

Video Decoder

2019年11月26日 20:46

所有的mojo消息都是通过 mojo/core/core.cc 来收发的，因此，可以在此处打断点来追踪消息的发送时机和接收时机。

Core::CreateMessage: 所有创建消息都会调用。

运行在Chrome_IOThread线程，构造一条ResourceHandler链，用于接收下载后的资源，链尾是MojoAsyncResourceHandler，它通过mojo接口把数据传输出去。ResourceLoader包装了URLRequest，用来执行下载。

```
#0 0x00007ffffd3ba0e6 in net::URLRequest::Start() (this=0x3a6284911020) at ../../src/net/url_request/url_request.cc:524
#1 0x00007ffffd3e6d86 in content::ResourceLoader::StartRequestInternal() (this=0x3a628596ec20)
at ../../src/content/browser/loader/resource_loader.cc:627
#2 0x00007ffffd3e65be in content::ResourceLoader::Resume(bool, base::Optional<net::HttpRequestHeaders> const&) (this=0x3a628596ec20,
called_from_resource_controller=false, modified_request_headers=...) at ../../src/content/browser/loader/
resource_loader.cc:546
#3 0x00007ffffd3e805d in content::ResourceLoader::ScopedDeferral::~ScopedDeferral() (this=0x7fffc174a600)
at ../../src/content/browser/loader/resource_loader.cc:217
#4 0x00007ffffd3e03ef in content::ResourceLoader::StartRequest() (this=0x3a628596ec20)
at ../../src/content/browser/loader/resource_loader.cc:283
#5 0x00007ffffd3cc3a5 in content::ResourceDispatcherHostImpl::StartLoading(content::ResourceRequestInfoImpl*, std::__
1::unique_ptr<content::ResourceLoader, std::__1::default_delete<content::ResourceLoader> >) (this=0x3a62840bf180, info=0x
3a6287dfc140, loader=...) at ../../src/content/browser/loader/resource_dispatcher_host_impl.cc:1904
#6 0x00007ffffd3c1058 in content::ResourceDispatcherHostImpl::BeginRequestInternal(std::__1::unique_ptr<net::URLRequest, std::__
1::default_delete<net::URLRequest> >, std::__1::unique_ptr<content::ResourceHandler, std::__1::default_delete<
content::ResourceHandler> >, bool, std::__1::unique_ptr<network::ScopedThrottlingToken, std::__
1::default_delete<network::ScopedThrottlingToken> >) (this=0x3a62840bf180, request=..., handler=..., is_initiated_by_fetch_api=false,
throttling
_token=...) at ../../src/content/browser/loader/resource_dispatcher_host_impl.cc:1785
#7 0x00007ffffd3cc8f8 in content::ResourceDispatcherHostImpl::BeginNavigationRequest(content::ResourceContext*, net::URLRequestContext*,
storage::FileSystemContext*, content::NavigationRequestInfo const&, std::__1::unique_ptr<content::Nav
igationUIData, std::__1::default_delete<content::NavigationUIData> >, mojo::InterfacePtr<network::mojom::URLLoaderClient>,
mojo::InterfaceRequest<network::mojom::URLLoader>, content::ServiceWorkerNavigationHandleCore*, content::AppCacheNav
igationHandleCore*, unsigned int, content::GlobalRequestID const&) (this=0x3a62840bf180, resource_context=0x3a6284141840, request_context=
0x3a62842e6a20, upload_file_system_context=0x3a6284256800, info=..., navigation_ui_data=..., url_load
er_client=..., url_loader_request=..., service_worker_handle_core=0x3a62865bde60, appcache_handle_core=0x0, url_loader_options=27,
global_request_id=...) at ../../src/content/browser/loader/resource_dispatcher_host_impl.cc:1647
#8 0x00007ffffd395c29 in
content::NavigationURLLoaderImpl::URLLoaderRequestController::CreateNonNetworkServiceURLLoader(net::URLRequestContextGetter*,
storage::FileSystemContext*, content::ServiceWorkerNavigationHandleCore*, content::AppC
acheNavigationHandleCore*, scoped_refptr<content::SignedExchangePrefetchMetricRecorder>, network::ResourceRequest const&,
mojo::InterfaceRequest<network::mojom::URLLoader>, mojo::InterfacePtr<network::mojom::URLLoaderClient>) (this=0x3a628
