Programming for Cloud

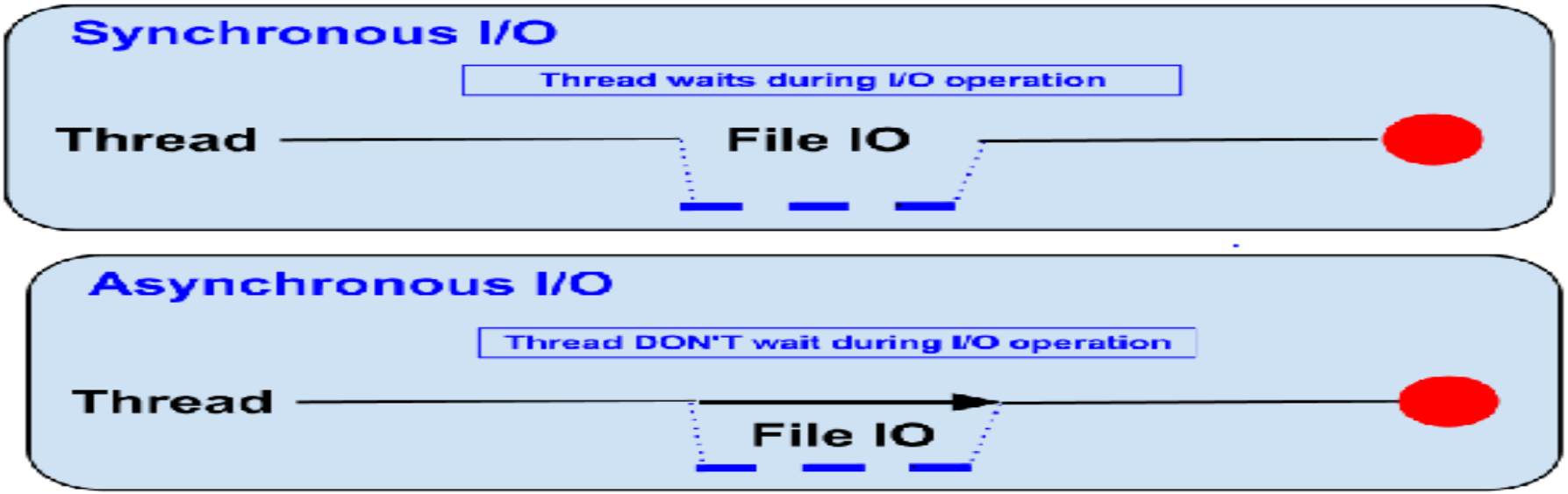
Dr. R. Karthi

Node.js

- Node.js is an open-source, cross-platform JavaScript runtime environment.
- Node.js runs V8 Chrome JavaScript engine outside the browser.
- Node has an event driven architecture capable of performing non-blocking (asynchronous) I/O operations.
- Node.js support client-side and server-side coding using java script.
- Node makes use of the event driven nature of JS by attaching callbacks to I/O requests.

Blocking vs Non-Blocking……

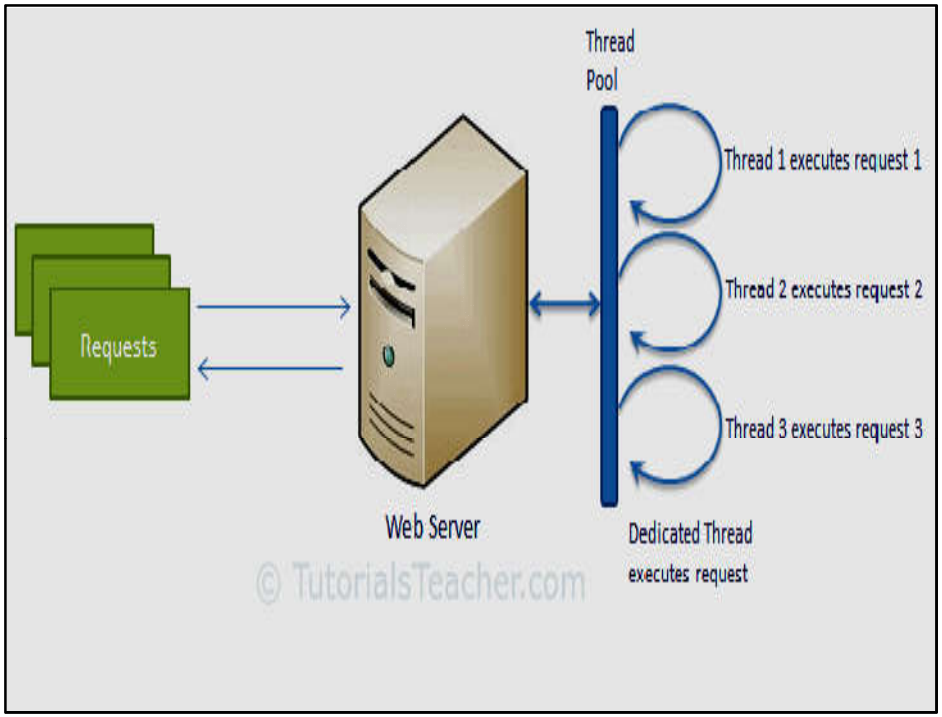
Read data from file and display the data



