

c. Scheduler activations permit a user-level process to manage its own threads. To manage a machine's processors, however, additional support is needed. Describe sample functionality an OS might offer for managing its processors, specifically addressing the need to run PARALLEL programs comprised of multiple address spaces and threads.

d. *Differentiate the following two strategies in the Cache Affinity paper: LMI (limited minimum intervening) and LMR (LMI routing). Then explain why LMR has less variance than LMI.

e. *When targeting low latency for message communications, both the protocol stack and the interactions with network interface cards must be considered. List two issues with protocol stack implementation that tend to cause performance problems. Then list two issues with network interfaces that can compound such protocol level problems.

2) Medium Length Question (25 mins.):

Active networking appears to be a solution in search of a problem, at least to its critics. In response, in this problem, you are to design a sample active networking application, using the capsule-based approach. It is not important that you remember exactly the capsule API. Instead, just design some suitable API, then go with it.

The application you are asked to implement is one that

- looks at a packet's payload (its body)
- extracts some information from its application-specific header, assuming that this header provides the type and layout information on all data contained in the body (like the interface descriptions used in the implementation of RPC)
- then filters (eliminates) some of this data and forwards the filtered packet to wherever else it's supposed to go,
- and/or replicates (mirrors) packets based on header information,
- or both.

Mirroring a packet is an application-level form of multicast, where certain packets, based on their payload, are sent to multiple sites.

Note that, in general, the data contained in a packet generated as a result of applying a capsule's function (action) to an incoming packet's data is of some type that is different from the incoming packet's type.

The specific application is one that processes flight data received from the FAA (e.g., planes' current positions, heading, speed, etc.).

- a) Describe your capsule API.
- b) Describe when/why you might want to do mirroring. Be explicit, basically, describe the mirroring function (in something like pseudo-code).
- c) Describe when/why/what processing you might want to do on packet bodies.

- d) Be sure to design a processing chain in which a single data stream is processed by **MULTIPLE** capsule actions as it traverses the network's active nodes. Explain this processing. Again, be explicit. Use your API.
- e) Give at least one example of processing that is not viable in the network (consider state!).
- f) Discuss the probable performance of your implementation, in **COMPARISON** with the performance of a similar implementation on host machines, at application level.
- What elements of your application would perform better at application level?
 - What elements would perform worse?