Exact Capture Ver 1.0

Lossless Packet Capture for ExaNICs

Exact Capture is a high-rate, lossless packet capture solution for ExaNIC network adapters. The system is fully open source and designed for performance as well as ease of configuration. It can be used with any ExaNIC network card, and is optimised for use with ExaDisk high speed flash drives. The system can be deployed on any suitably powerful server system.

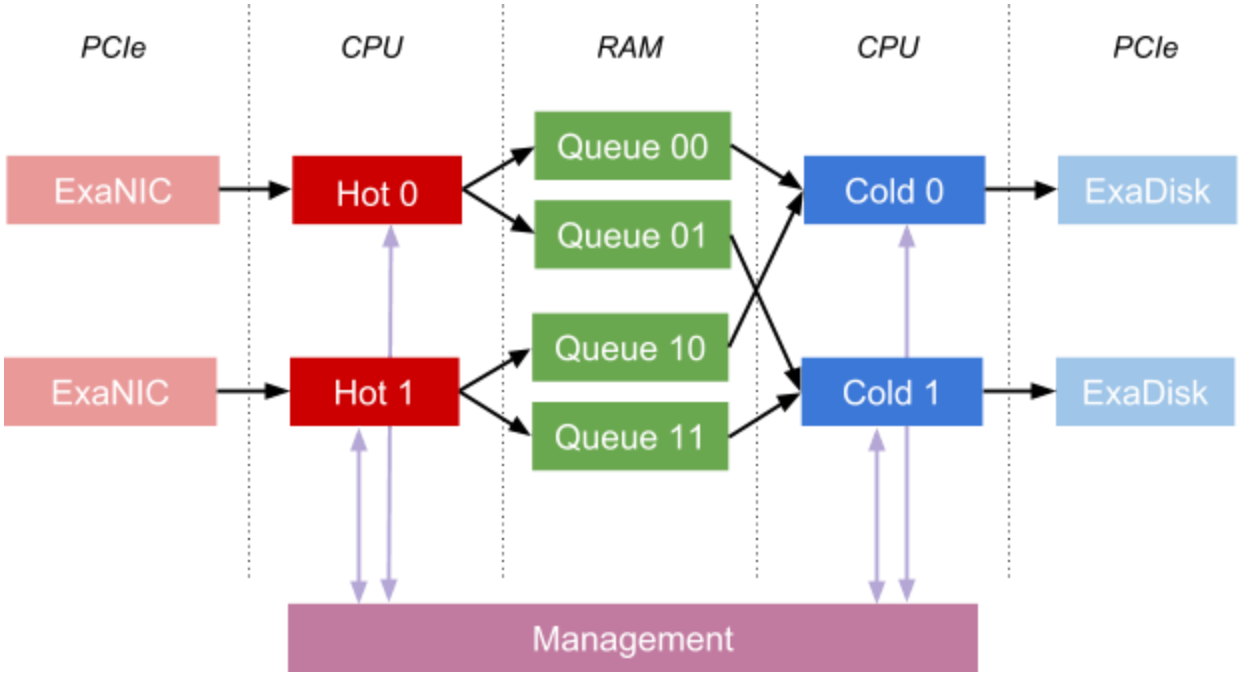
This document covers full details on the hardware and configuration requirements for Exact Capture as well as building, operating, tuning and debugging instructions.

Exact Capture Architecture Essentials - Hot, cold, and in-between

The Exact Capture software system comprises 4 major internal components:

1) One or more “hot” threads - to read packets from the ExaNIC(s) into memory
2) One or more “cold” threads - to write packets from memory to disk(s)
3) One or more shared memory queues - to connect the hot and cold threads to each other
4) One management thread - responsible for control and statistics collection / reporting

These basic architectural components are illustrated below, with the addition of the ExaNIC and ExaDisk resources. The head of each column highlights the performance limiting resource for that component:



Minimum Hardware Requirements - More is More

You are free to run Exact Capture on any system hardware. The following table describes our recommendations for the minimum system hardware requirements. For general guidance, we have successfully run the system on suitably configured Dell R230 and R730 machines. Any of the Dell R*30 and R*40 machines are likely to be excellent candidates.

