Thread pool work scheduling

libuv provides a threadpool which can be used to run user code and get notified in the loop thread. This thread pool is internally used to run all file system operations, as well as getaddrinfo and getnameinfo requests.

Its default size is 4, but it can be changed at startup time by setting the `UV_THREADPOOL_SIZE` environment variable to any value (the absolute maximum is 1024).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 15)

Unknown directive type "versionchanged".

```
.. versionchanged:: 1.30.0 the maximum UV_THREADPOOL_SIZE allowed was increased from 128 to 1024.
```

The threadpool is global and shared across all event loops. When a particular function makes use of the threadpool (i.e. when using :c:func:`uv_queue_work`) libuv preallocates and initializes the maximum number of threads allowed by `UV_THREADPOOL_SIZE`. This causes a relatively minor memory overhead (~1MB for 128 threads) but increases the performance of threading at runtime.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 17); *backlink*

Unknown interpreted text role "c:func".

Note

Note that even though a global thread pool which is shared across all events loops is used, the functions are not thread safe.

Data types

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 31)

Unknown directive type "c:type".

```
.. c:type:: uv_work_t

    Work request type.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 35)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_work_cb)(uv_work_t* req)

    Callback passed to :c:func:`uv_queue_work` which will be run on the thread
    pool.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 40)

Unknown directive type "c:type".

```
.. c:type:: void (*uv_after_work_cb)(uv_work_t* req, int status)

    Callback passed to :c:func:`uv_queue_work` which will be called on the loop
    thread after the work on the threadpool has been completed. If the work
    was cancelled using :c:func:`uv_cancel` `status` will be ``UV_ECANCELED``.
```

Public members

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 50)

Unknown directive type "c:member".

```
.. c:member:: uv_loop_t* uv_work_t.loop

    Loop that started this request and where completion will be reported.
    Readonly.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 55)

Unknown directive type "seealso".

```
.. seealso:: The :c:type:`uv_req_t` members also apply.
```

API

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 61)

Unknown directive type "c:function".

```
.. c:function:: int uv_queue_work(uv_loop_t* loop, uv_work_t* req, uv_work_cb work_cb, uv_after_work_cb afte

    Initializes a work request which will run the given `work_cb` in a thread
    from the threadpool. Once `work_cb` is completed, `after_work_cb` will be
    called on the loop thread.

    This request can be cancelled with :c:func:`uv_cancel`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-master\deps\uv\docs\src\(node-master)(deps)(uv)(docs)(src)threadpool.rst, line 69)

Unknown directive type "seealso".

```
.. seealso:: The :c:type:`uv_req_t` API functions also apply.
```