Reduce Memory

There are a few parameters that can be dialed down to reduce the memory usage of `bitcoind`. This can be useful on embedded systems or small VPSes.

In-memory caches

The size of some in-memory caches can be reduced. As caches trade off memory usage for performance, reducing these will usually have a negative effect on performance.

- `-dbcache=<n>` - the UTXO database cache size, this defaults to `450`. The unit is MiB (1024).
  - The minimum value for `-dbcache` is 4.
  - A lower `-dbcache` makes initial sync time much longer. After the initial sync, the effect is less pronounced for most use-cases, unless fast validation of blocks is important, such as for mining.

Memory pool

- In Bitcoin Core there is a memory pool limiter which can be configured with `-maxmempool=<n>`, where `<n>` is the size in MB (1000). The default value is `300`.

  - The minimum value for `-maxmempool` is 5.
  - A lower maximum mempool size means that transactions will be evicted sooner. This will affect any uses of `bitcoind` that process unconfirmed transactions.

- To completely disable mempool functionality there is the option `-blocksonly`. This will make the client opt out of receiving (and thus relaying) transactions completely, except as part of blocks.

  - Do not use this when using the client to broadcast transactions as any transaction sent will stick out like a sore thumb, affecting privacy. When used with the wallet it should be combined with `-walletbroadcast=0` and `-spendzeroconfchange=0`. Another mechanism for broadcasting outgoing transactions (if any) should be used.

- Since `0.14.0`, unused memory allocated to the mempool (default: 300MB) is shared with the UTXO cache, so when trying to reduce memory usage you should limit the mempool, with the `-maxmempool` command line argument.

Number of peers

- `-maxconnections=<n>` - the maximum number of connections, which defaults to 125. Each active connection takes up some memory. This option applies only if inbound connections are enabled; otherwise, the number of connections will not be more than 11. Of the 11 outbound peers, there can be 8 full-relay connections, 2 block-relay-only ones, and occasionally 1 short-lived feeler or extra outbound block-relay-only connection.

- These limits do not apply to connections added manually with the `-addnode` configuration option or the `addnode` RPC, which have a separate limit of 8 connections.

Thread configuration

For each thread a thread stack needs to be allocated. By default on Linux, threads take up 8MiB for the thread stack on a 64-bit system, and 4MiB in a 32-bit system.

- `-par=<n>` - the number of script verification threads, defaults to the number of cores in the system minus one.
- `-rpcthreads=<n>` - the number of threads used for processing RPC requests, defaults to `4`.

Linux specific

By default, since glibc `2.10`, the C library will create up to two heap arenas per core. This is known to cause excessive memory usage in some scenarios. To avoid this make a script that sets `MALLOC_ARENA_MAX` before starting bitcoind:

```bash
#!/usr/bin/env bash
export MALLOC_ARENA_MAX=1
bitcoind
```

The behavior was introduced to increase CPU locality of allocated memory and performance with concurrent allocation, so this setting could in theory reduce performance. However, in Bitcoin Core very little parallel allocation happens, so the impact is expected to be small or absent.