57ec020, url_request_context_getter=0x3a62842b6c80, upload_file_system_context=0x3a6284256800, service_worker_navigation_handle_core=
0x3a62865bde60, appcache_handle_core=0x0, signed_exchange_prefetch_metric_recorder=..., url_loader=..., ur
l_loader_client=...) at ../../src/content/browser/loader/navigation_url_loader_impl.cc:543
#9 0x00007ffffd21b647 in content::SingleRequestURLLoaderFactory::HandlerState::HandleRequest(network::ResourceRequest const&,
mojo::InterfaceRequest<network::mojom::URLLoader>, mojo::InterfacePtr<network::mojom::URLLoaderClient>) (this=0x
3a6287b92b60, resource_request=..., loader=..., client=...) at ../../src/content/common/single_request_url_loader_factory.cc:38
#10 0x00007ffffd21ade7 in content::SingleRequestURLLoaderFactory::CreateLoaderAndStart(mojo::InterfaceRequest<network::mojom::URLLoader>, int,
int, unsigned int, network::ResourceRequest const&, mojo::InterfacePtr<network::mojom::URLLoader
Client>, net::MutableNetworkTrafficAnnotationTag const&) (this=0x3a628718d3e0, loader=..., routing_id=-1, request_id=0, options=0, request=...,
client=..., traffic_annotation=...) at ../../src/content/common/single_request_url_loader_facto
ry.cc:91
#11 0x00007ffffd21b76 in content::ThrottlingURLLoader::StartNow() (this=0x3a62854e3920)
at ../../src/content/common/throttling_url_loader.cc:377
#12 0x00007ffffd21f839 in content::ThrottlingURLLoader::Start(scoped_refptr<network::SharedURLLoaderFactory>, int, int, unsigned int,
network::ResourceRequest*, scoped_refptr<base::SingleThreadTaskRunner>) (this=0x3a62854e3920, factory=...
, routing_id=-1, request_id=0, options=0, url_request=0x3a62876483a0, task_runner=...) at ../../src/content/common/throttling_url_loader.cc:343
#13 0x00007ffffd21e301 in content::ThrottlingURLLoader::CreateLoaderAndStart(scoped_refptr<network::SharedURLLoaderFactory>, std::__
1::vector<std::__1::unique_ptr<content::URLLoaderThrottle, std::__1::default_delete<content::URLLoaderThrot
tle> >, std::__1::allocator<std::__1::unique_ptr<content::URLLoaderThrottle, std::__1::default_delete<content::URLLoaderThrottle> > >, int,
int, unsigned int, network::ResourceRequest*, network::mojom::URLLoaderClient*, net::NetworkTraff
icAnnotationTag const&, scoped_refptr<base::SingleThreadTaskRunner>) (factory=..., throttles=..., routing_id=-1, request_id=0, options=0,
url_request=0x3a62876483a0, client=0x3a62857ec020, traffic_annotation=..., task_runner=...) at ../../
src/content/common/throttling_url_loader.cc:172
#14 0x00007ffffd438c8d7 in
content::NavigationURLLoaderImpl::URLLoaderRequestController::StartWithoutNetworkService(net::URLRequestContextGetter*,
storage::FileSystemContext*, content::ServiceWorkerNavigationHandleCore*, content::AppCacheNa
vigationHandleCore*, scoped_refptr<content::SignedExchangePrefetchMetricRecorder>, std::__1::unique_ptr<content::NavigationRequestInfo, std::__
1::default_delete<content::NavigationRequestInfo> >, std::__1::unique_ptr<content::NavigationUIData, std::__
1::default_delete<content::NavigationUIData> >) (this=0x3a62857ec020, url_request_context_getter=0x3a62842b6c80, upload_file_system_context=
0x3a6284256800, service_worker_navigation_handle_core=0x3a62865bde60, appcache_handle_c
ore=0x0, signed_exchange_prefetch_metric_recorder=..., request_info=..., navigation_ui_data=...)
at ../../src/content/browser/loader/navigation_url_loader_impl.cc:628
