Supporting Paradigms On Converse

- Paradigms supported
  - Message Driven Objects
  - SPMD (PVM, NXLib)
  - Threads (with message passing)
- Common facilities needed for supporting paradigms/language runtimes
  - Message managers
  - Thread objects

Message Managers

- Essentially, indexed mailboxes
- Provide calls for tag based insertion, retrieval and probing for messages
- Wildcarded retrieval is allowed
- Support message sequencing, optionally

Thread Objects

- Thread functionality seperated modularly
  - Thread state (stack, registers, caller etc)
  - Scheduler (queue of threads)
  - Synchronization mechanisms (locks and condition variables)
- Provides calls for
  - create : specifies stack size, scheduler, starting function ..
  - resume
  - yield
  - exit
  - awaken : Enqueue the thread object into its scheduler's queue
  - suspend : pause the execution of the current thread

Thread Objects (Cont.)

- Portable with low overhead
- Each thread can have a different scheduler
- Different types of schedulers can be built
  - round robin, hierarchical, message driven
- Threads can be scheduled using Converse scheduler

Simple Messaging - SM

- Example of a "language runtime"
- Tag based point to point communication primitives
  - send, probe and blocking receive.
- SPMD SM and Thread SM
- Handlers either place message into message manager or continue suspended computation

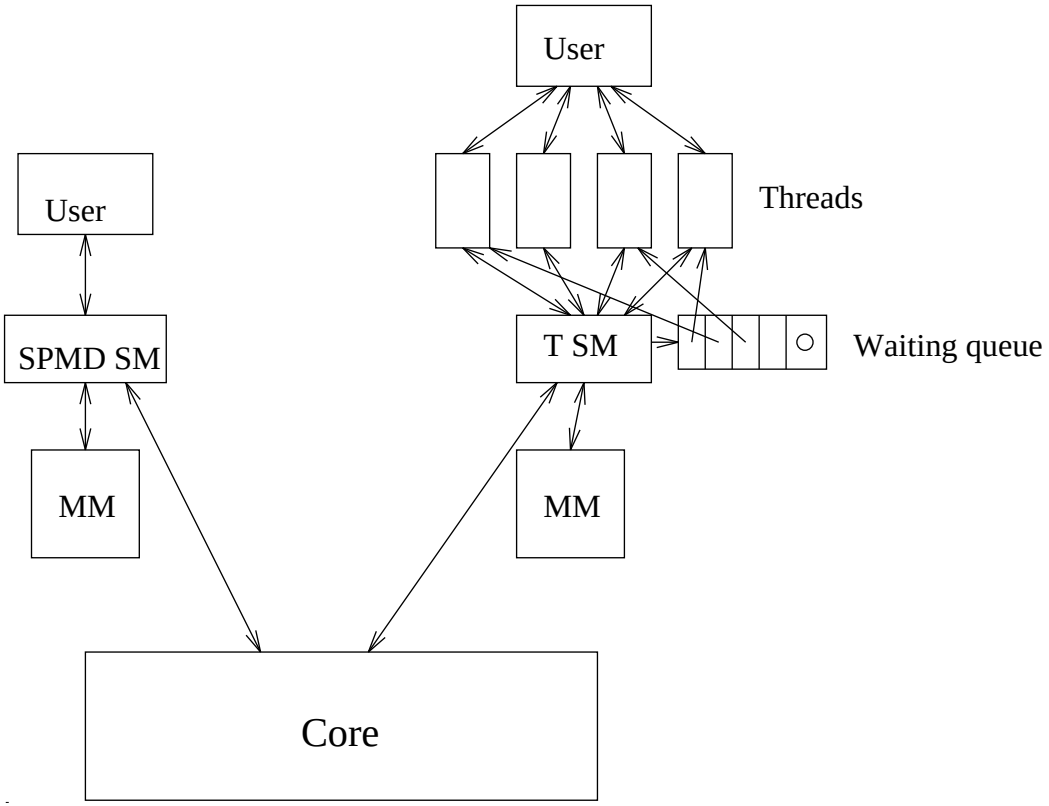

Figure 1: SM

Converse Status

- Languages implemented using Converse
  - PVM-messaging, NXLib, SM
  - Threaded SM
  - Charm and Charm++
  - Planned : MPI, ...
- Converse runs on
  - Workstation networks (using UDP/IP, ATM, FM/Myrinet)
    - Reliable UDP implementation using packetization and windowing.
    - FM : packetization for large messages.
  - IBM SP-2, Cray T3D/FM, Intel Paragon
  - being ported to other machines