ETA:
Experience with an IA processor as a Packet Processing Engine

Hot Interconnects Symposium
August 2003

Greg Regnier
Intel Network Architecture Lab

Intel
Research &
Development

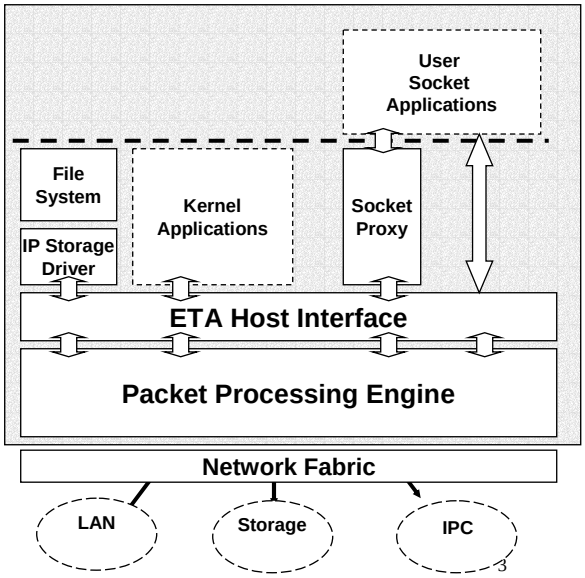
ETA Overview (Embedded Transport Acceleration)

- ETA Architectural Goals
  - Investigate the requirements and attributes of an effective Packet Processing Engine (PPE)
  - Define an efficient, asynchronous queuing model for Host/PPE communications
  - Explore Platform and OS integration of a PPE
- ETA Prototype Goals
  - Use as a development vehicle for measurement and analysis
  - Understand packet processing capabilities of a general-purpose IA CPU

Intel
Research &
Development

ETA System Architecture

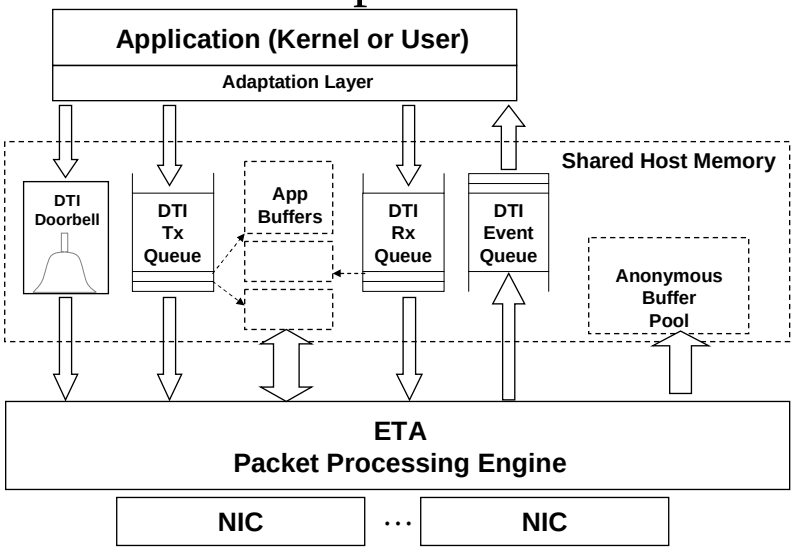
- **Network stack**
- **Virtualized, asynchronous queuing and event handling**
- **Engine Architecture & platform integration**



