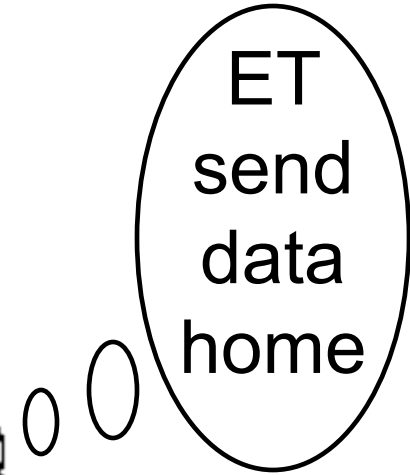

Event Transfer (ET) System



Carl Timmer

What is the ET system?

- Software to move events (data buffers) from process to process
 - One main ET process
 - Stores all events (buffers)
 - Distributes events to users
 - Monitors users
 - Library for users
 - Modify existing ET systems
 - Can even become an ET system
 - Get & put events from ET systems
 - Monitoring programs – text & graphical

ET System Design

- FAST

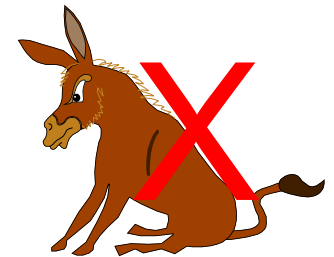
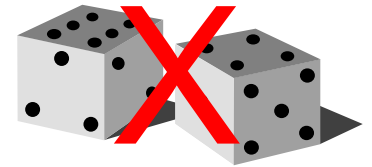
- Shared memory
- Fully multithreaded with POSIX pthreads
- Written in C on Solaris, Linux, & Mac OS X

