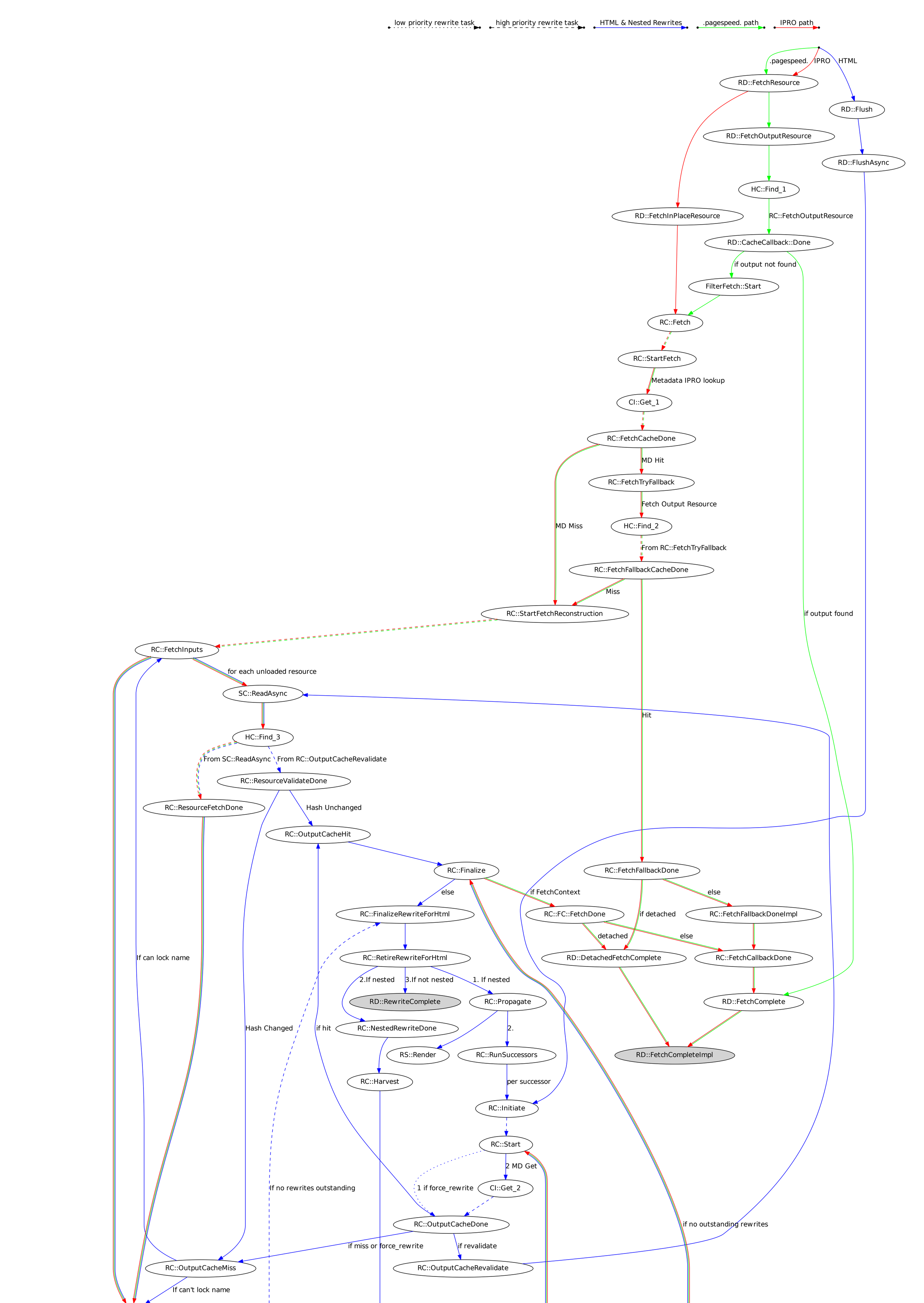

low priority rewrite task
high priority rewrite task
HTML & Nested Rewrites
.pagespeed. path
IPRO path
.pagespeed.
IPRO
HTML
RD::FetchResource
RD::Flush
RD::FetchOutputResource
RD::FlushAsync
HC::Find_1
RD::FetchInPlaceResource
RC::FetchOutputResource
RD::CacheCallback::Done
if output not found
FilterFetch::Start
RC::Fetch
RC::StartFetch
Metadata IPRO lookup
CI::Get_1
RC::FetchCacheDone
MD Hit
RC::FetchTryFallback
Fetch Output Resource
MD Miss
HC::Find_2
From RC::FetchTryFallback
RC::FetchFallbackCacheDone
Miss
RC::StartFetchReconstruction
if output found
RC::FetchInputs
for each unloaded resource
SC::ReadAsync
Hit
HC::Find_3
From SC::ReadAsync
From RC::OutputCacheRevalidate
RC::ResourceValidateDone
RC::ResourceFetchDone
Hash Unchanged
RC::OutputCacheHit
RC::Finalize
RC::FetchFallbackDone
else
if FetchContext
RC::FinalizeRewriteForHtml
RC::FC::FetchDone
if detached
RC::FetchFallbackDoneImpl
detached
else
If can lock name
RC::RetireRewriteForHtml
RD::DetachedFetchComplete
RC::FetchCallbackDone
2.If nested
3.If not nested
1. If nested
RD::RewriteComplete
RC::Propagate
RD::FetchComplete
Hash Changed
if hit
RC::NestedRewriteDone
2.
RS::Render
RC::RunSuccessors
RD::FetchCompleteImpl
RC::Harvest
per successor
RC::Initiate
RC::Start
2 MD Get
If no rewrites outstanding
1 if force_rewrite
CI::Get_2
RC::OutputCacheDone
if no outstanding rewrites
if miss or force_rewrite
if revalidate
RC::OutputCacheMiss
RC::OutputCacheRevalidate
If can't lock name
RC::Activate
Once fetches complete, HTML
RC::StartRewriteForHtml
Once fetches complete, IPRO
RC::PartitionAsync
2
RC::StartRewriteForFetch
RC::PartitionDone
if rewritable & can fall back
if rewritable & can't fall back
else
not rewritable
RC::Rewrite
2. After filter done rewriting
1. If necessary (IPRO sets force_rewrite on nested tasks)
RC::RewriteDone
RC::StartNestedTasks
Note: Nested rewrites colored as HTML
RC::RewriteDoneImpl
RC::StartNestedTasksImpl
Pagespeed Optimization Library (PSOL)