Intel Research & Development

3

Direct Transport Interface

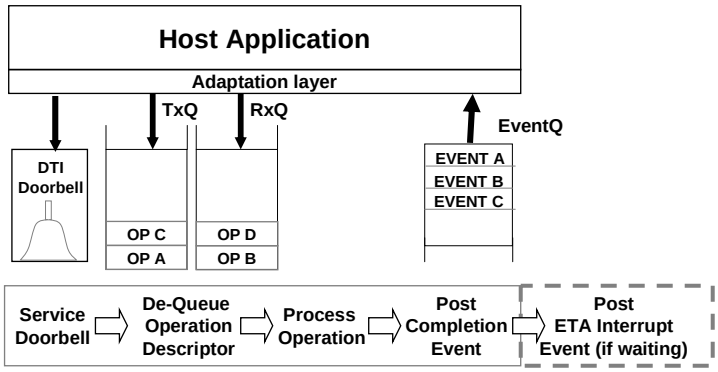


Intel Research & Development

4

DTI Operation Model

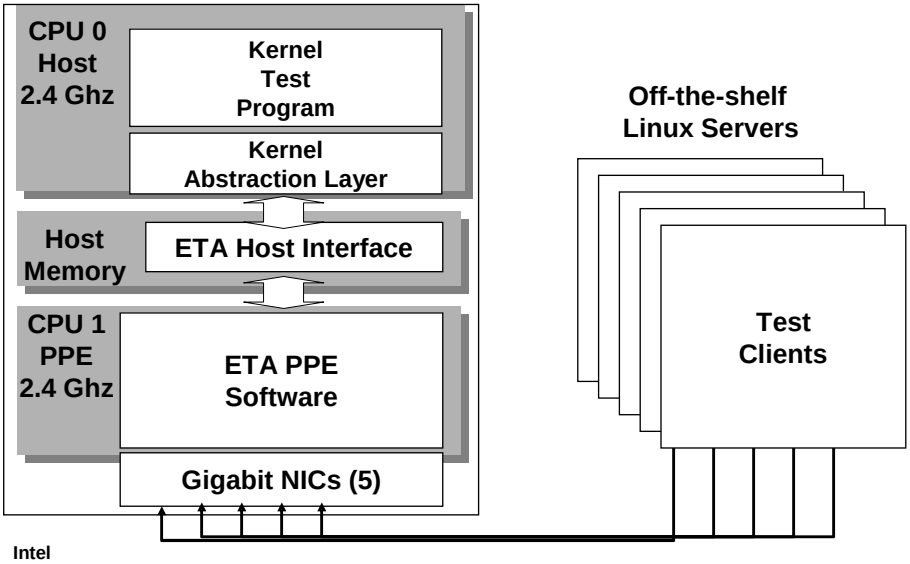
- DTI operations:
  - •Connection requests (Connect, Listen, Bind, Accept, Close, …)
  - •Data transfer requests (Send, Receive)
  - •Misc. operations (Set/Get Options,…)