Traditional Web Server Model

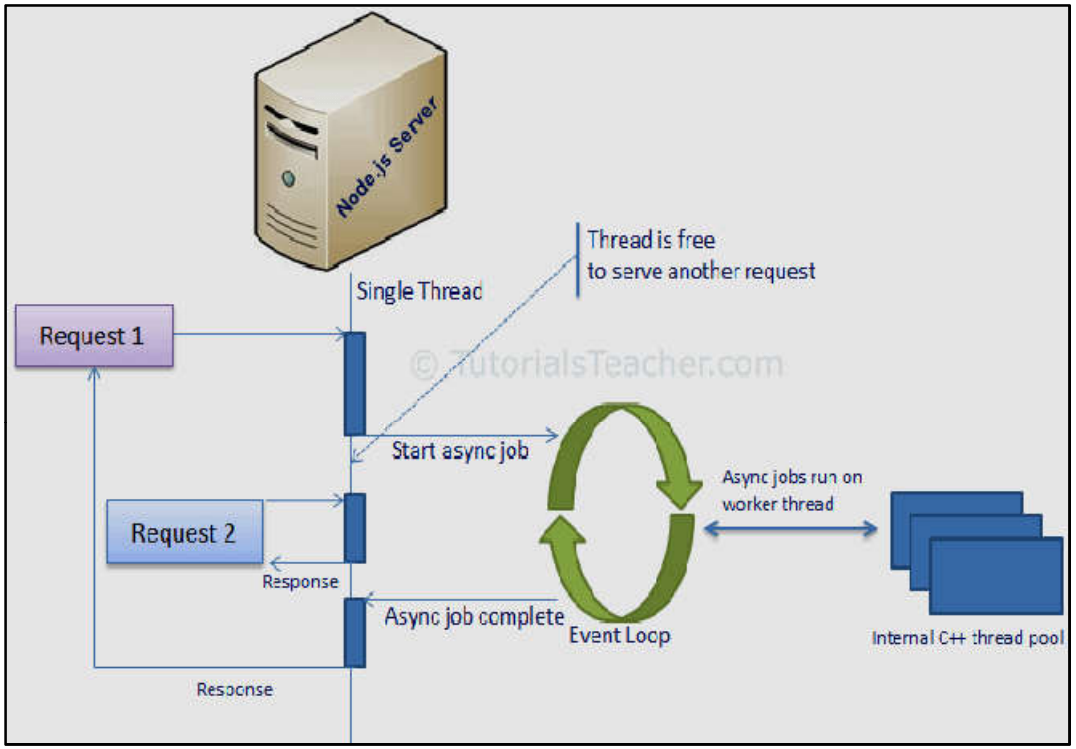
In web server model:

- Each request is handled by a dedicated **thread** from the thread pool.
- Dedicated thread executes a particular request and does not return until completion.
- If no thread is available request waits till the next thread is available.
- IO task or high computing processes slows the server performance.

