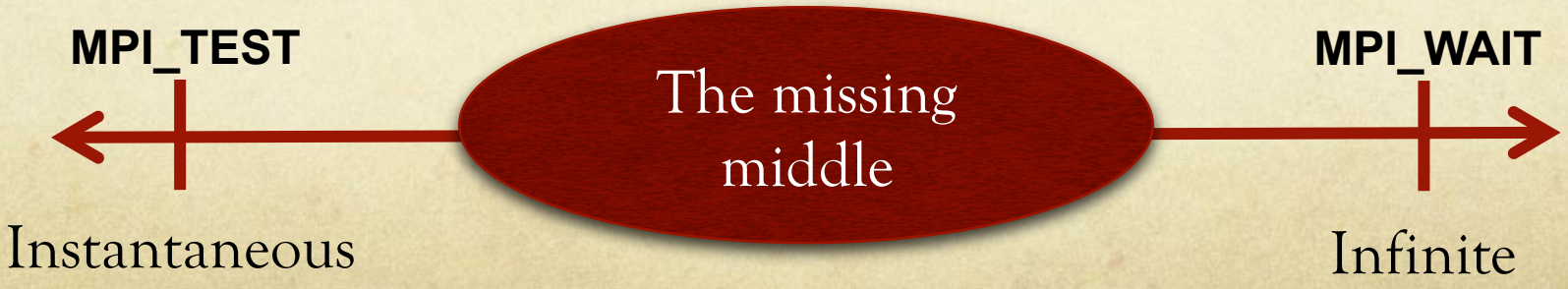

MPI Timers
Tick tock tick tock

Wouldn't it be nice...

- To block in MPI_WAIT*
  - ...but only for a while? (even if nothing completes)
- Useful for:
  - Checking progress of non-MPI things
  - Avoid putting MPI into its own progress thread
  - Avoid using MPI_THREAD_MULTIPLE
  - Dynamically updating array of requests to MPI_WAIT*



New type of MPI_Request: timer

- MPI_TIMER_CREATE(double completion_time, MPI_Request *request)
  - Creates a new timer request
  - Request can be tested, waited, canceled, and freed – just like any other request
- Will complete when MPI_Wtime() >= completion_time
  - Completed timer requests return the empty status

Use cases

- MPI_WAIT
  - Completes when MPI_Wtime() >= completion_time
- MPI_WAITANY (including a timer request)
  - Timer request may be the first to complete
  - Forces a "premature" return from WAITANY
- MPI_WAITALL (including a timer request)
  - Same as usual: completes when all complete
  - Timer request sets a lower bound on when all complete
- Works with all MPI_WAIT* and MPI_TEST* functions (obviously)

Use cases

- MPI_CANCEL a timer request
  - Cancels just like any other request (i.e., optional)

    → Although we expect all MPI implementations to be able to implement canceling timers easily
  - Still has to be tested or waited, just like any other request
- MPI_REQUEST_FREE a timer request
  - Moral equivalent of a successful MPI_CANCEL followed by an MPI_TEST

Use cases

- MPI_Timer_reset(double completion_time, MPI_Request *request)
  - Lets application re-use timer requests cheaply
  - Any timer can be reset if it has not yet been completed via TEST or WAIT
  - Specifically: it is ok if MPI_Wtime() >= completion_time

- See example on next slide →

Restarting timer in a loop

```
MPI_Request req[21];

fill_20_requests(req);
MPI_Timer_create(MPI_Wtime() + 5, &req[20]);

while (1) {
   MPI_Waitany(21, req, &index, MPI_STATUS_IGNORE);
   if (index != 20) {
       go_handle_completed_request(req, index);
       check_for_other_progress();
       MPI_Timer_restart(MPI_Wtime() + 5, &req[20]);
   } else {
       check_for_other_progress();
       MPI_Timer_create(MPI_Wtime() + 5, &req[20]);
   }
}
```

If the Forum likes the idea...

- What chapter should text about timers go in?
  - Environment control (with WTIME, WTICK)
  - Point to point (with TEST, WAIT, etc.)
  - ...?