C10K問題（クライアント1万台問題）

　C10K問題とは，Webサーバの性能に問題はなくとも，クライアントの数が膨大になるとサーバがパンクしてしまう問題である．この問題の要因として，ブロッキングI/Oが挙げられる．ブロッキングI/Oとは，「入出力が終わるまで他の処理を行わない方式」である．一般的なWebサーバはブロッキングI/Oで通信を行うが，通信が完了するまで処理を止めていてはクライアント(プロセス)数が増加した場合に通信が滞ってしまう．そこで，Webサーバをマルチスレッドにすれば良いと思われるが，そうするとWebサーバ側のスレッドが増えすぎるため，メモリ等が不足してしまうという問題がある．

erlang/Elixir

erlang/Elixirのプロセスは非同期式である．つまりerlang/Elixirで記述したWebサーバはクライアントの応答を待たずに処理を続けられるという特徴があり，C10K問題における「通信の完了を待たなければならない」という問題点を解決できる．

Node.js

Node.jsの特徴として，「ノンブロッキングI/O」と「イベントループ」が挙げられる．

ノンブロッキングI/O

ノンブロッキングI/OとはブロッキングI/Oの対義語であり，「入出力の完了を待たずに次の処理を受け付ける方式」を指す．これにより，シングルスレッドで複数の処理を非同期に行うことができ，少ないメモリ消費で多くのクライアントとの通信を実現できる．

イベントループ

イベントループとはスレッドの対義語である．これはリクエストやコールバックをイベントとして扱い，あるイベントの処理が終わった際に次のイベントの処理に移るといったものである．イベントループの欠点として，1つのイベントが止まる(ブロックされる)と全体が止まってしまうということが挙げられる．しかし，Node.jsは前述したノンブロッキングI/Oによって，この欠点を回避しているといえる．

参考

<http://www.hyuki.com/yukiwiki/wiki.cgi?TheC10kProblem>

<http://www.atmarkit.co.jp/news/analysis/200701/09/c10k.html>

<http://qiita.com/kyrieleison/items/f726061803a0e536c3e1>

<http://qiita.com/hayabusa333/items/7f950d59f4e96b5f9e5a>