CPU	**Speed** The minimum required CPU speed depends on the maximum capture rate required. For the purposes of this document, we assume that 10G line rate, at minimum sized (64B) frames is the capture rate requirement (i.e. approx. 14 million packets per second). At this rate, we have found that 3Ghz+ CPUs are sufficient. **Core Count** The number of CPU cores required depends on: 1. The number of interfaces that you wish to capture on 2. The type and speed of the disk drives 3. The maximum capture rate you need to sustain Assuming that ExaDisks are the target drives, each drive slice is capable of writing at a sustained rate of 10Gb/s. As an example, a minimal 10Gb/s Exact Capture installation will require 3 CPU cores. One core for a (hot) listener thread, one core for a (cold) disk writer thread and one management core. The management core is low priority and can be safely shared with other general purpose system cores. The listener thread should not be shared with any other process (i.e. be sure to make use of the isolcpus kernel parameter). In general, for n line-rate 10G ports, the system requires 2n + 1 CPU cores. e.g a 4x10G capture system will require 9 cores in total. *NB: CPU core counts are based on actual cores, rather than hyperthreads. In general, we recommend disabling hyperthreads on CPUs that support them.* **Recommendation** We have had good results with Intel Xeon E5-26xx range CPUs with a 3Ghz+ clock speed. For example the Intel Xeon E5-2643.
RAM	RAM usage will vary based on the number of NICs and disks that you are using. By default, each memory queue is organised into 256x 2MB slots for a total memory usage of approximately 512MB per queue. The total number of memory queues the product of the number of hot (ExaNIC) and cold (ExaDisk) threads. For a minimal 10Gb/s capture solution, with a single ExaNIC and

	ExaDisk, only 1 memory queue is required for a total of approximately 512MB of memory. For 4x10Gbs system, with 4 disks, 4x4 = 16 queues will be required, for a minimum memory usage of ~8GB. **Recommendation** We recommend at least 16GB of DDR IV RAM in your machine.
PCIe	For sustained, minimum sized packet capture, each 10Gb/s ExaNIC interface requires approximately 4x PCIe Gen 3 lanes. **The hot threads must run on the CPU socket directly connected to these PCIe lanes.** For sustained high performance writing, each ExaDisk interface requires 2x PCIe Gen 3 lanes. **The cold threads must run on the CPU socket directly connected to these PCIe lanes.**
ExaNIC	All ExaNIC network cards will work with Exact Capture. Following is a summary of the features, requirements and limitations of each card: - **ExaNIC X2/X4** (2/4x 10GbE) - these cards only run PCIe Gen 2x8. This limits the device to 1 port of line-rate capture for small packet sizes. For larger (average) packet sizes (e.g. 512B+) both ports will be supported. Timestamp resolution is 6.2ns. - **ExaNIC X10 / GM** (2x 10GbE) - these cards can be used without restriction on suitable PCIe Gen 3x8 slots. Timestamp resolution is 6.2ns. - **ExaNIC HPT** (2x 10GbE) - these cards can be used without restriction on suitable PCIe Gen 3x8 slots. Timestamp resolution is 0.25ns (250ps) - **ExaNIC X40 / VXP** (8x 10GbE) - Only 2 ports can be used at line rate for all packet sizes. Up to 4 ports can be used at larger (average) packet sizes (e.g. 512B+). Timestamp resolution is 6.2ns. - **ExaNIC X40** (2x 40GbE) - Speeds up to 20Gb/s are likely to work out of the box on any single interface (though this is untested). Load balancing/packet spraying across multiple receive rings is also likely to assist line rate capture, though this is feature is not (yet) implemented.
Disk Drives	Exact Capture is tested and optimized to run on ExaDisk FX14/FX18 drives. Each ExaDisk is capable of running 40Gb/s sustained write speed to disk in a PCIe Gen 3x 8 slot. The drives are currently available in 4TB and 8TB capacities. The system will (in principle) operate with any disks. High speed flash drives, especially NVMe disks are highly recommended to keep the number of threads and memory usage down. For slower disks (e.g. SATA based flash disks) sharing CPU cores for writer threads is likely to reduce the CPU core count requirements without affecting overall performance. This is untested.

Building and Installing Exact Capture

Source Code and Licensing

Exact Capture is available from github.com as an open source project. If you would like to discuss alternative licensing schemes, please contact the Exablaze sales team.

Software Requirements

To build the software, you will need a recent C compiler supporting C99 or higher, and to have installed the ExaNIC software libraries (also available from github.com).

Building

There are 3 build options for Exact Capture:

1) Performance build
2) Error assertions build
3) Debug build

By default, Exact Capture is built in performance mode. In performance mode, unnecessary internal error checking is disabled. For example, bounds checks on memory access. To build Exact Capture in performance mode, simply run “make” in the top level.

To build a version with stricter internal error checking assertions, run “make assert”. This version is still capable of operating at 10Gb/s on many systems, though will suffer marginal performance degradation, especially on slower CPUs.

