AIP: ArbOS Version 11

Constitutional

Abstract

This AIP introduces a number of improvements to Arbitrum chains, including support for the EVM Shanghai upgrade and the PUSH0 opcode, along with miscellaneous bug fixes. These improvements are now audited and ready for adoption, including by Arbitrum Orbit chains, Arbitrum One, and Arbitrum Nova. This proposal concerns the latter two, as they are governed by the Arbitrum DAO. On a high level an ArbOS upgrade can be seen as Arbitrum's equivalent of a hardfork - more can be read about the subject over in Arbitrum ArbOS upgrades

Changes Included

1. EVM Shanghai support (including the PUSH0 opcode)

Recent versions of go-ethereum already includes support for the changes to the EVM made in the Shanghai L1 upgrade, but we need to enable them for Arbitrum chains. Instead of using a time based activation, it's better to activate support based on the ArbOS version, which makes sure that even if the upgrade is delayed, Shanghai support will take effect uniformly and without causing divergences with out of date node software.

PRs:

github.com/OffchainLabs/nitro

Build against geth Shanghai branch

OffchainLabs:master

← OffchainLabs:Shanghai

opened 07:36PM - 20 Apr 23 UTC

[

amsanghi
](https://github.com/amsanghi)

[+18

-48

](https://github.com/OffchainLabs/nitro/pull/1583/files)

This PR contains the changes needed to Nitro to use the OffchainLabs geth fork w..

ith upstream's Shanghai merged into it.

Note: This PR currently changes the git submodule for go-ethereum to use the Shanghai branch, which is being reviewed in https://github.com/OffchainLabs/go-ethereum/pull/211. We'll have to change it back to using master after the other PR is merged.

github.com/OffchainLabs/nitro

Regenerate the state and header after the internal transaction in ProduceBlockAdvanced

OffchainLabs:master

← OffchainLabs:ProduceBlockAdvance_during_upgrade

opened 04:57PM - 24 May 23 UTC

[

amsanghi
](https://github.com/amsanghi)

[+106

-40

](https://github.com/OffchainLabs/nitro/pull/1664/files)

github.com/OffchainLabs/nitro

Only update ArbOS version after recreating header after internal tx

OffchainLabs:master

← OffchainLabs:only-update-arbos-version

opened 05:02PM - 20 Jun 23 UTC

[

PlasmaPower
](https://github.com/PlasmaPower)

[+7

-27

](https://github.com/OffchainLabs/nitro/pull/1712/files)

I believe attempting to recreate the header entirely was causing issues because…

e.g. the basefee ArbOS thinks we should have might've changed after the internal transaction, and we need to preserve behavior.

2. Retryable fixes

Retryable fees previously always used the network fee account, instead of also using the infrastructure fee account. The infrastructure fee account should be paid fees from the basefee, and the network fee account should be paid any surplus fees when the gas price is elevated. That was correctly implemented for normal transactions, but retryables only dealt with the network fee account. Retryable redemption also reported an incorrect gas usage in the block header. This matters to Arbitrum Nova, where the infrastructure fee account pays out some fees to the Data Availability Committee members, but the network fee account does not.

PRs:

github.com/OffchainLabs/nitro

Fix retrayble fee refund

OffchainLabs:master

← OffchainLabs:fix-retrayble-refund

opened 05:41PM - 01 Aug 23 UTC

[

magicxyyz
](https://github.com/magicxyyz)

[+364

-55

](https://github.com/OffchainLabs/nitro/pull/1789/files)

github.com/OffchainLabs/nitro

Fix retryable scheduling gas usage

OffchainLabs:master

← OffchainLabs:retryable-gas-usage-fix

opened 06:14AM - 22 Nov 23 UTC

[

PlasmaPower
](https://github.com/PlasmaPower)

[+15

-0

](https://github.com/OffchainLabs/nitro/pull/1980/files)

This PR also marks ArbOS 11 as finalized

3. Fix the chain owner list returned by precompile

This change doesn’t affect the actual chain owner set, but the list being returned by the ArbOwnerPublic precompile was incorrect for Arbitrum Nova due to an internal ArbOS issue. To be clear, this does not affect who was able to make chain owner actions. As intended, only the DAO is able to make chain owner actions on Arbitrum One and Arbitrum Nova. This change only affects the list of chain owners presented by the ArbOwnerPublic precompile.

