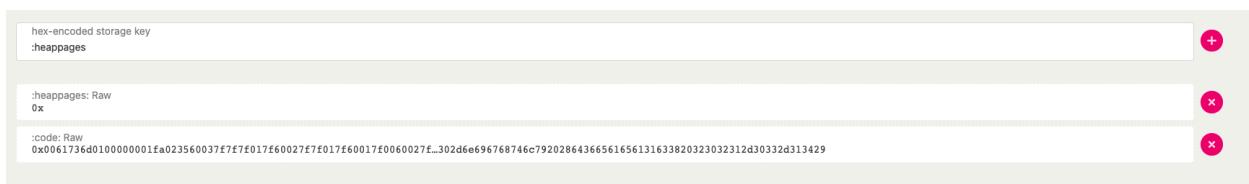


Proposal: Increasing the WASM heap pages.

Recently, we've noticed that some validator nodes on polkadot, encounter a “*allocator failed to allocate memory*” error around the time when the staking election (phragmen) is being computed. Recall that this computation is happening in an **offchain worker thread**. Nonetheless, in the current implementation, the offchain worker thread, which itself is executed in WASM, uses the same WASM environment as the canonical runtime, and therefore has **limited memory**. To prevent the situation where no validator submits an election solution, the amount of memory accessible to the offchain worker environment should be increased.

Note that ideally, this change should only affect the offchain worker thread, and not the main runtime. But as mentioned, currently they share the *same configurations*. A long term solution is to separate these configurations and only increase the amount of memory that is accessible to the offchain worker ([PR](#)). But, since this is not *readily* possible, we opt to propose increasing the overall WASM runtime memory pages, which affects both block execution and offchain worker. We expect this decision to have **no notable effect** on the main runtime's progress and execution.

The amount of memory available to WASM is determined in the number of “pages of memory” that can be allocated in the called, aka. “**Heap pages**”. This value lives on-chain, and you can inspect it similar to other [well-known-keys via “:heappages”](#). Currently, no value is set there and [a default of 1024](#) is used.



Luckily, [an extrinsic is already exposed](#) in the system pallet that exactly sets this value. Therefore, the proposal is simply this call, giving us the proposal hash of: **0x5423d109316e89a312b0f92bbfb7c46b390f5a0480b8b75bdadc9b204d28ffab**



Note that a reason for the urgency of this proposal is that if the *offchain* election fails due to any reason, the chain does the *on-chain backup* election. This election does not have a

post-processing that trims nominations as much as possible, therefore most nominators will split their backing between *many* validators, instead of backing *just one* of them. This will in turn cause the validators to become **severely oversubscribed**. An example of this can be seen [here](#).