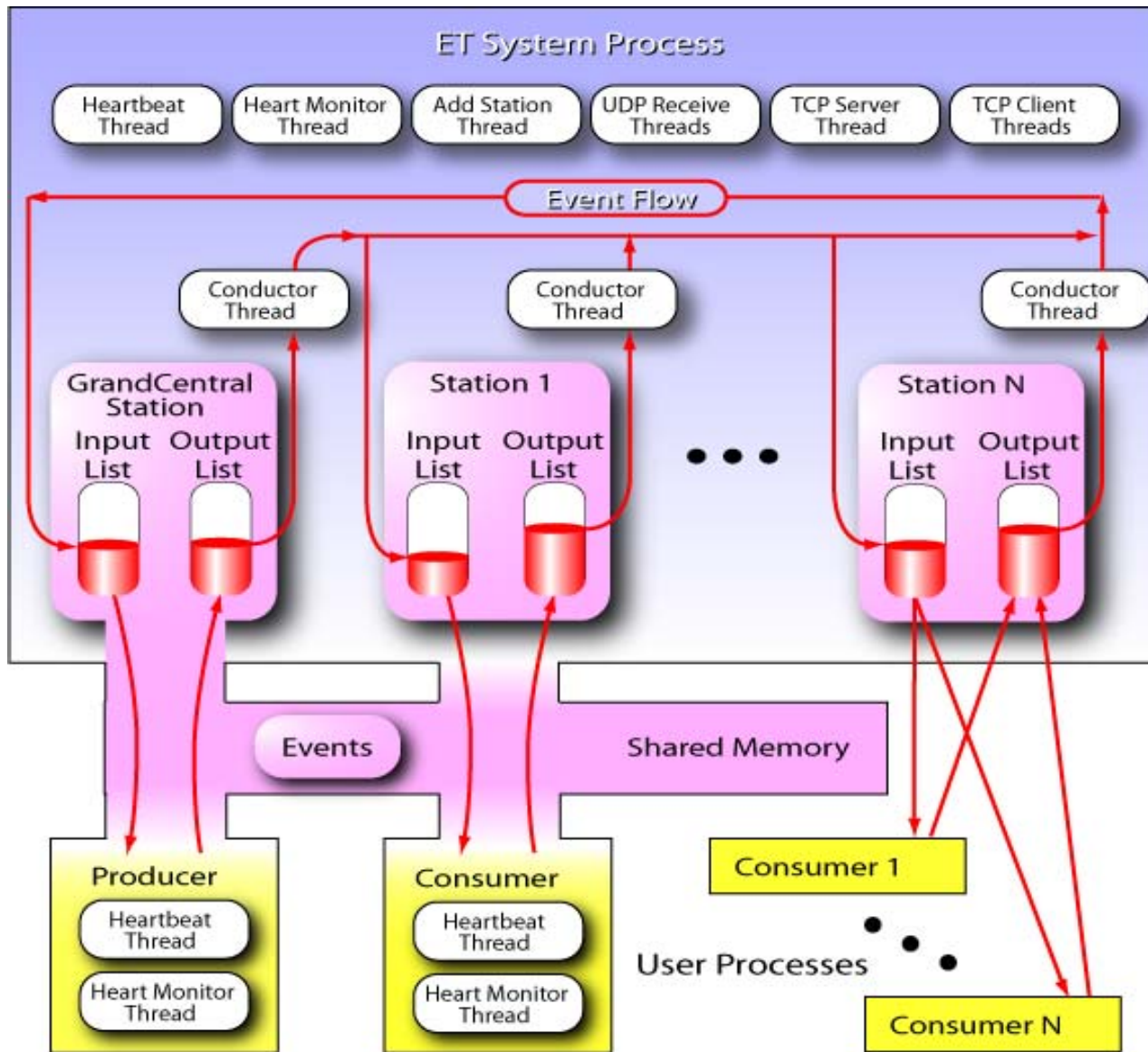
- RELIABLE

- Complete error recovery for system & users
- Data from crashed users are recoverable

- FRIENDLY

- Threads transparent to users
- Network transparent to users
- Java version available
- Reentrant – run as many copies on 1 computer as you like





Event Features

- Each event has integer array as metadata
- Events can be any size (within computer's memory limits)
- Keeps track of endian changes
- Users can get and put events in arrays. This increases performance over 10X.
- Data in CODA format can be swapped
- 2 levels of priority

