1. Because access data in memory is time-consuming, you want to memory that is closer to CPU than main memory and that only store frequently-accessed data(because SRAM is expensive) so that you can access those data faster.
2. Amdahl’s law : 1/ {(1-p) + p/n} , where n is how many cores you have and p is the parallel part of the program you have.

The total S is 33%

1. 1/{(1- 0.67) + 0.67/ 1} = 1 ;
2. 1/{(1- 0.67) + 0.67/ 2} = 1.5037 ;
3. 1/{(1- 0.67) + 0.67/ 4} = 2.010 ;
4. 1/{(1- 0.67) + 0.67/ 8} = 2.417 ;
5. 1/{(1- 0.67) + 0.67/ 12} = 2.591 ;
6. 1/{(1- 0.67) + 0.67/ 16} = 2.689 ;
7. 1/{(1- 0.67) + 0.67/∞ } = 3.030 ;
8. Concurrency refers to breaking program or programs into pieces and run with random order so that multiple tasks can be done within certain time span, but parallel refers to running two parts of a program or programs at the exact the same time.

Concurrent with one core: programs are broken into pieces and schedule those pieces to run in one core one by one.

Concurrent with three cores: programs are broken into pieces, dispatched to each core, and run simultaneously.

Parallel with three cores: programs are dispatched to each core and run simultaneously.

1. For one thing, SRAM is expenses. For another, cache is an associative memory which implies that with the algorithm available so far, the bigger the cache is the longer it takes to access data in cache.
2. The main goal of cache is to speed up the memory access time by leveraging spatial and temporal locality. Cache serves for the same purpose for both single and multi-cores processor, but the implementations are different because in multi-cores scenario, hardware engineers have to consider the coherence issue to guarantee the write coherence in the shared memory. In single processor scenario, this issue will not occur.
3. PRAM. If requests from customers do not require heavy computation, we only need one computation node to handle all the task and do is leverage multiple threads to fulfill all the requests. If both the computation power and storage is enough for the current task, there is no need to pay extra cost for message passing between computation nodes just for getting more computational power or storage.