Intel Research & Development

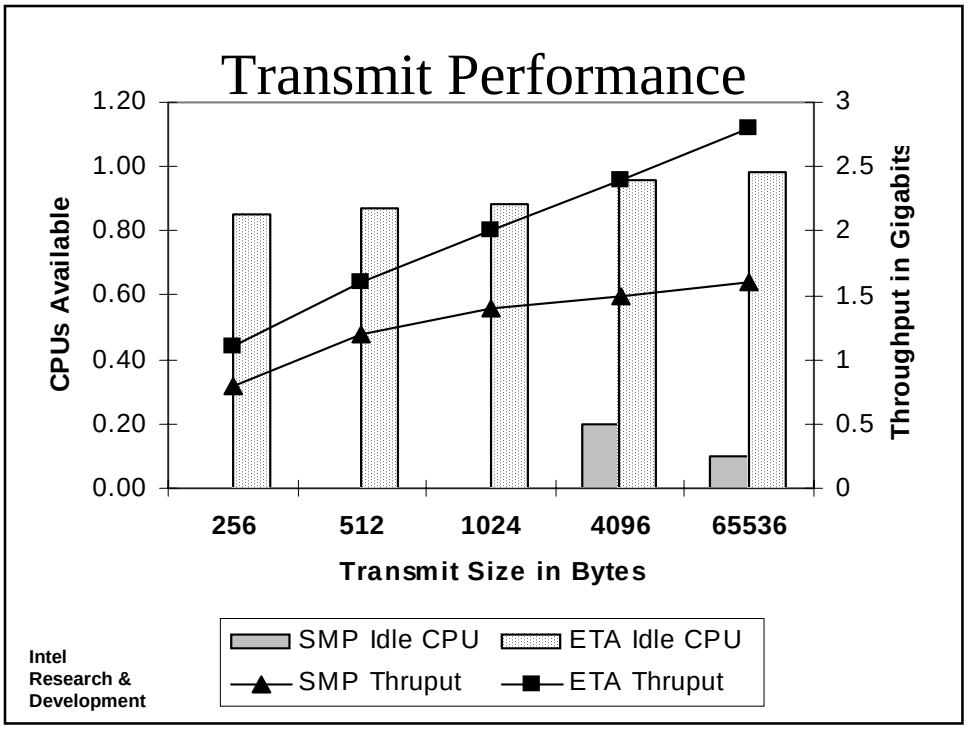
5

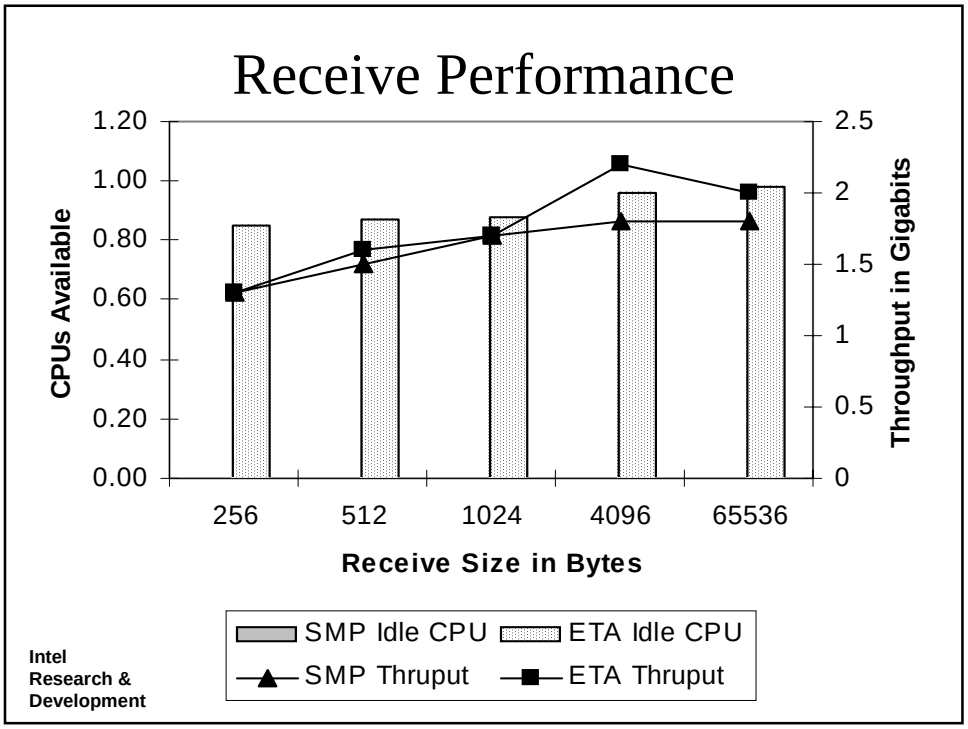
ETA Test Environment

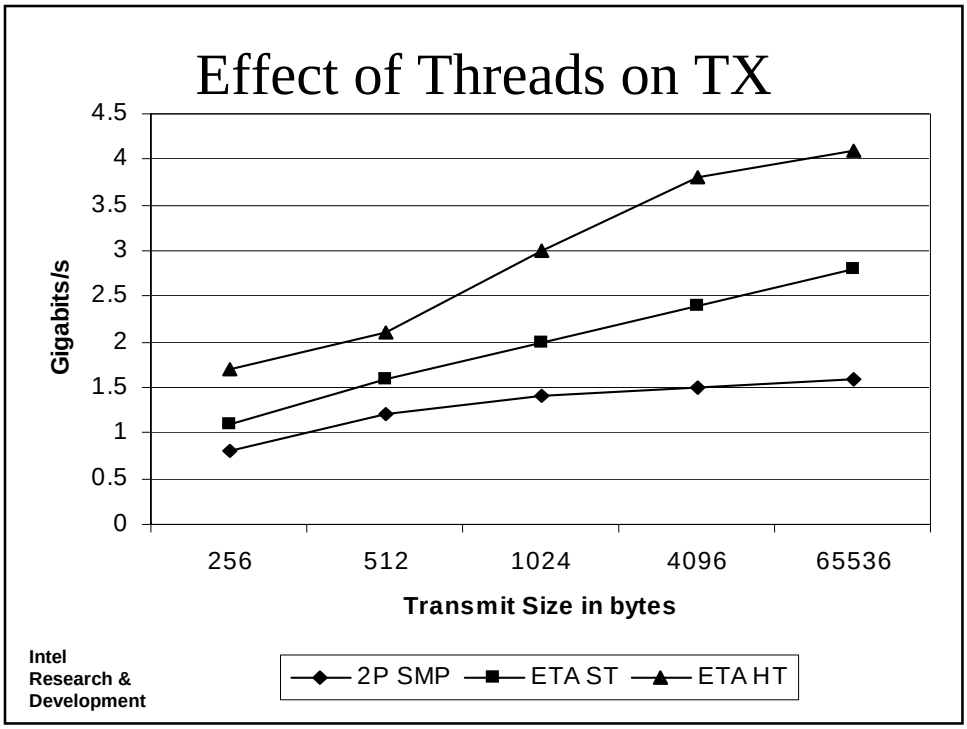


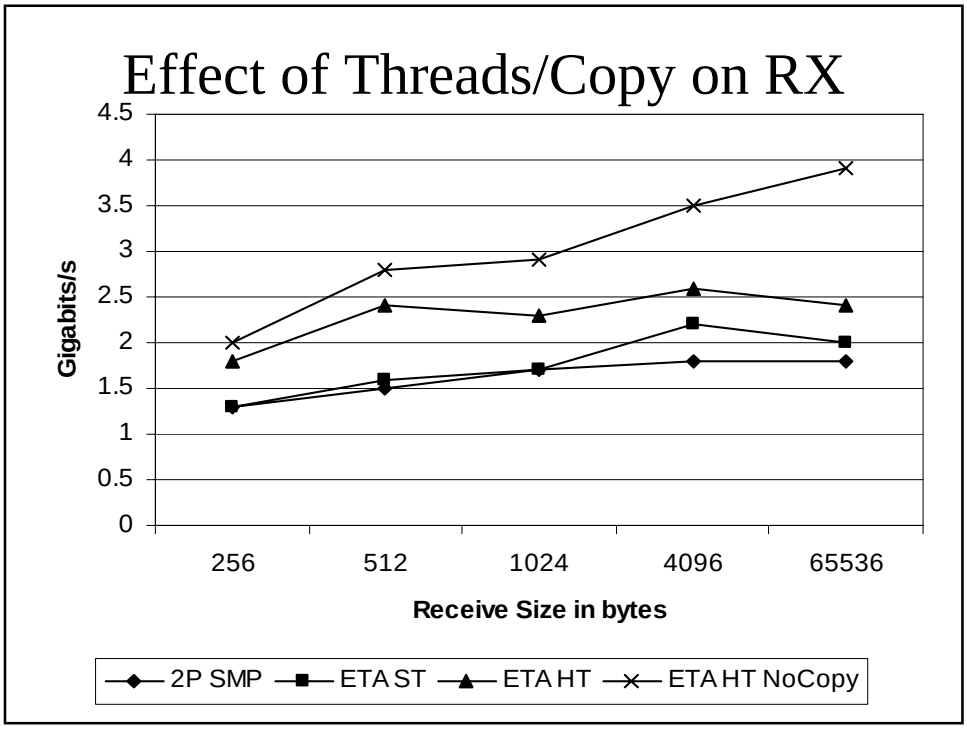
Intel Research & Development

6


Transmit Performance
1.20
1.00
0.80
0.60
0.40
0.20
0.00
CPUs Available
3
2.5
2
1.5
1
0.5
0
Throughput in Gigabits
256
512
1024
4096
65536
Transmit Size in Bytes
SMP Idle CPU
ETA Idle CPU
SMP Thruput
ETA Thruput
Intel
Research &
Development

