* **Comprehension**: Describe interference
* thread safety,
* process synchronization
* critical section problem.
* **Application**: Use semaphores (critical section, synchronization, resource counting).

**Counting semaphores vs resource counting semaphores**: counting count the workers

* **Knowledge**: Memorize hardware and OpenMP atomic-action primitives.
* **Comprehension**: Describe the uses of processor affinity and remote procedure call to implement parallel algorithms.
* **Knowledge**: Memorize the four necessary conditions for deadlock.

**Mutual exclusion, hold and wait, no preemption, circular wait**

* **Comprehension**: Describe the properties of resource deadlock prevention, handle just one of the deadlock conditions
* avoidance Banker’s algorithm
* detection/recovery. Process termination and resource preemtion
* **Comprehension:**Describe the advantages of message-passing using MPI for distributed programming

**Multiple nodes with their own filesystems.**

* **Comprehension:**Describe the uses of MPI send : send message to specific node
* Receive receive message
* broadcast: send message to all nnodes
* scatter
* gather   
  reduction operations.
* **Knowledge:**Memorize the definition of RPC
* Stub: packs parameters into a message (marshaling)
* procedure
* IDL specifies how parameters are handled during a call
* **Knowledge:**Memorize the ACID properties of a transaction.

**Atomic, Consistent, Isolation, Durable**

* **Knowledge:**Memorize the definition of dataflow computing.

**Data goes from one piece of a program to another and becomes altered in that piece.** Those pieces of the program do not interact with eachother.

* **Analysis:**Compare and contrast Hadoop's and BOINC's approach to cloud computing.

**HADOOP:** MapReduce, massive file system. BOINC: People “volunteer ” processing power (See folding@home)

* **Comprehension:**Describe the three steps in Hadoop's map-sort-reduce processing model.

**Map:** break up file

**Combine:** use memory to create list

**Reduce:** reduce pieces back into one file