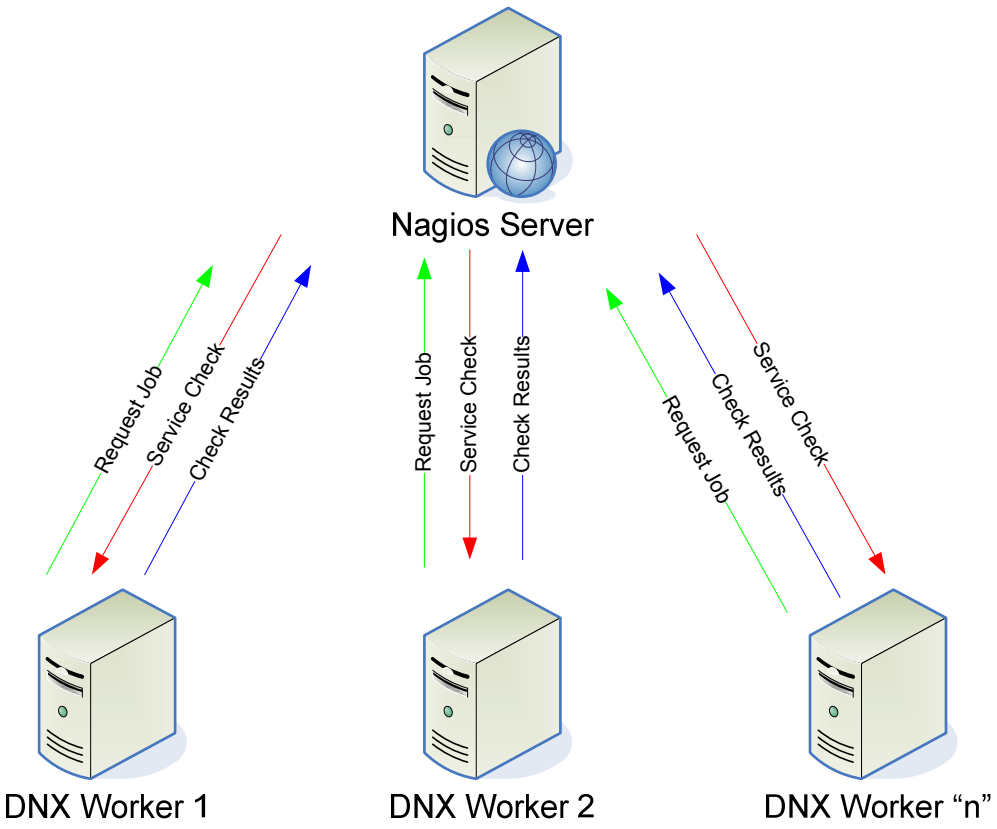
Distributed Nagios eXecutor Work Flow

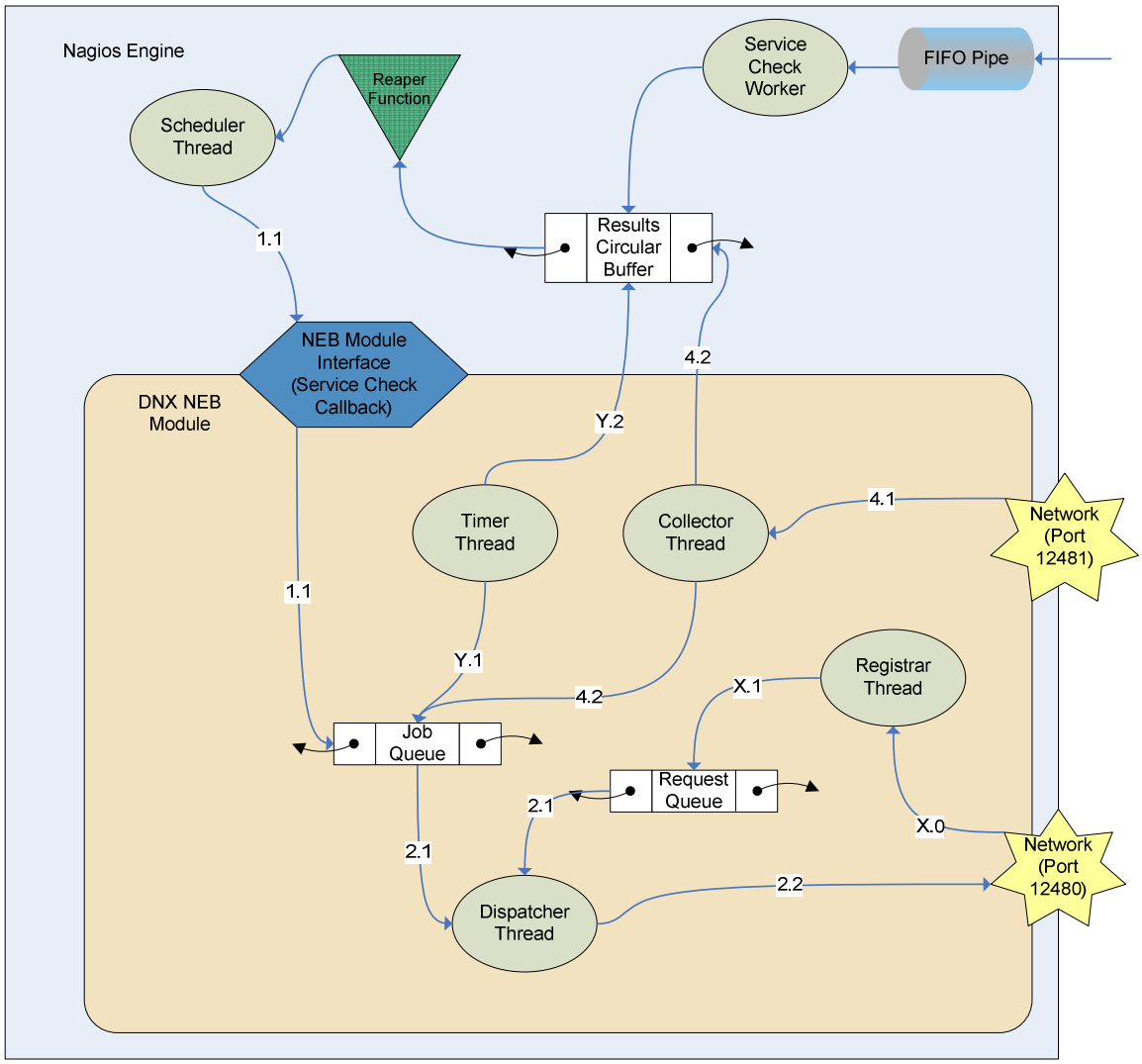
The sole purpose of DNX is to off load the work of plug-in execution to servers other than the Nagios scheduling server. Many have experienced high load and high check latency on larger Nagios installations. NSCA is one way to alleviate this, but requires highly coupled configuration in disparate locations. Through the Nagios Event Broker (NEB) interface DNX will distribute the work load to other servers while keeping its configuration simple and unified.

