

**De La Salle University • College of Computer Studies**

**Node JS and Load Tester**

Name (last name first) : Fernandez, Ryan austin

Poblete, Clarisse Felicia M.

San Pedro, Marc Dominic

Tan, Johansson E.

Section : G01

Date of Submission : March 31, 2016

**1. How is NodeJS architecture different from other servers?**

Most server-side frameworks make use of a synchronous architecture, which executes actions in order. NodeJS, on the other hand, makes use of an asynchronous architecture, which triggers actions through events. Instead of performing everything chronologically, it separates the regular tasks and IO tasks into different queues, so that the regular tasks do not always need to wait for the IO to finish before the program can proceed. It does this by running an event loop that handles all requests, delegations and callbacks.

**2. How does NodeJS improve the performance?**

The problem with synchronous architectures sometimes is they spend a lot of time waiting, especially waiting for IO, since a synchronous architecture would require the IO task to finish first before proceeding to the next one. This doesn’t always scale as well as an asynchronous architecture would. NodeJS’ asynchronous architecture allows it to not be as slowed down by IO and not spend as much time waiting, which often improves performance.

**3. What use-case/development is going to run slow on NodeJS?**

As previously mentioned, NodeJS makes use of an event loop to manage its tasks, but this event loop runs on a single thread. It is because of this that any time a synchronous operation is executed, NodeJS actually has to wait for that operation to finish before it can proceed, blocking any concurrent requests in the process due to the single-threaded nature of the event loop.