

HyperScan

A High-Performance Regular Expression Matching Engine



"HyperScan's runtime is engineered for high-performance from the ground up."

EXECUTIVE SUMMARY

HyperScan is a software pattern matching library that can match large groups of regular expressions against blocks or streams of data. Ideal for applications that need to scan large amounts of data at high speed, such as Intrusion Prevention (IPS), Antivirus (AV), Unified Threat Management (UTM) and Deep Packet Inspection (DPI) systems, HyperScan runs entirely in software and is deployed on a wide range of Intel® processors and operating systems.

Design

HyperScan delivers the highest scanning performance, with all configurations done through function calls by the calling application. It can be broadly split up into 'compiler' and 'runtime' components.

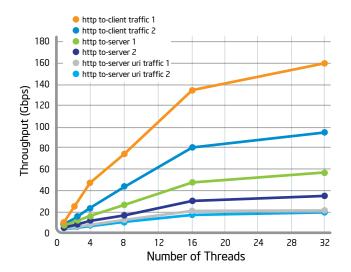
Compiler

- Patterns for HyperScan are specified in the industry standard PCRE format.
- A set of patterns is compiled into a fixed database, targeted for a particular mode of operation (such as streaming mode or block mode) and a particular target platform (e.g., Intel® Atom™ processor or Intel® Xeon® processor).
- A HyperScan database can be used inmemory after compilation, or serialized and shipped off to another system for use.

Runtime

The HyperScan runtime presents a synchronous API that matches data against a compiled pattern database, returning each match discovered to the application via a callback. HyperScan's runtime is engineered for high performance from the ground up:

- Each architecture port takes advantage of advanced processor features (such as SIMD instruction sets) where available.
- Pattern databases are read-only at runtime, enabling efficient use of a single database by multiple concurrent threads or processes.
- Not a backtracking engine: memory usage at runtime is fixed and small, making it appropriate for resourceconstrained environments.
- Software-only, synchronous approach enables good performance even in scenarios with small writes and high match rates.



Streaming Operation

HyperScan's simplest use-case is a block scanning application. Such an application merely wants to scan a single contiguous block of data with a set of regular expressions and collect any matches that occur. For these cases, HyperScan provides a block mode interface which stores no state and returns all of the matches before it completes.

Many applications operate on data that may not be available as a single block. For example, network traffic scanning applications often cannot hold in memory all of the packets that make up a message, and simply scanning each packet ignores matches that straddle packet boundaries. To support those cases, HyperScan also provides a streaming API, enabling such applications to easily implement crosspacket inspection.

In streaming mode, the application can pass HyperScan a stream of data blocks, one at a time, and HyperScan will return matches as they occur, even matches that cross the boundaries between these

blocks. Streaming support is a firstclass citizen¹ for HyperScan: matching is supported across an arbitrary number of block writes, and the full complement of supported PCRE constructs can be used.

Streaming operation requires a small fixed-size stream record to store the state associated with each stream, and HyperScan provides an easy-to-use set of interfaces for manipulating these records.

Scalability

To take advantage of all of the resources available on modern processors, an application must be able to operate on multiple cores simultaneously. HyperScan is often deployed in scenarios where many streams of data are being scanned against the same pattern database, and it has been designed to operate as efficiently as possible in this configuration.

These performance results were generated on an Intel® Xeon® Processor E5-2600 product family platform with 16 cores, using a variety of signature sets and traffic mixes.

A Software Library

HyperScan is a single library with a rich C API, making it straightforward to integrate into existing applications without the need to manage additional processes or configuration files.

- The compilation of patterns into a HyperScan database is done ahead of time, and support for database serialization and cross-compilation means that this can be done on a different host, even on a different platform.
- The runtime matching API is synchronous, with matches delivered to the application via callbacks. This allows for far simpler integration than the asynchronous model used by many hardware accelerators, and provides greater control over matching behavior.
- HyperScan does not impose a threading or process model: instead, it provides the primitives necessary to support concurrent contexts, and leaves control of how that concurrency is managed to the application.

PCRE Syntax

The regular expressions accepted by HyperScan conform to a large subset of the industry standard Perl-compatible Regular Expressions (PCRE) syntax. Some constructs, such as the use of back-references, are unsupported due to their incompatibility with efficient streaming operation, but most PCRE behavior is implemented and available for use.

Wide Platform Support

HyperScan has been ported to a variety of different platforms and provides complete functionality across all of them. Each port makes aggressive use of the unique features available on that processor, in order to maximize performance. This allows a vendor to deploy a single patternmatching engine across an entire range of devices, from small consumer devices to high-end appliances for the enterprise.

Performance

HyperScan has been designed from the ground up for high performance and low latency, properties that are both critical for pattern matching in modern network security applications. HyperScan delivers excellent performance by:

- Extensive analysis of the signature set at compilation time, building a pattern database tuned for efficient matching.
- Takes advantage of the unique features of the CPU architecture (such as SIMD instruction sets for Intel* processors) to build runtime engines for optimal performance.
- Only performing memory allocations where necessary (such as for fixed-size stream state records), and providing pluggable allocations for applications that need to control them.

Example

The following data was collected by benchmarking HyperScan on a dual-socket Intel Xeon processor E5-2600 product family platform (16 cores), using HTTP test traffic and a complete set of IPS signatures sourced from a leading (Tier-1) security equipment vendor.

The benchmarking application used for these tests passed captured HTTP traffic through HyperScan, recording the time spent actually matching traffic against the signature database. This data was scanned packet-by-packet, simulating the behavior of a real network application such as an IPS or web proxy appliance.

Data was matched in streaming mode for cases where the threats might be spread across multiple packets, and in block mode for treats that would be contained within a single block of data such as an URI (which would typically be normalized by the application before scanning.)

Supported Platforms

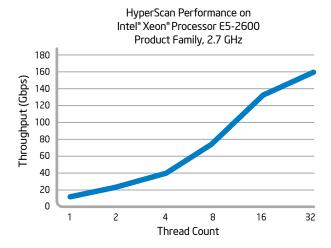
HyperScan is available across a variety of operating systems, and can be used in both user-space and kernel-space applications on some platforms.

Processor Architectures

Intel® architecture processors

Operating Systems

- Linux*
- FreeBSD*
- Wind River* Linux
- Wind River VxWorks*
- Windows*
- Mac* OS X*





¹ A first-class citizen (also object, entity, or value) is an entity that supports all the operations generally available to other entities. (Wikipedia)
Copyright © 2013 Intel Corporation. All rights reserved. Intel, the Intel logo and Xeon are trademarks of Intel Corporation in the United States and/or other countries.

*Other names and brands may be claimed as the property of others.
Printed in USA 1013/MS/SD/PDF ♣ Please Recycle 329711-001US

For more information about Intel processors, visit www.intel.com.