

»» Dynimizer

Instantly Improve MariaDB Performance By 30% With Just-In-Time Compiler Technology*

Dynimizer quickly optimizes the in-memory machine code of target processes based on profiling information it gathers at run-time, improving performance of target workloads. Dynimizer typically runs as a background system process and does not require the source code of target programs that it optimizes.

Key Benefits

- **Improved CPU performance**

IPC (instructions per cycle) improvements of up to 70%, resulting in greater performance for target workloads. Instruction cache misses, instruction TLB misses, and branch mispredictions may be reduced to provide the effect of having greater CPU resources.

- **Reduced CPU usage**

By increasing IPC, processors require fewer cycles to perform the same amount of work, thereby reducing energy consumption and improving performance per watt. Applications can now spend less time wasting CPU cycles, saving energy and freeing up CPU resources for other work.

- **Fully autonomous**

Dynimizer runs as a stand alone background process and automatically optimizes CPU intensive applications.

- **Highly configurable**

Dynimizer can be run autonomously or invoked directly on specific program instances or processes. Controls can be put in place to limit the applications it targets.

- **Zero downtime, instant ROI**

Dynimizer can be applied to off-the-shelf applications without making changes to them or their host systems. Running applications do not need to be restarted, and their source code is not required. Applications can be optimized in 60 seconds.

*Performance improvements may vary with workload and hardware

How does Dynimizer work?

- **A JIT Compiler**

Dynimizer is a Just-In-Time (JIT) compiler that profiles programs, and uses that profiling information to better rewrite those program's in-memory machine code for improved performance.

- **Exploiting Run-Time Information**

Because it optimizes machine code at run-time, Dynimizer has far more information about how the program is being used and its run-time environment than the original compiler that produced the program and shared libraries, allowing it to generate higher quality machine code.

- **Optimizes In-Memory Instructions**

Dynimizer does not modify in any way an optimized program's on-disk files, such as data, configuration, executable or shared library files.

- **Lightweight Daemon Process**

Dynimizer runs as a lightweight background process, and optimizes the in-memory instructions of other processes running on the same host OS, using the standard ptrace Linux system call to make changes to the processes being optimized.

- **Machine Code Specialization**

This machine code specialization is done each time a program is run, and can be done repeatedly throughout the lifetime of a program if its workload drastically changes from the last time it was optimized by Dynimizer.

- **A New Frontier For JIT Compilers**

Just-In-Time (JIT) compilers have been used for years to speedup managed run-times such as the Java Virtual Machine and .NET. They use as input a virtual machine code format (such as Java bytecode), and perform profile directed optimizations when transforming that into real machine code. Dynimizer does this while using real machine code as input instead.

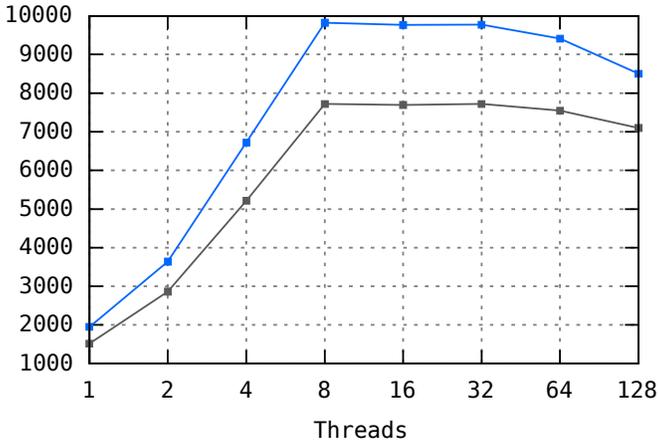


MySQL 5.7 + Sysbench Benchmarks

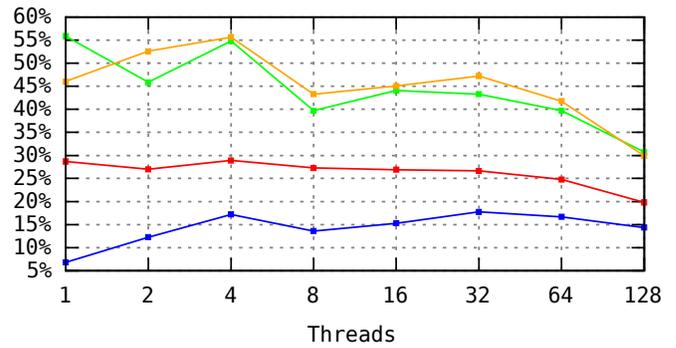
WITH Dynimizer —■—
 WITHOUT Dynimizer —■—