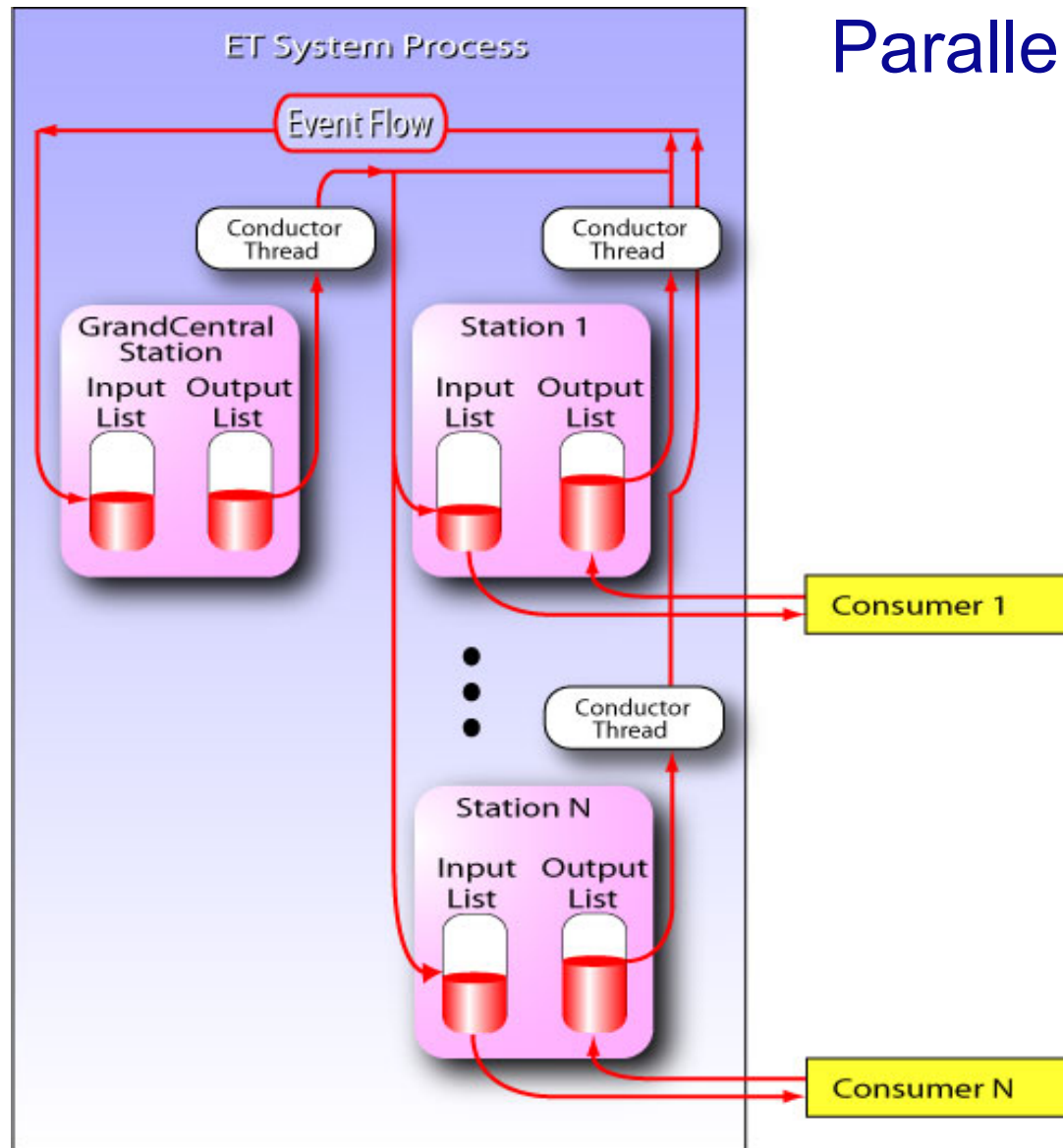
Station Features

- Users can define and add their own stations
- Multiple users can attach to the same station
- Each station can have a user-defined function to select events of interest
- Station order can be changed