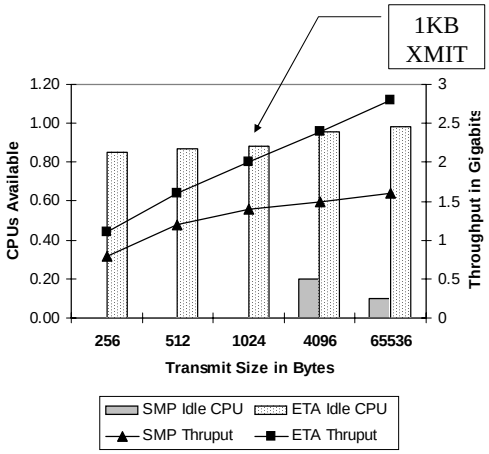

Receive Performance
1.20
1.00
0.80
0.60
0.40
0.20
0.00
CPUs Available
2.5
2
1.5
1
0.5
0
Throughput in Gigabits
256
512
1024
4096
65536
Receive Size in Bytes
SMP Idle CPU
ETA Idle CPU
SMP Thruput
ETA Thruput
Intel
Research &
Development


Effect of Threads on TX
Gigabits/s
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
256
512
1024
4096
65536
Transmit Size in bytes
Intel Research & Development
2P SMP
ETA ST
ETA HT

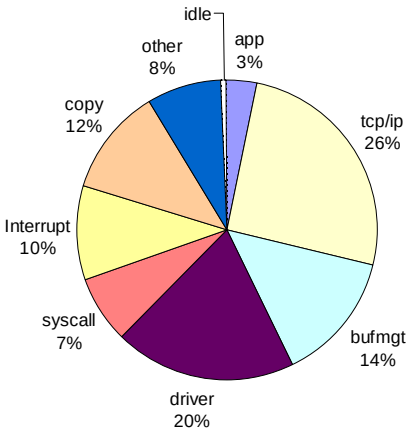

Effect of Threads/Copy on RX
Gigabits/s
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
256
512
1024
4096
65536
Receive Size in bytes
2P SMP
ETA ST
ETA HT
ETA HT NoCopy

Performance Analysis

- Look at one datapoint
  - 1KB Transmit case (Single-threaded)
  - Compare SMP to ETA
- Profile using VTune™
  - Statistical sampling using instruction and cycle count events

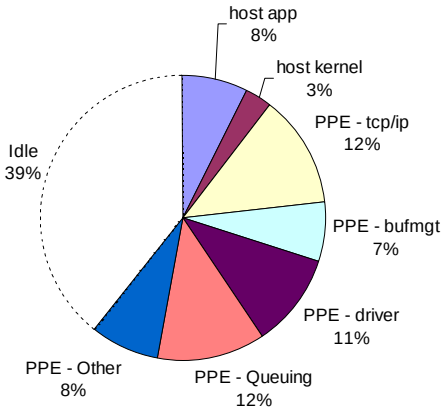


2P SMP Profile



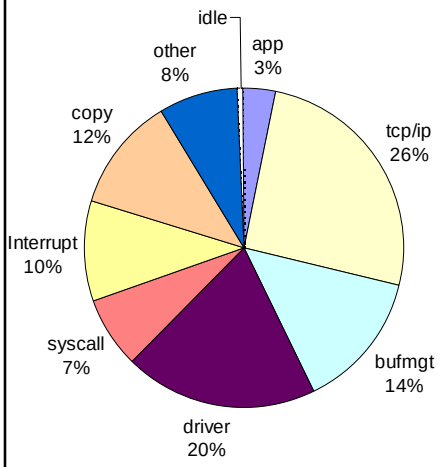
- Processing requirements in multiple components
  - TCP/IP is the largest single component, but is small compared to total
  - The copy overhead is required to support legacy (synchronous) socket semantics
  - Interrupts and system calls are required in order to time-share the CPU resources

ETA Profile (1 host CPU + 1 PPE)