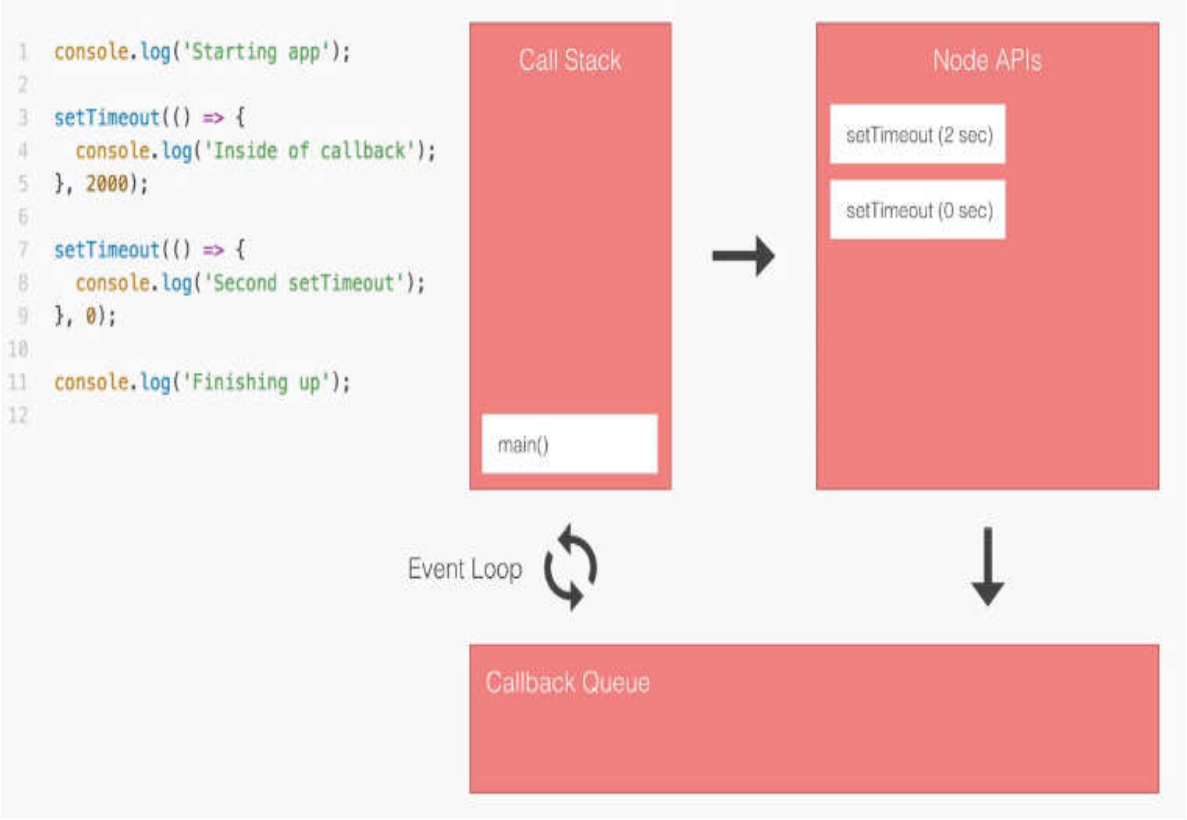


Node.js Process Model

- A Node.js application runs in a process with single thread.
- All the user requests to web application will be handled by the single thread.
- All the I/O operations or long running job is performed asynchronously for a particular request.
- Single thread does not wait for the request to complete and is free to handle the next request.
- When the asynchronous I/O operation is complete the thread processes the request further and sends the response.
- An event loop is constantly monitoring for the events to be raised for an asynchronous job and executes callback function when the asynchronous job is returns after completion.



Event Loops in Nodejs



Starting app
Finishing up
Second setTimeout
Inside of callback

https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/
https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/

Install Node and NPM

- **Download the installer from the Nodes.js® web site** https://nodejs.dev/ **and execute the installer.**
- Check the installation by using: node –version
- Node can be executed from command mode using node command
- npm is the standard package manager for Node.js. npm manages downloads of dependencies of your project.
  - npm install <package-name>
  - npm update <package-name>

Node programs

```
// Display the contents of current directory
const fs = require('fs')
fs.readdir('./', function(err, files)
{
    if(err)
        console.log('error',err);
    else
    console.log('DiR FILES',files);
}
);
```

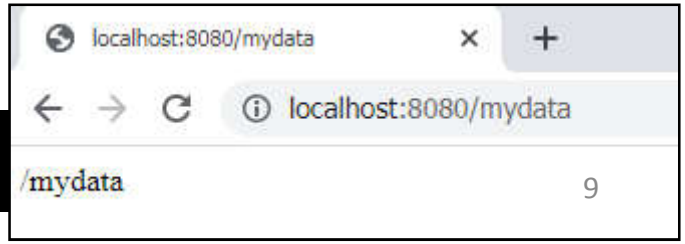
```
C:\Windows\system32\cmd.exe
    at startup (internal/bootstrap/node.js:283:19)
    at bootstrapNodeJSCore (internal/bootstrap/node.js:622:3)

E:\CloudComputing\2020\Class Material\programs>node nodepgm1.js
DiR FILES [ 'construct.js',
  'css_example.html',
  'doc_ex1.html',
  'doc_prop.html',
  'doc_table.html',
  'dom-event.html',
  'Dom_example2.html',
  'dom_onmouse.html',
```

Node Server

```
// send a request - response from server
var http = require('http');                    // built-in module called HTTP
const server = http.createServer( function (req, res)    // Creates a server
{
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(req.url);                                  // send response to user
    res.end();
}
);
server.listen(8080);        // server listen in this port
```

