

Re: [RFC,PATCH] loopback: calls netif_receive_skb() instead of netif_rx()

Re: [RFC,PATCH] loopback: calls netif_receive_skb() instead of netif_rx()

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg11942.html>

- *From:* David Miller <davem@xxxxxxxxxxxxxx>
 - *Date:* Mon, 31 Mar 2008 04:02:42 -0700 (PDT)
-

From: Ingo Molnar <mingo@xxxxxxx>
Date: Mon, 31 Mar 2008 12:44:03 +0200

and it's not just about scalability, the plain algorithmic overhead is way too high as well:

```
$ taskset 1 ./bw_tcp -s
$ taskset 1 ./bw_tcp localhost
Socket bandwidth using localhost: 2607.09 MB/sec
$ taskset 1 ./bw_pipe
Pipe bandwidth: 3680.44 MB/sec
```

Set your loopback MTU to some larger value if this result and the locking overhead upsets you.

Also, woe be to the application that wants fast local interprocess communication and doesn't use IPC_SHM, MAP_SHARED, pipes, or AF_UNIX sockets. (there's not just one better facility, there are *_four_!*)

From this perspective, people way-overemphasize loopback performance, and 999 times out of 1000 they prove their points using synthetic benchmarks.

And don't give me this garbage about the application wanting to be generic and therefore use IP sockets for everything. Either they want to be generic, or they want the absolute best performance. Trying to get an "or" and have both at the same time will result in ludicrous hacks ending up in the kernel.

—
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>

Re: [RFC,PATCH] loopback: calls netif_receive_skb() instead of netif_rx()