Node.js is very efficient with real-time applications: it facilitates handling multiple client requests, enables sharing and reusing packages of library code, and the data sync between the client and server happens very fast.

accepted

My feeling is that Node.js is especially suited for applications where you'd like to maintain a persistent connection from the browser back to the server. Using a technique known as ["long-polling"](http://en.wikipedia.org/wiki/Push_technology#Long_polling), you can write an application that sends updates to the user in real time. Doing long polling on many of the web's giants, like [Ruby on Rails](http://en.wikipedia.org/wiki/Ruby_on_Rails) or [Django](http://en.wikipedia.org/wiki/Django_%28web_framework%29), would create immense load on the server, because each active client eats up one server process. This situation amounts to a [tarpit](http://en.wikipedia.org/wiki/Tarpit_(networking)) attack. When you use something like Node.js, the server has no need of maintaining separate threads for each open connection.