- function - **requestListener function** is the function that is executed each time the server gets a request
- Req - HTTP request the user sends is captured in a Request object
- Res - HTTP response that we return to the user is captured in respond object

```
E:\CloudComputing\2020\Class Material\programs>node nodepgm1.js
```



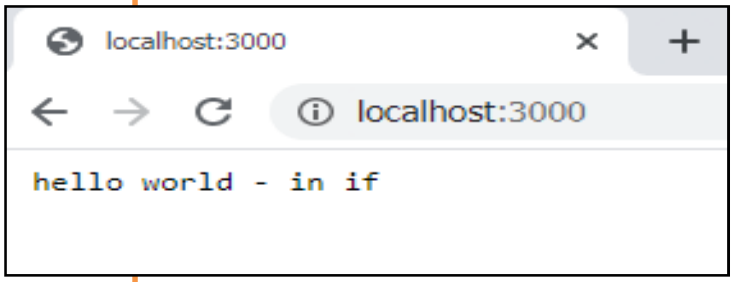
Node – Display HTML content

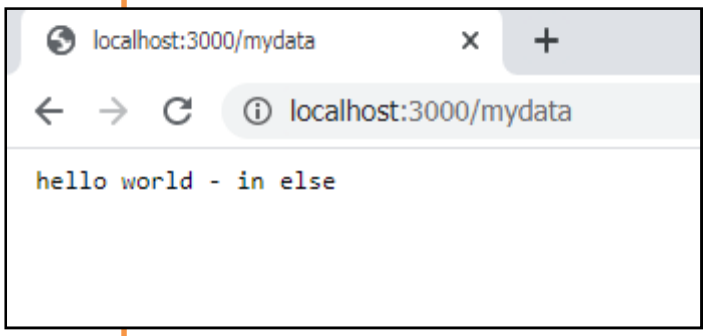
```
// Display a HTML content
const requestListener = function (req, res) {
        res.setHeader("Content-Type", "text/html");
        res.writeHead(200);
        res.end("<html><body><h1>This is HTML</h1></body></html>");
    };
    var http = require('http');
    const server = http.createServer(requestListener);
    server.listen(3000);
```

```
E:\CloudComputing\2020\Class Material\programs>node nodepgm1.js
```

Node – Route

```
// identify the route
const http = require('http');
const server = http.createServer(function(req,res){
   if(req.url ==='/')
   {
      res.write(" hello world - in if ");
      res.end();
   }
   else
   {
      res.write(" hello world - in else ");
      res.end();
   }
});
server.listen(3000);
```

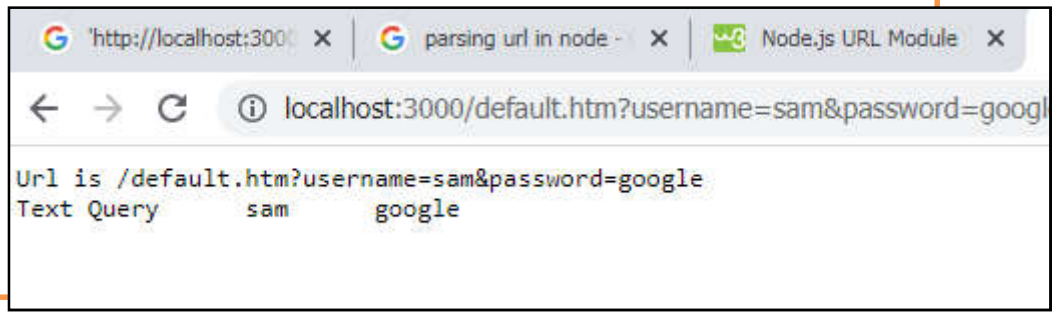





```
E:\CloudComputing\2020\Class Material\programs>node nodepgm1.js
```

Node – Parsing a URL

```
var http = require("http");
var url = require("url");
var server = http.createServer(function (req, res) {
   res.writeHead(200, {'Content-Type': 'text/plain'});
   res.write('Url is '+ req.url );
   var q = url.parse(req.url, true).query;
   var txt = q.username + "\t " + q.password;
   res.write('\nText Query \t' + txt);
   res.end();
});
server.listen(3000);
```



http://localhost:8080/app.js?foo=bad&baz=foo

- ❑ url.parse(req.url,true).**query** returns { foo: 'bad', baz: 'foo' }.
- ❑ url.parse(req.url,true).**host** returns 'localhost:8080'.
- ❑ url.parse(req.url,true).**pathname** returns '/app.js'.
- ❑ url.parse(req.url,true).**search** returns '?foo=bad&baz=foo'.

Activity

- https://www.youtube.com/watch?v=ztspvPYybIY
- https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/
- https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/

Thank You