OLTP Read-Only —■—
 OLTP Simple —■—
 Select —■—
 Select Random Ranges —■—

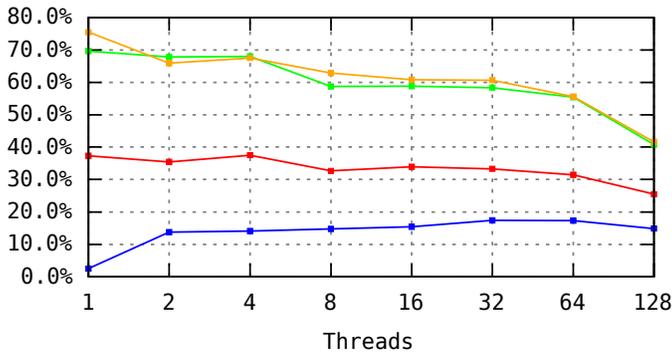
OLTP Read-Only Transactions Per Second



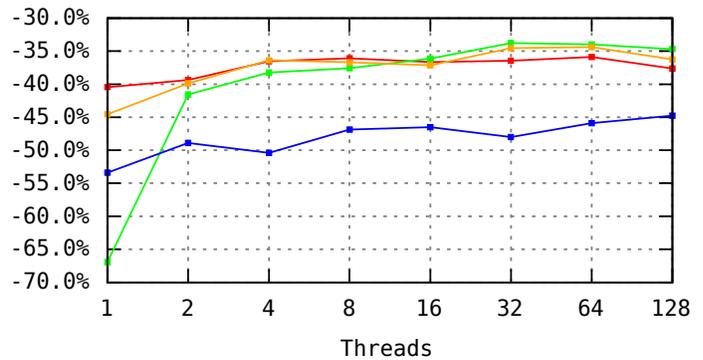
Increased Transactions Per Second With Dynimizer



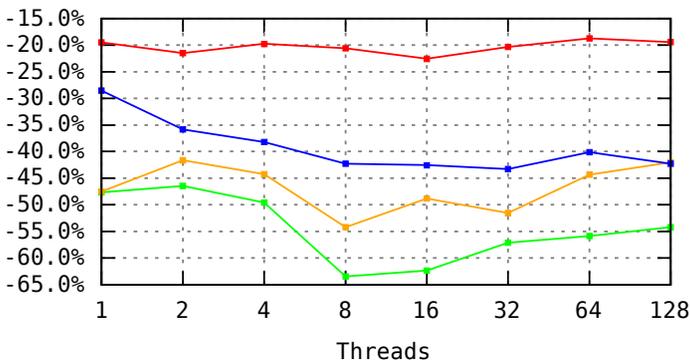
Increase In CPU Instructions Per Cycle With Dynimizer



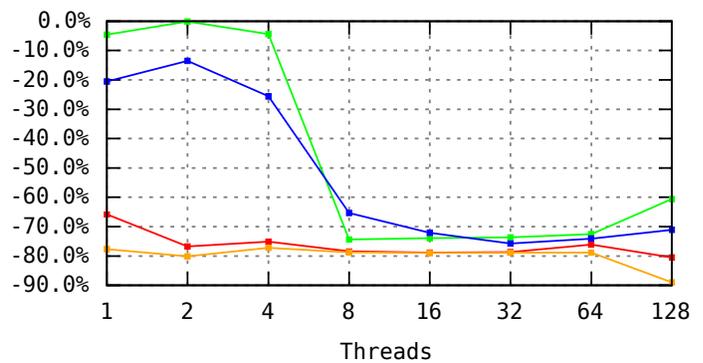
Decrease in CPU Instruction Cache Misses With Dynimizer



Decrease in CPU Branch Mispredictions With Dynimizer



Decrease in CPU Instruction TLB Misses With Dynimizer



CPU: Intel(R) Xeon(R) CPU E3-1270 v6 @ 3.80GHz, 4 cores, 8 Threads (Kaby Lake)

RAM: 32 GB of 2400 MHz DDR4

*InnoDB storage engine with working set in memory.

*Dedicated server rented from OVH, model: SP-32 Server, data center: BHS 5.

*Relative improvement similar across different table sizes or number of tables.

*Details on how to recreate runs and download raw results data can be found here:

<https://dynimize.com/blog/discussions/dynimizer-mysql-cross-microarchitecture-analysis/>

Dynimizer lifecycle

Dynimizer is first installed with the following command:

```
$ sudo bash -c 'bash <(wget -O - https://dynimize.com/install) -default'
```

Below are the stages Dynimizer goes through when optimizing programs:

