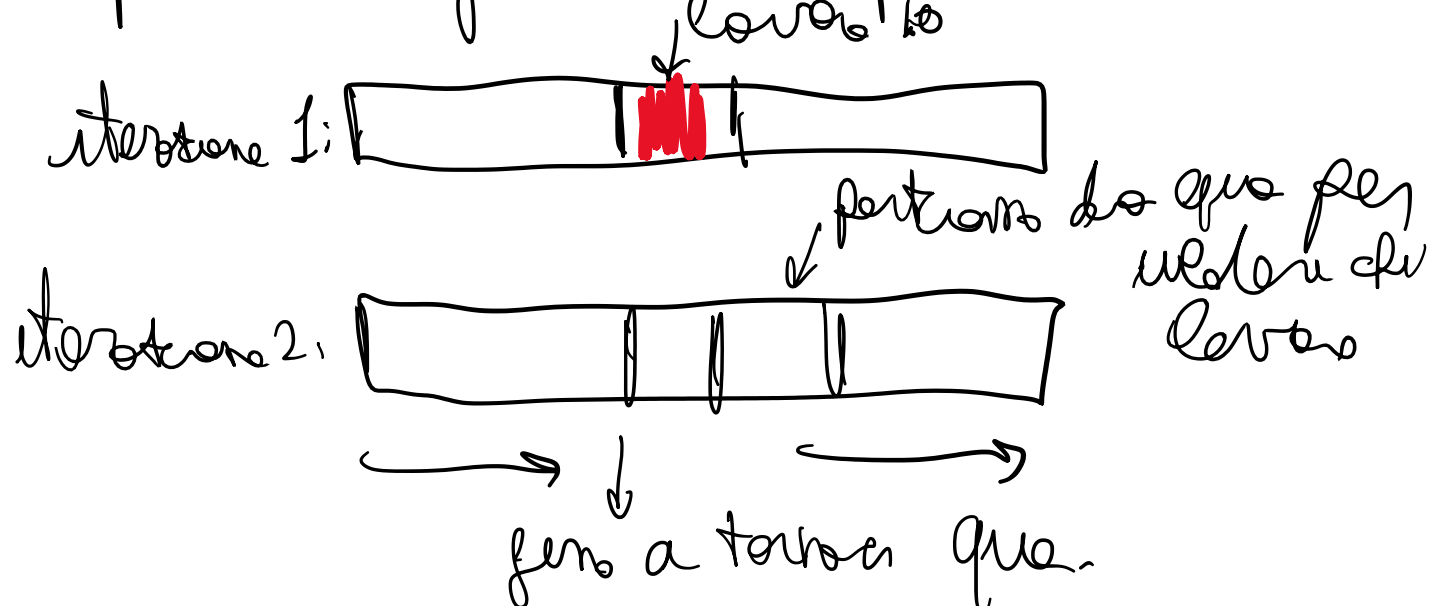
SCHEDULING DATI

- Pipeline → Non ha uno scheduling
- Farm (e Ofarm) → ci sono varie possibilità:

→ Di base, lo scheduling è round Robin, in particolare se le code non hanno una dimensione fissa è completamente round Robin. Se abbiamo una dimensione fissa non è proprio round Robin.

→ Possiamo avere uno scheduling On-demand

→ Possiamo implementare una politica per lo scheduling, ad esempio possiamo avere un array con i worker, dare il lavoro ad uno e poi partire dal successivo per assegnare il prossimo lavoro.



Divisione in chunk:

Se abbiamo fare una map o una map-reduce è importante la divisione in chunk:

- Dividiamo i dati in base al numero di worker, in questo modo però ci troviamo ad avere una divisione che potrebbe essere sbilanciata.

Nel senso che magari un worker lavora di più, gli altri meno e quindi abbiamo dei tempi di attesa di chi ha lavorato di più.

- Facciamo una divisione dinamica, nel senso che diamo parte del lavoro ai vari worker e poi ad un certo punto ci mettiamo gli diamo altro lavoro in modo da bilanciare.

