Hidden State:
σj:the vector to represent number of requests of processing batch in each replica

I ∈ {0, 1}^n:the vector to represent whether each replica is serving queue 1 or
queue 2, where 0 indicates queue 1 and 1 indicates queue 2.

N(t): represent how many request arrive before time t.

State Space:

X(t): contain the queue length of queue1 and queue2 in each replica. Here we use a 2*n vector to represent it.

- Replica 1 queue 1 length
- Replica 1 queue 2 length
- Replica 2 queue 1 length
- Replica 2 queue 2 length
- ……..

Y(t): contain the prefilled tokens of each request in queue 1, processed tokens of each request in queue 2 and reserve tokens of each request in queue2 across all replica. Here we will use the M * 3n matrix to represent it, M is the value we set for the maximum length for each queue.

Will not be observed by the

M*1 vector to store processed tokens for each request in queue 2 replica 1	M*1 vector to store prefilled tokens for each request in queue 1 replica 1	M*1 vector to store reserved tokens for each request in queue 2 replica 1	M*1 vector to store processed tokens for each request in queue 2 replica 2	M*1 vector to store processed tokens for each request in queue 2 replica 2	M*1 vector to store reserved tokens for each request in queue 2 replica 2	……

Z(t) = (T(t),W(t)), where T(t) represent the inter arrival time for next request, W(t) represent the reserved time for each replica. We use n+1 dimension vector to represent it.

Action Space: Allocate the request to different sever, here I use the {0,1} ^n*N matrix, where N is the total number of requests arrive.

Parameter：
number of tokens in block
- NumBlocks:The total number of blocks in replica
- Watermarkblocks: The number of blocks reserved in replica
- Maxtokensinbatch: The maximum number of blocks in Batch
- Configbatchsize: The maximum number of requests in replica
- maxmicrobatchsize: The maximum number of requests in a batch

Event Space:

Event 1: The request arrive at queue 0	Event 2: The replica 1 complete the batch processing	Event k: The replica k-1 complete the batch processing	Event n+1: The replica n complete the batch processing

Find out the argmin (z(t)) to deduce the next event

If event 1: using n(t) and action to find out Which replica we are going to send the request to.

Update x(t),y(t),z(t)

If event k+1: using the information of σj and I to update y(t)

Comparing the 3k column and 3k-2 column to check whether there's request complete.

Update the x(t),y(t),z(t), and come up the new batch depend on the function I mentioned in paper. Update σj and I.

If we want to change the batching policy, we can use different function here.