NEW FEATURE

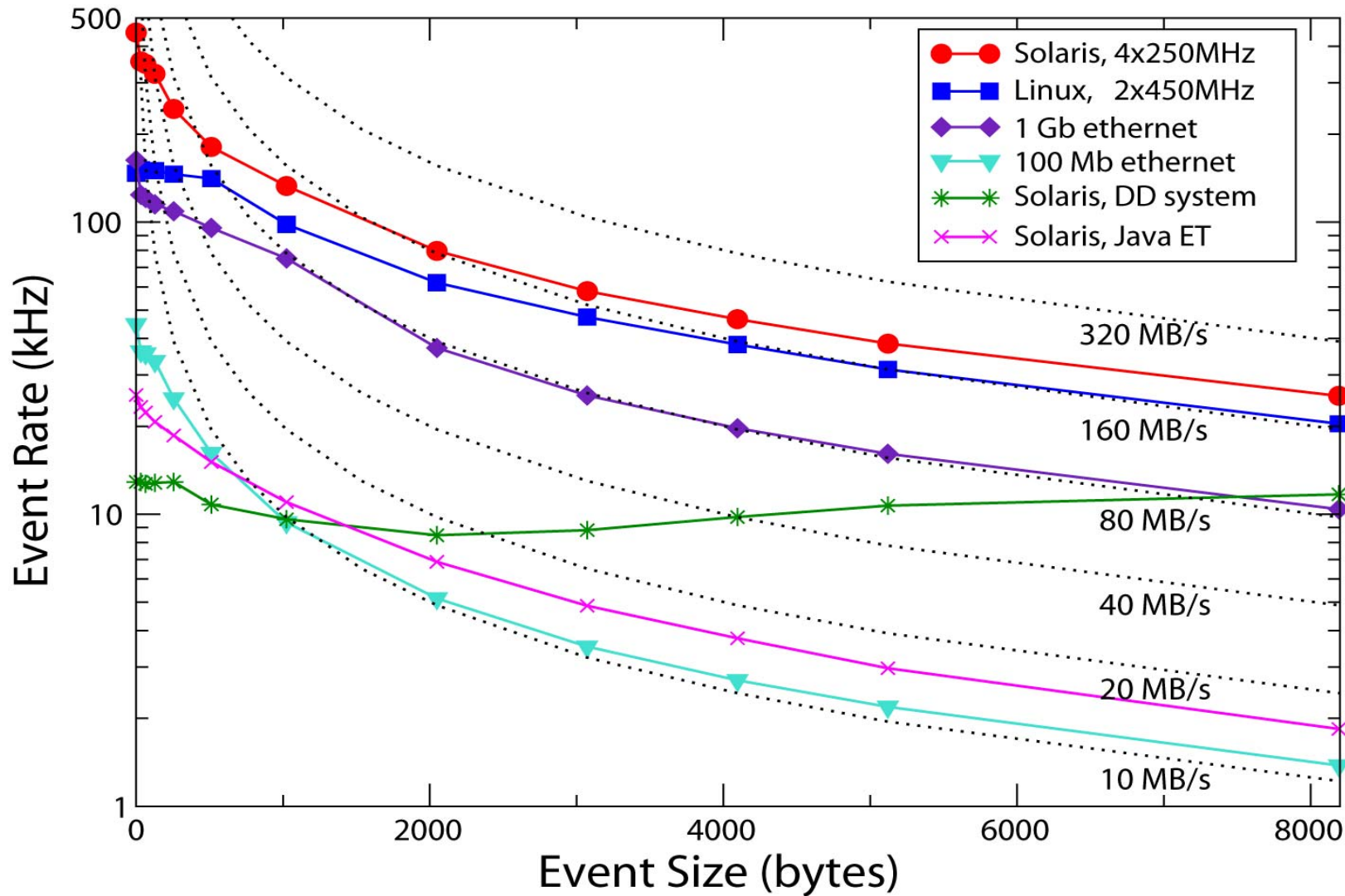
- Stations can be added in parallel with round-robin or load-balancing event distribution algorithms