High Level Operations

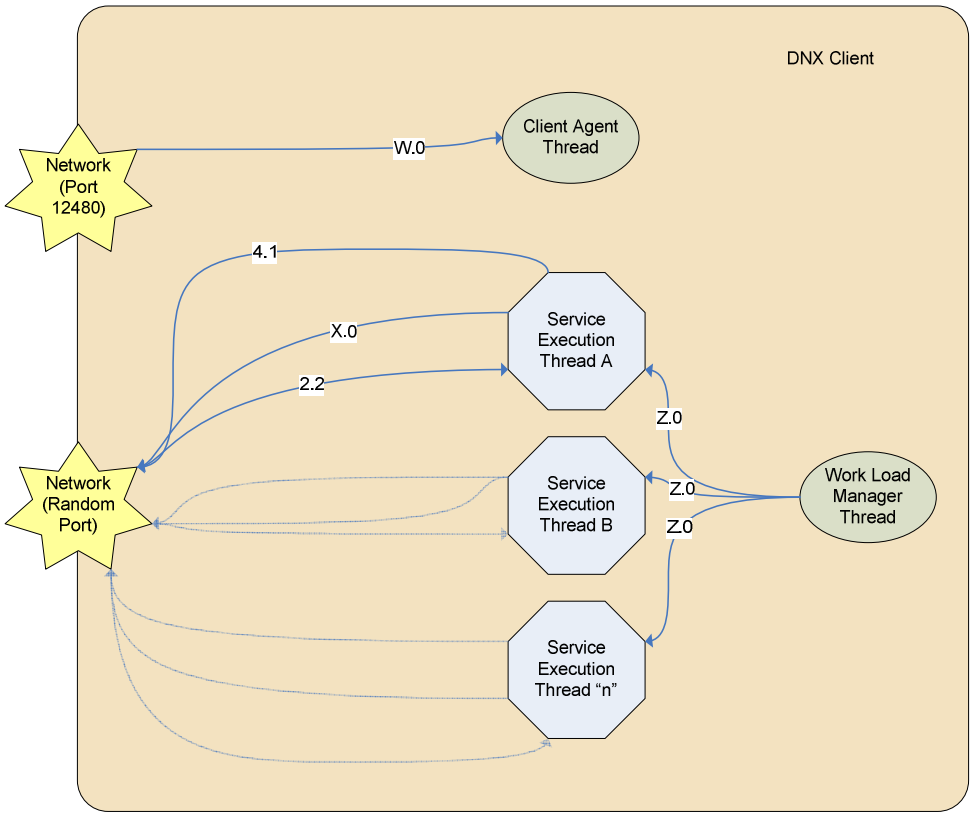


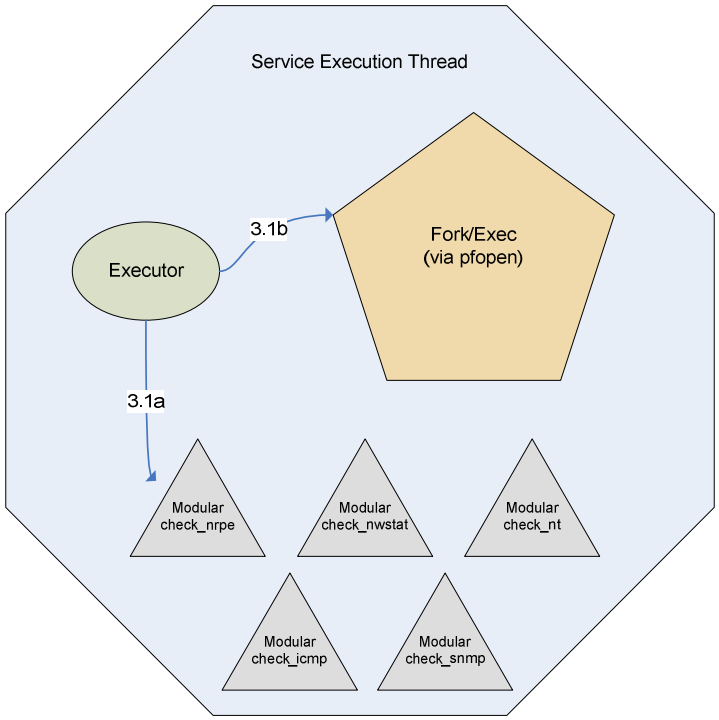
DNX Worker Servers request work to do from the Nagios (Scheduling) Server. They execute the plug-in and return the results to Nagios.

In-depth Work Flow – Server Side



In-depth Workflow – Client Side





Work Flow Description

W.0 – External command interface for shutdown, restart, performance queries, etc.
X.0 – Service Execution Thread requests a job to perform and put in a queue.
Y.0 – Timer Thread looks for jobs which are old and reports a timeout to Nagios.
Z.0 – Work Load Manager creates and destroys execution threads as needed.

1.1 – DNX NEB Module “intercepts” service checks and puts them in the Job Queue.

2.1 -- Dispatcher Thread takes any Jobs in the job queue and assigns them to a Request from the request queue.
2.2 – Dispatcher sends the command to the requesting execution thread.

3.1 – Service Execution Thread runs the plug-in looking first for “in-core (client)” libraries and doing a fork/exec if needed.

4.1 – Service Execution Thread returns the output from the plug-in to the Collector on the Server side.
4.2 – Collector Thread matches the incoming results with a “dispatched” job in the job queue and posts the results back to Nagios proper.