PRs:

github.com/OffchainLabs/nitro

Fix address set AllMembers list

OffchainLabs:master

← OffchainLabs:fix-address-set

opened 02:46PM - 16 Mar 23 UTC

[

PlasmaPower
](https://github.com/PlasmaPower)

[+99

-10

](https://github.com/OffchainLabs/nitro/pull/1525/files)

This didn't impact whether an address is actually in the set (IsMember), just it…

erating over AllMembers. The fix will take place in ArbOS version 11.

github.com/OffchainLabs/nitro

Resolve chain owners list in ArbOS version 11 upgrade

OffchainLabs:master

← OffchainLabs:resolve-chain-owners

opened 09:23PM - 18 Jul 23 UTC

[

ganeshvanahalli
](https://github.com/ganeshvanahalli)

[+225

-4

](https://github.com/OffchainLabs/nitro/pull/1768/files)

This PR creates a new function to resolve incorrect mapping between ChainOwners…

map and list.

PR for changes to the ArbOwner interface file inside solidity contract -> https://github.com/OffchainLabs/nitro-contracts/pull/46

4. Fix some precompile methods taking up all gas when reverting

Some precompile methods such as ArbSys's arbBlockHash method took up all gas when reverting. That meant that if a transaction called arbBlockHash with an out-of-range block number, it'd use up all the gas when reverting.

PR:

github.com/OffchainLabs/nitro

Fix precompiles using all gas on non-solidity errors

OffchainLabs:master

← OffchainLabs:precompile-error-gas-usage

opened 02:28AM - 14 Feb 23 UTC

[

PlasmaPower
](https://github.com/PlasmaPower)

[+127

-27

](https://github.com/OffchainLabs/nitro/pull/1490/files)

Pulls in https://github.com/OffchainLabs/go-ethereum/pull/199

This PR also ad…

ds a nice Solidity error for out of bound ArbSys arbBlockHash calls :)

5. Create missing precompile methods to view some L1 pricing parameters

The L1RewardReceipient and L1RewardRate were previously not exposed via precompiles. This change adds methods to get them to ArbGasInfo so that the current chain configuration can be easily checked.

PR:

github.com/OffchainLabs/nitro

Add GetL1RewardRate and GetL1RewardRecipient methods to ArbGasInfo

OffchainLabs:master

← OffchainLabs:l1Pricer-reward-methods

opened 03:27PM - 20 Jul 23 UTC

[

ganeshvanahalli
](https://github.com/ganeshvanahalli)

[+21

-1

](https://github.com/OffchainLabs/nitro/pull/1775/files)

This PR creates two new functions to get L1 pricer reward rate and reward recipi…

ent in ArbOS version 11

PR for changes to the ArbGasInfo interface file inside solidity contract -> https://github.com/OffchainLabs/nitro-contracts/pull/44

6. Fix the possibility of a staticcall from the owner to ArbOwner emitting a log

This shouldn’t matter in practice, but it was theoretically for a staticcall from the chain owner to the ArbOwner precompile to emit a log. In the EVM, staticcalls should never be able to emit logs. This PR fixes the Arbitrum precompile logic to disallow emitting logs in staticcall contexts.

PR:

github.com/OffchainLabs/nitro

ArbOS: Revert when creating logs in static contexts

OffchainLabs:master

← OffchainLabs:precompile-static-logs

opened 08:42PM - 11 Apr 23 UTC

[

rachel-bousfield
](https://github.com/rachel-bousfield)

[+57

-16

](https://github.com/OffchainLabs/nitro/pull/1562/files)

This PR adds a check in ArbOS to revert when attempting to create logs in static...

contexts. While safer to have, this scenario isn't possible in production as there's only two ways to do this - A chain owner (the DAO or Security Counsel) could call ArbOwner - A call to ArbDebug, which only exists on test chains

Nevertheless for future proofing's sake, this PR is a reasonable change.

7. Fix default L1 pricing params

This shouldn’t matter for Arbitrum One and Arbitrum Nova, because these parameters were already corrected in AIP-7. However, it’s included in ArbOS version 11 so that any Arbitrum Orbit chains automatically get the correct parameters.

