* async queuing system, cont.

How many consumers to run?

- depends on system conditions, and the nature of the work consumer does
- rule of thumb: have as many as you have free CPU cores
- if fewer consumers than CPU cores, cores idle, can get more throughput
- if more consumers than CPU cores, scheduling issues, not really better throughput

Prod/consume rates and queue's eventual states:

R(p): rate that producers create new work
R(c): rate that consumers process queued work
N: fixed size of queue

R(p) >> R(c): eventually queue is full, regardless of length, and producers have to be throttled.

R(p) << R(c): eventually queue is empty, consumers are idle (waiting)

R(p) e> R(c): (producer rate is just SLIGHTLY faster than consumer, by some epsilon). Eventually, queue will also be full, but take longer to get to that steady state.

R(p) <e R(c): (producer rate is just SLIGHTLY lower than consumer, by some epsilon). Queue also empty eventually.

R(p) == R(c): producer and consumer rates match perfectly. Queue is never empty or full. Very hard to get to perfect equilibrium. OSs and dist. systems. try very hard to adjust rates and #consumers and #producers to get close.

E.g., in linux, the pdflush with two thresholds:

- producers -- user processes touching file data
- consumers -- pdflush kthreads to write dirty data to f/s

problem: in systems w/ different I/O devices, SSDs are faster than HDDs. Can't assume that consumer rates are fixed.

New system: Backing Device Information (BDI)

- measure rates of consumers (I/O rates) to each media and even each f/s
- dynamic load balancing and proportional sharing
- e.g., HDD is 3 times slower than SSD, so can adjust consumer rates -- 3 consumers for SSD devices for each consumer to an HDD device.