To build a debug version, run “make debug”. The debug build applies stricter warning checking requirements at build time, and enables detailed debug tracing throughout the application. This version is unlikely to keep up at high-rate.

(un)Installation

To install Exact Capture, run “make install” as the root user. To uninstall, run “make uninstall” as the root user.

System Configuration

For detailed instructions on configuring systems for high performance, please see our benchmarking guide. https://exablaze.com/docs/exanic/user-guide/benchmarking

Operation

Quick Start

To run Exact Capture on a single 10GbE interface, writing to a single disk slice, the following invocation is sufficient:

```
$ exact-capture --input=exanic0:0 --output=/data0/ --cpus=0:1:2
```

Note 1: Canonical Linux interface names such as “eth1” or “enp0s1” can be freely used in place of Exablaze ExaNIC device names (e.g. “exanic0:0”). The interface must however be an ExaNIC.
Note 2: The CPU specification is a colon (“:”) separated list containing the management CPU, the ExaNIC listener thread CPU(s), and the ExaDisk writer thread CPU(s). It is assumed that CPU cores have been isolated according to the System Configuration instructions above.

To run exact capture on a pair of 10GbE interfaces, writing to a three disk slices, then the following invocation would be necessary:

```
$ exact-capture --input=exanic0:0 --input=exanic0:1 --output=/data0/
--output=/data1/ --output=/data2/ --cpus=0:1,2:3,4,5
```

Complete Command Line Parameters

The Exact Capture application supports the following command line parameters.

Short Form	**Long form**	**Default / Required**	**Description**
-i	--input	required	The ExaNIC interface(s) to capture on
-o	--output	required	The destination directory and filename stub to output to. Filenames will be output in the following format /output/dir/base_xx.expcap. Where xx is a unique file index. For details on the expcap format please see the Exact Capture Output Format (expcap) section later in this document.
-c	--cpus	required	The list of CPUs to assign threads to for management, listening and writing threads. This is specified in the the following format: m:ls:ws

			Where m is the core number for management, and ls/ws are comma separated lists of listener and writer CPU core numbers. For example --cpus=5:2,3:7,6,1 would configure Exact Capture to run the management thread on CPU 5, 2 listener (hot) threads on on CPUs 2 and 3 respectively, and 3 (or more) writer threads on cores 1,6 & 7 respectively. Note: the number of listener CPUs must be exactly equal to the number of ExaNIC --interfaces in use. Furthermore listener threads cannot share CPUs with management or writer threads. If there are fewer writer threads than --outputs, writer threads will be reused.
-s	--snaplen	2048B	In some cases it is not necessary / useful to capture the entire packet. Set the snap length to determine the maximum size of packet that can be captured. This value cannot be 0 or less.
-m	--maxfile	(0) unlimited	High rate capture can produce very large file sizes. To reduce the file sizes, Exact Capture can cap the file size to a maximum, and will start a new file each time it is reached. A value of 0 or less puts no limit on the output file size.
-l	--logfile	none	Exact capture can optionally write log messages to a log file specified.
-t	--log-report-int	1.0 sec	This sets the statistics calculation and logging interval in seconds.
-v	--verbose	flag	Enabling verbose mode will produce 2 output log lines every log interval (see above). These log lines will include summary statistics of the performance of all listener threads and all writer threads.
-V	--more-verbose	flag	Enabling more verbose mode will produce 1 output log line for every listener and writer thread. Each log line will include per-thread statistics counters/statistics. This can be combined with -v mode above.
-d	--debug-logging	flag	Debug logging mode enables display of the full file path, process ID and thread ID in each output log line. This is useful to track where a given log message originated.