PR:

github.com/OffchainLabs/nitro

Update L1 pricing params to more accurate values in ArbOS version 11

OffchainLabs:master

← OffchainLabs:update-l1-pricing-params

opened 04:30AM - 04 Apr 23 UTC

[

PlasmaPower
](https://github.com/PlasmaPower)

[+23

-14

](https://github.com/OffchainLabs/nitro/pull/1549/files)

Implementation

The canonical version of ArbOS 11 this proposal aims to adopt is implemented in the Arbitrum Nitro git commit hash df93361a1293574903f28fbbbe0469a3ea5c644d

That commit builds a WASM module root of 0xf4389b835497a910d7ba3ebfb77aa93da985634f3c052de1290360635be40c4a

, which is a hash of the code that’s put on-chain for fraud proofs. You can verify this WASM module root on a x86_64 computer (i.e. not on ARM which builds slightly different symbol names) with Docker setup by checking out the previously mentioned commit of the nitro repository, running git submodule update --init --recursive

, and then running docker build . --target module-root-calc --tag wavm-machine && docker run --rm wavm-machine cat /workspace/target/machines/latest/module-root.txt

This uses Docker to perform a reproducible build of the WebAssembly code used in fraud proofs, and outputs its hash.

The above code has audited by Trail of Bits - the audit report can be viewed inOffchain Labs ArbOS 11 Essential Upgrade.pdf - Google Drive

Upgrade Action smart contracts

The Action smart contracts used to execute the on-chain upgrade can be viewed in

github.com/ArbitrumFoundation/governance

upgrade arbos action contracts

ArbitrumFoundation:main

← ArbitrumFoundation:arbos-upgrade-actions

opened 04:45PM - 01 Aug 23 UTC

[

DZGoldman
](https://github.com/DZGoldman)

[+425

-36

](https://github.com/ArbitrumFoundation/governance/pull/170/files)

- Gov action contracts for ArbOS 11 https://forum.arbitrum.foundation/t/aip-arbo…

s-version-11/19696

Action contract addresses:

SetArbOneArbOS11ModuleRootAction

SetNovaArbOS11ModuleRootAction

SetArbOS11VersionAction

SetArbOS11VersionAction

The above code has audited by Trail of Bits - the audit report can be viewed inOffchain Labs Governance Actions Summary Report.pdf - Google Drive

Verifying the ArbOS Code Difference

The current ArbOS version used on Arbitrum One and Arbitrum Nova is ArbOS 10, corresponding to the Arbitrum Nitro consensus-v10

git tag. You can verify this by running the previously mentioned steps to build the WASM module root on that git tag, which produces the WASM module root 0x6b94a7fc388fd8ef3def759297828dc311761e88d8179c7ee8d3887dc554f3c3

, which is what the rollup contract's wasmModuleRoot()

method returns for both Arbitrum One and Arbitrum Nova.

To audit the code difference from ArbOS 10 to ArbOS 11, you could simple generate a full nitro diff with git diff consensus-v10 df93361a1293574903f28fbbbe0469a3ea5c644d

(and also generate a diff of the go-ethereum submodule mentioned in that nitro diff). However, that includes a lot of code that isn't part of the WASM module root. To filter down to just the replay binary which defines the state transition function, you can start by generating a list of files in the nitro and go-ethereum repositories included by the replay binary in either ArbOS 10 or ArbOS 11 with bash:

