Jinx Performance

Introduction

Because Jinx is targeted at real-time environments such as videogames, it's important for developers to have a realistic assessment of Jinx's overall performance characteristics. This paper presents the results of a performance test that exercises a wide range of features in various threaded environments, and on several different machine types.

Performance-Related Features

Jinx features a number of features specifically designed to help improve performance in real-time applications.

Thread-Safe Scripting

Jinx is designed to safely execute scripts in arbitrary threads. Because scripts naturally execute as co-routines, there is a minimal dependency on global resources, except when accessing library-wide functionality, such as getting or setting a property. As such, typical scripts generally do not suffer from much thread contention, and scale well on multiple cores.

This ensures that your own code can use Jinx scripts in a threaded environment without the use of performance-killing global locks.

Built-In Allocator

Jinx utilizes its own block allocator designed to prioritize efficiency for small, frequent allocations, as is typical of scripting requirements. Additionally, it makes use of thread-local storage pools to ensure minimal contention between scripts executing independently on different threads.

Performance APIs

Jinx provides two API calls, `IRuntime::GetScriptPerformanceStats()` and `Jinx::GetMemoryStats()`, used for retrieving performance and memory stats respectively. This can help to provide runtime insights for both memory and CPU use to ensure Jinx stays within acceptable performance boundaries. You can see more details of these functions and the data they return in the online documentation.

Performance Tests

We conduct some synthetic benchmarks on three different machines, each running a different OS, in order to get a realistic idea of Jinx's performance characteristics. The performance test is included as part of the standard Jinx distribution, and is called *PerfTest*.

Machine One

- Type: 2009 Desktop PC
- CPU: 3.20GHz Intel Core i7 CPU 960 4 Cores
- OS: Windows 10

```
--- Performance (1 thread) ---
Total run time: 3.314056 seconds
Total script execution time: 3.244217 seconds
Number of scripts executed: 340000 (102593 per second)
Number of scripts completed: 40000 (12069 per second)
Number of instructions executed: 22880000 (6.90M per second)

--- Performance (2 threads) ---
Total run time: 1.752988 seconds
Total script execution time: 3.415886 seconds
Number of scripts executed: 340000 (193954 per second)
Number of scripts completed: 40000 (22818 per second)
Number of instructions executed: 22880000 (13.05M per second)

--- Performance (3 threads) ---
Total run time: 1.237908 seconds
Total script execution time: 3.583193 seconds
Number of scripts executed: 340000 (274656 per second)
Number of scripts completed: 40000 (32312 per second)
Number of instructions executed: 22880000 (18.48M per second)
```

```
--- Performance (4 threads) ---
Total run time: 1.020461 seconds
Total script execution time: 3.874900 seconds
Number of scripts executed: 340000 (333182 per second)
Number of scripts completed: 40000 (39197 per second)
Number of instructions executed: 22880000 (22.42M per second)

--- Performance (5 threads) ---
Total run time: 1.010381 seconds
Total script execution time: 4.508929 seconds
Number of scripts executed: 340000 (336506 per second)
Number of scripts completed: 40000 (39589 per second)
Number of instructions executed: 22880000 (22.64M per second)

--- Performance (6 threads) ---
Total run time: 0.945903 seconds
Total script execution time: 5.203672 seconds
Number of scripts executed: 340000 (359444 per second)
Number of scripts completed: 40000 (42287 per second)
Number of instructions executed: 22880000 (24.19M per second)

--- Performance (7 threads) ---
Total run time: 0.846344 seconds
Total script execution time: 5.597741 seconds
Number of scripts executed: 340000 (401728 per second)
Number of scripts completed: 40000 (47262 per second)
Number of instructions executed: 22880000 (27.03M per second)

--- Performance (8 threads) ---
Total run time: 0.807679 seconds
Total script execution time: 6.141680 seconds
Number of scripts executed: 340000 (420959 per second)
Number of scripts completed: 40000 (49524 per second)
Number of instructions executed: 22880000 (28.33M per second)
```

Machine Two

- Type: 2012 Mac Mini
- CPU: 2.5GHz Intel Core i5 2 Cores
- OS: macOS “Sierra”

Machine Three

- Type: 2016 Mini PC
- CPU: 3.2GHz Intel Core i5 6500 4 Cores
- Ubuntu 16