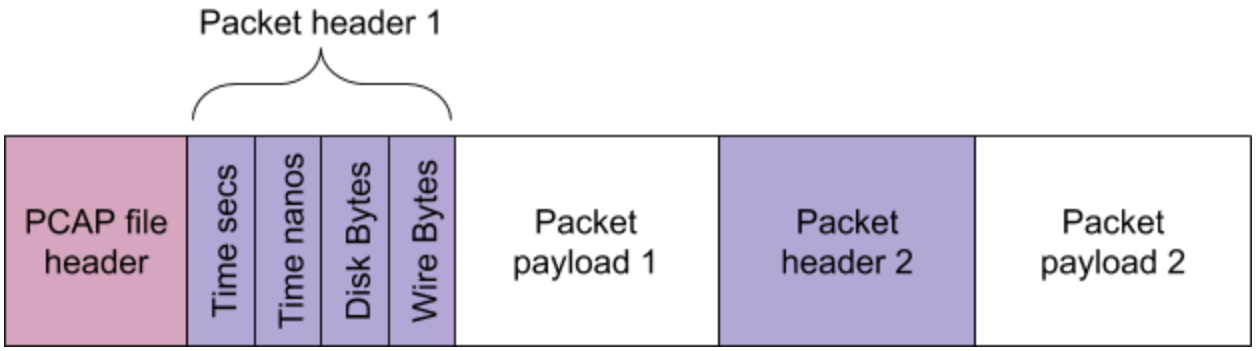
-T	--no-log-ts	flag	By default, logs include a timestamp. This can make the output overly verbose. Use this flag to disable timestamps.
-w	--no-warn-overflow	flag	Software overflows will produce a warning. This may be problematic if the system is underperforming and these happen often. The flag disables these warnings.
-n	--no-spin	flag	By default Exact Capture outputs a progress “spinner” to the console. This flag disables it.
-p	--perf-test	0	Exact Capture supports several performance testing modes. These can be used to give a sense of the best possible performance that you can expect from your system configuration. The modes are as follows: 0. No performance testing 1. Replace all ExaNIC interfaces with a dummy interface. ExaNICs are no longer a performance limitation. This tests the maximum possible receive rate that your system can achieve for 64B frames. **Note that 10GbE line-rate with 64B frames is about 7Gb/s** (due to Ethernet interfame gap overheads)**.** 2. Replace the internal memory queue with a dummy interface on both sides. This tests the maximum performance possible when when system memory is not the bottleneck. This is also a good test of disk writing speed (for minimum sized packets). 3. Replace the ExaDisk interface with a dummy. This tests the performance through the system when disk writing speed is not a limitation. This may be helpful to debug cases where your disks are not configured/performing correctly. 4. Replace both the ExaNIC and the internal memory ring with dummies. Can be used to measure the absolute best performance possible when NICs and memory are not the limitations. Can also help to find interference bugs between ExaNICs and ExaDisks sharing limited PCIe bandwidth. 5. Replace the ExaNIC and ExaDisk with dummies. This is useful for testing the

			maximum achievable application throughput, including through system memory, but excluding reading from and writing to real hardware. 6. Replace the memory queues and ExaDisk with dummies. Can be help to find interference bugs between ExaNICs and ExaDisks sharing limited PCIe bandwidth. 7. Replace the ExaNIC, memory queue and ExaDisk interfaces with dummies. Useful for determining the overheads within the application (i.e. CPU speed) issues.

Exact Capture Output Format (expcap)

Exact Capture outputs packet captures to a modified pcap format called expcap. The expcap format is a backwards compatible extension to standard pcap format. A number of tools and utilities are included for operating on these files and converting them into standard pcap format if necessary.

For reference, standard pcap files are formatted as follows (more details can be found on the Wireshark website):

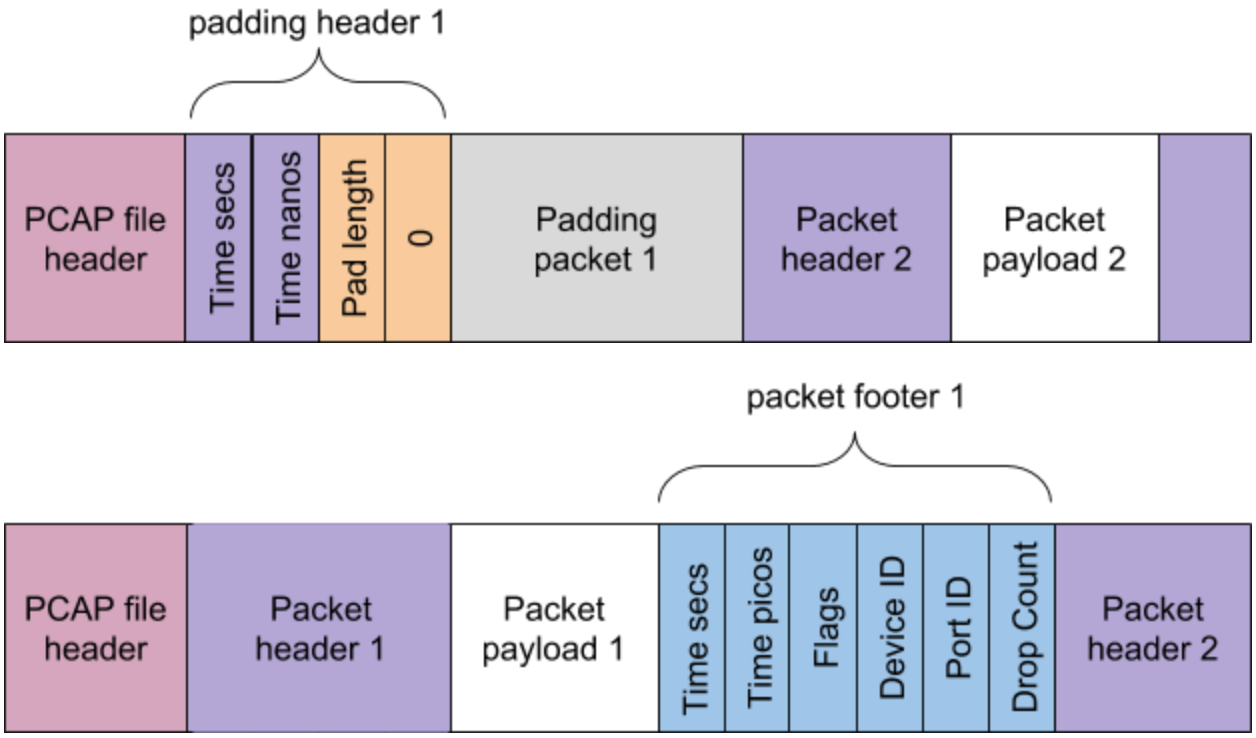


The expcap format extends this format in two ways:

1. Padding Packets

As a performance optimisation, Exact Capture occasionally needs to insert padding packets (e.g. to align to a 4k boundary) into the output file. These packets are easily recognizable because pcap header wire length field (as above) is set to 0, with the on disk length field (again as above) is set to the size of the padding.

Although setting these header fields precisely captures the semantics of the padding packets (i.e bytes written to disk that were never found on the wire), this is technically a pcap file specification violation. Presumably the writers never envisaged this kind of use case. Nevertheless, standard tools like Wireshark operate correctly and ignore these packets as they should. The following figure depicts a padding packet directly after the file header. This will be found in all Exact Capture output files.



2. Packet Footers

Each captured packet is extended with a packet footer. This footer contains a variety of extra fields, not available in the standard pcap format. When the footer is added, the standard pcap disk bytes field is updated to reflect the extra length on disk. Once again, this means that the byes on disk value may exceed the bytes on the wire value (though not always. e.g. when a snaplength is set). The addition of a footer adds bytes to the disk that were never found on the wire is again, technically a PCAP specification violation. However, once again, standard pcap processing tools like Wireshark operate correctly and ignore these extra bytes as they should. The above figure shows a representation of expcap packet footers added to the first packet.

The additional expcap footer fields are described in detail in the table below. They borrow the spirit of some of the fields found in the ERF format.

Field	Width (bits)	Description

Time seconds	32	Time in seconds since the epoch
Time picoseconds	40	Time in picoseconds since the last second boundary
Flags	8	The following flags bits are currently supported: 0. New CRC Calculated (The CRC field contains a new new value including the footer) 1. Frame aborted - this frame was aborted on the wire by the sender. 2. Frame corrupt - the hardware CRC checker detected an error with this frame. 3. Frame truncated - this packet was longer than the snap length and has been truncated.
Device Number	8	The ID of the device that captured these packets. For example, when capturing on the exanic3:7 interface, the device number would be 3.
Port Number	8	The port on the device that was used to capture the packet. For example, capturing on exanic3:7 interface, the port number would be 7.
Dropped count / CRC top 16 bits.	16	If the new CRC flag is **not** set, contains the number of packets dropped between this packet and the previous packet. Otherwise this is the top 16 bits of the new CRC.
CRC bottom 16	16	If the new CRC flag is set, contains the bottom 16 bits of the new CRC. Otherwise, unused.

Tools and Utilities

Exact Capture is supplied with a small collection of tools for operating on expcap files. These are:

- **exact-pcap-extract** - This tool has two purposes. Firstly, it is used to extract all packets associated with a given port and device number from a collection of expcap files. Secondly, it can be used to convert from expcap format into standard pcap format.
- **exact-pcap-parse** - This tool is useful for creating ASCII text dumps of pcap and expcap files and for working with picosecond timestamps.
- **exact-pcap-match** - A common use case of is for Exact Capture is latency calculations. This tool can be used to match identical frames from two pcap or expcap files and calculate the latency between them. It can be easily extended to support matching frames that are not identical (e.g. tick-to-trade latency calculations). (edited)