!/usr/bin/env bash

set -e mkdir -p ~/tmp # this script uses ~/tmp as scratch space and output

this script should be run in the nitro repository

git checkout df93361a1293574903f28fbbbe0469a3ea5c644d git submodule update --init --recursive make solgen go list -f "{{.Deps}}" ./cmd/replay | tr -d '[]' | sed 's/ /\n/g' | grep 'github.com/offchainlabs/nitro/' | sed 's@github.com/offchainlabs/nitro/@@' | while read dir; do find "$dir" -type f -name '*.go' -maxdepth 1; done | grep -v '_test.go$' > ~/tmp/consensus-v11-nitro-files.txt go list -f "{{.Deps}}" ./cmd/replay | tr -d '[]' | sed 's/ /\n/g' | grep 'github.com/ethereum/go-ethereum/' | sed 's@github.com/ethereum/go-ethereum/@go-ethereum/@' | while read dir; do find "$dir" -type f -name '*.go' -maxdepth 1; done | grep -v '_test.go$' > ~/tmp/consensus-v11-geth-files.txt rm -r contracts git checkout consensus-v10 git submodule update --init --recursive make solgen rm -r blockscout go list -f "{{.Deps}}" ./cmd/replay | tr -d '[]' | sed 's/ /\n/g' | grep 'github.com/offchainlabs/nitro/' | sed 's@github.com/offchainlabs/nitro/@@' | while read dir; do find "$dir" -type f -name '*.go' -maxdepth 1; done | grep -v '_test.go$' > ~/tmp/consensus-v10-nitro-files.txt go list -f "{{.Deps}}" ./cmd/replay | tr -d '[]' | sed 's/ /\n/g' | grep 'github.com/ethereum/go-ethereum/' | sed 's@github.com/ethereum/go-ethereum/@go-ethereum/@' | while read dir; do find "$dir" -type f -name '*.go' -maxdepth 1; done | grep -v '_test.go$' > ~/tmp/consensus-v10-geth-files.txt sort -u ~/tmp/consensus-v10-nitro-files.txt ~/tmp/consensus-v11-nitro-files.txt > ~/tmp/replay-binary-nitro-dependencies.txt sort -u ~/tmp/consensus-v10-geth-files.txt ~/tmp/consensus-v11-geth-files.txt | sed 's@^[./]*go-ethereum/@@' > ~/tmp/replay-binary-geth-dependencies.txt

Now, ~/tmp/replay-binary-dependencies.txt

contains a list of dependencies of the replay binary that were present in ArbOS 10 or 11. To use that to generate a smaller diff of the nitro repository, you can run:

git diff consensus-v10 df93361a1293574903f28fbbbe0469a3ea5c644d -- cmd/replay $(cat ~/tmp/replay-binary-nitro-dependencies.txt)

For the go-ethereum submodule, you can first find out what go-ethereum commit ArbOS 10 and 11 used:

$ git ls-tree consensus-v10 go-ethereum 128 ↩ 160000 commit 941aa323e5cbbde1d5806eb4e3dd70553c5a1298 go-ethereum $ git ls-tree df93361a1293574903f28fbbbe0469a3ea5c644d go-ethereum 160000 commit abe584818e104dd5b4fdb8f60381a14eede896de go-ethereum

Those commit hashes are the go-ethereum commit hashes pinned by each nitro commit. Then, you can again use git diff

and the files generated by the earlier script to generate a diff limited to code used by the replay binary:

this should be run inside the go-ethereum submodule folder

git diff 941aa323e5cbbde1d5806eb4e3dd70553c5a1298 abe584818e104dd5b4fdb8f60381a14eede896de -- $(cat ~/tmp/replay-binary-geth-dependencies.txt)

This diff also includes the diff between upstream go-ethereum versions v1.10.25 and v1.11.6, as ArbOS 10 used the former and ArbOS 11 uses the latter. To filter out that difference, you can use this tool to find the intersection of two git diffs: Git diff intersection finder · GitHub

We can use that to find the intersection of the diff of ArbOS 11’s go-ethereum against ArbOS 10’s go-ethereum and the diff of ArbOS 11’s go-etheruem against upstream go-ethereum v1.11.6:

this should be run inside the go-ethereum submodule folder

git diff 941aa323e5cbbde1d5806eb4e3dd70553c5a1298 abe584818e104dd5b4fdb8f60381a14eede896de -- $(cat ~/tmp/replay-binary-geth-dependencies.txt) > ~/tmp/arbos-10-vs-11-geth.diff git diff v1.11.6 abe584818e104dd5b4fdb8f60381a14eede896de -- $(cat ~/tmp/replay-binary-geth-dependencies.txt) > ~/tmp/arbos-11-vs-upstream-geth.diff diff-intersection.py ~/tmp/arbos-10-vs-11-geth.diff ~/tmp/arbos-11-vs-upstream-geth.diff

In addition, you can pass the following arguments to diff-intersection.py to ignore files that are included by the replay binary but whose components are not used: --ignore-files 'core/blockchain*go' arbitrum_types/txoptions.go 'rawdb/' 'rpc/*'*

Note that by default, diff-intersection.py

does a line based intersection. To instead do an intersection based on chunks in the diff, known as hunks in git terminology, you can add the --only-hunks

flag.