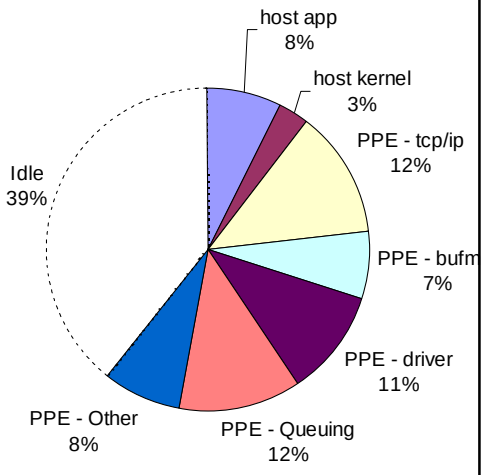


- Processing times are compressed
  - Idle time represents CPU resource that is usable for applications
  - Asynchronous queuing interface avoids copy overhead
  - Interrupts avoided by not time-sharing CPU
  - System calls avoided by ETA queuing model

Profile Comparisons

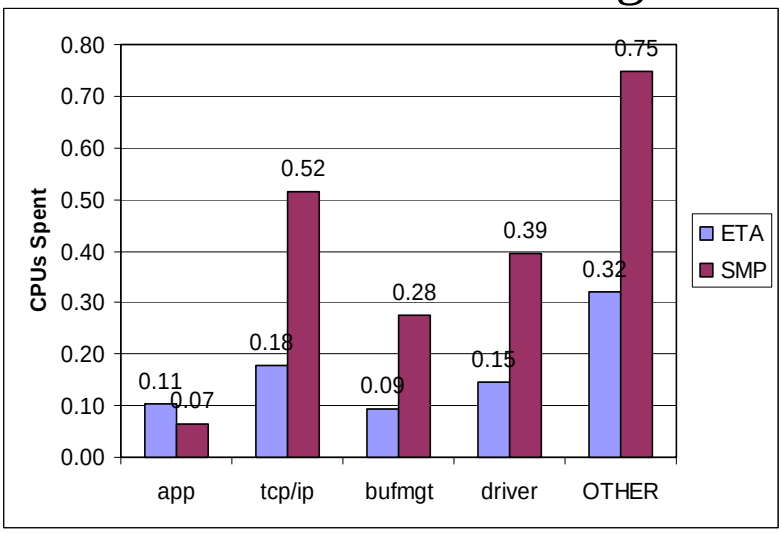


2P SMP



ETA

Normalized CPU Usage



Intel Research & Development

Normalized to SMP rate

Analysis

- Partitioning the system in ETA allows us to optimize the PPE in ways that are not possible when sharing the CPU with applications and the OS.
  - No kernel scheduling, NIC Interrupts not needed to preemptively schedule the driver and kernel
  - ETA optimized driver processing < half of SMP version by avoiding device register accesses (interrupt handling) and by doing educated pre-fetches
  - Copies are avoided by queuing transmit requests and asynchronously reaping completions… (Asynch. IO is important)
  - System calls are avoided because we're cheating ☺ (running the test in the kernel) but… we expect the same result at user level given user-level queuing and an asynchronous sockets API

Intel Research & Development

Analysis 2

- ETA TCP/IP processing component is < half of SMP version
  - Some path length reduction (explicit scheduling, locking)
  - Efficiencies gained from not being scheduled by the OS and interrupted by the NIC device, giving us better CPU pipeline and cache behavior
- Further Analysis
  - Based on new reference TCP/IP stack optimized for the ETA environment (in development)

Futures

- Optimized Packet Processing Engine Stack
  - Tuned to the ETA environment
  - Greater concurrency to hide memory access latencies
- Analysis
  - Connection acceleration
  - End-end latency measurement and analysis
  - iSCSI on ETA measurement and analysis
- Legacy Sockets Stack on ETA
  - Legacy application enabling
- Asynchronous sockets on ETA
  - Future apps

Summary

- Partitioning of processing resources ala ETA can greatly improve networking performance
  - General purpose CPUs can be used more efficiently for packet processing
- An asynchronous queuing model for efficient Host / PPE communication is important
  - Lessons learned in VI Architecture and IBA can be applied to streams and sockets

Acknowledgements

- Dave Minturn
- Annie Foong
- Gary McAlpine
- Vikram Saletore

Thank You.