Parallel Stations

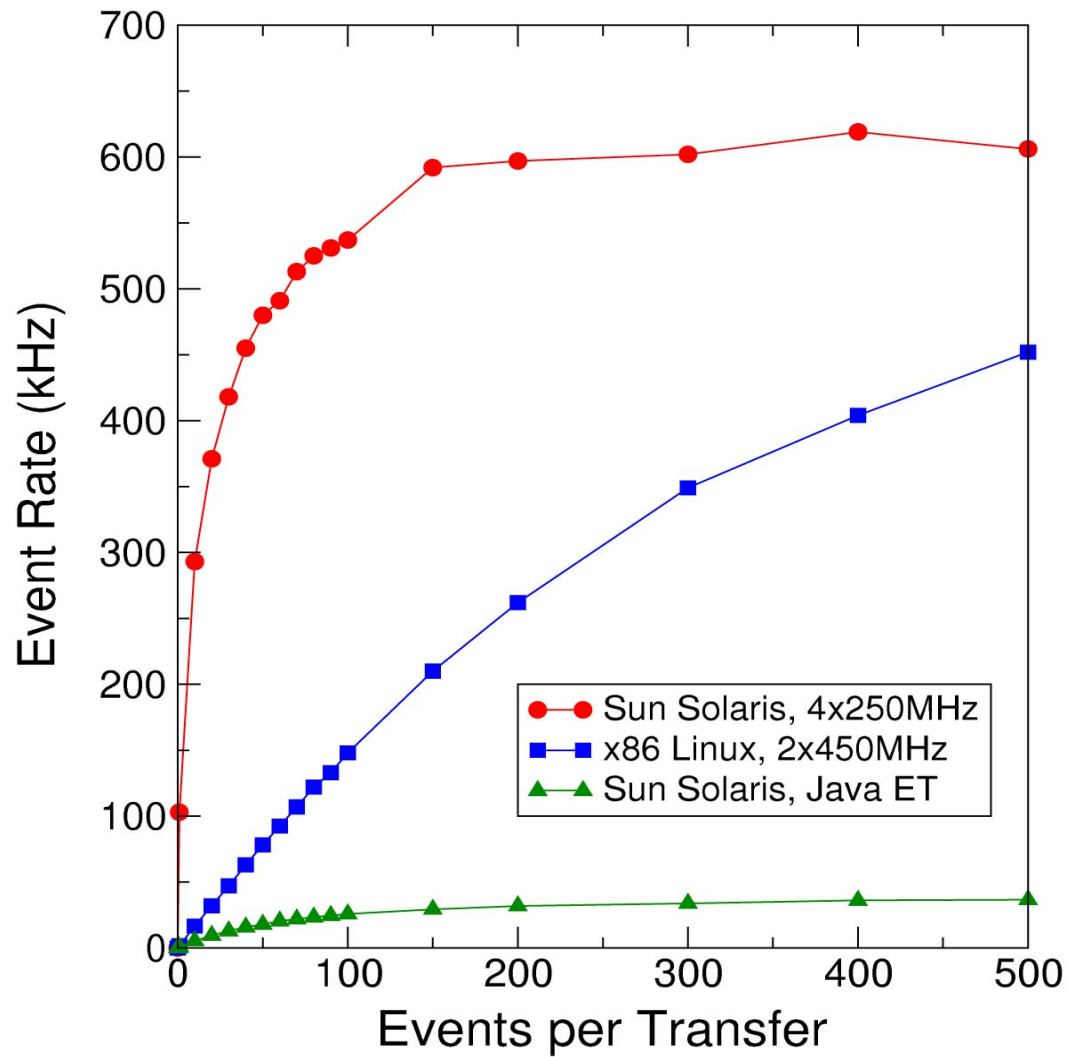


What about performance?

ET System Performance



ET System Performance



ET graphical Monitor

The screenshot displays the ET System Monitor interface. The left pane shows a tree view of system information:

- System**
 - Static Info
 - Host = aslan.jlab.org, language = Java
 - Ports: tcp = 11111, udp = 11111, mcast = 11112
 - Events: total = 3000, size = 32 bytes, temps = 0
 - Max #: stations = 20, attachments = 50
 - Network interfaces: 129.57.14.45
 - Multicast addresses: 239.200.0.0
 - Dynamic Info
 - Event rate = 72274 Hz
 - Events owned by: sys (2896), atts 1(0), 2(0), 4(0), 0(78)
 - Idle stations: Station3,
 - All stations: GRAND_CENTRAL, Station1, Station2, Station3
 - Stations = 3, attachments = 4, temp events = 0
- Stations**
 - GRAND_CENTRAL
 - Configuration
 - Status
 - Active, attachments: total = 1, ids = 0
 - Input events: 2800, total = 7606200
 - Output events: 0, total = 7606300
 - Station1
 - Station2
 - Station3
- Attachments**
 - 1
 - 2
 - 4
 - 0

The right pane shows a network diagram with the following components and connections:

- GRAND_CENTRAL** (cyan box) with 'In' and 'Out' ports. A red bar is visible in the 'In' port area. It is connected to attachment 0 (purple box).
- Station1** (cyan box) with 'In' and 'Out' ports. It is connected to attachments 1 and 4 (purple boxes).
- Station2** (cyan box) with 'In' and 'Out' ports. It is connected to attachment 2 (purple box).
- Station3** (red box) with 'In' and 'Out' ports.
- Connections: GRAND_CENTRAL connects to Station1 and Station3. Station1 connects to Station2. Station2 connects to Station3. Station3 connects back to GRAND_CENTRAL.

Future Developments

- Rewrite roc, event builder, & event recorder to use ET as communication mechanism
- Run speed tests with and optimize for Gigabit ethernet and TCP Jumbo Frames
- Fix currently broken ET web manual at <http://coda.jlab.org>
- Bug fixes