运行于content_shell线程，从以下地方发起异步请求。
#15 content::NavigationURLLoaderImpl::NavigationURLLoaderImpl (this=0x3a62871f0fa0, resource_context=0x3a6284141840, storage_partition=
0x3a6284069c20, request_info=..., navigation_ui_data=..., service_worker_navigation_handle=0x3a6286d7e56
0, appcache_handle=0x0, delegate=0x3a6285cea720, initial_interceptors=...)
at ../../src/content/browser/loader/navigation_url_loader_impl.cc:1646
#16 0x00007ffffd43835a in content::NavigationURLLoader::Create(content::ResourceContext*, content::StoragePartition*, std::__
1::unique_ptr<content::NavigationRequestInfo, std::__1::default_delete<content::NavigationRequestInfo> >, std::__1
::unique_ptr<content::NavigationUIData, std::__1::default_delete<content::NavigationUIData> >, content::ServiceWorkerNavigationHandle*,
content::AppCacheNavigationHandle*, content::NavigationURLLoaderDelegate*) (resource_context=0x3a628414
1840, storage_partition=0x3a6284069c20, request_info=..., navigation_ui_data=..., service_worker_handle=0x3a6286d7e560, appcache_handle=0x0,
delegate=0x3a6285cea720) at ../../src/content/browser/loader/navigation_url_loader.cc:35
#17 0x00007ffffd40c28ea in content::NavigationRequest::OnStartChecksComplete(content::NavigationThrottle::ThrottleCheckResult) (this=
0x3a6285cea720, result=...) at ../../src/content/browser/frame_host/navigation_request.cc:1429
#18 0x00007ffffd40ae8c2 in content::NavigationHandleImpl::RunCompleteCallback(content::NavigationThrottle::ThrottleCheckResult) (this=
0x3a6286205320, result=...) at ../../src/content/browser/frame_host/navigation_handle_impl.cc:1322
#19 0x00007ffffd40a7e7c in content::NavigationHandleImpl::WillStartRequest(base::RepeatingCallback<void
```

```
(content::NavigationThrottle::ThrottleCheckResult)> const&)(this=0x3a6286205320, callback=...) at ../../src/content/browser/frame_host/navigation_handle_impl.cc:707
#13 0x00007ffff40bdd50 in content::NavigationRequest::BeginNavigation() (this=0x3a6285cea720)
at ../../src/content/browser/frame_host/navigation_request.cc:658
#14 0x00007ffff40d2e69 in content::NavigatorImpl::OnBeforeUnloadACK(content::FrameTreeNode*, bool, base::TimeTicks const&)(this=
0x3a62842b5910, frame_tree_node=0x3a6283b96020, proceed=true, proceed_time=...) at ../../src/content/browser/f
rame_host/navigator_impl.cc:574
#15 0x00007ffff40e278a in content::RenderFrameHostImpl::ProcessBeforeUnloadACKFromFrame(bool, bool, content::RenderFrameHostImpl*, bool,
base::TimeTicks const&, base::TimeTicks const&)(this=0x3a6284330220, proceed=true, treat_as_final_ack
=false, frame=0x3a6284330220, is_frame_being_destroyed=false, renderer_before_unload_start_time=..., renderer_before_unload_end_time=...)
at ../../src/content/browser/frame_host/render_frame_host_impl.cc:2261
#16 0x00007ffff40f46bc in content::RenderFrameHostImpl::OnBeforeUnloadACK(bool, bool, base::TimeTicks const&, base::TimeTicks const&)(
this=0x3a6284330220, proceed=true, treat_as_final_ack=false, renderer_before_unload_start_time=...,
renderer_before_unload_end_time=...) at ../../src/content/browser/frame_host/render_frame_host_impl.cc:2181
#17 0x00007ffff40e975d in content::RenderFrameHostImpl::OnBeforeUnloadACK(bool, base::TimeTicks const&, base::TimeTicks const&)(this=
0x3a6284330220, proceed=true, renderer_before_unload_start_time=..., renderer_before_unload_end_time=...)
at ../../src/content/browser/frame_host/render_frame_host_impl.cc:2154
#20 0x00007ffff413d144 in IPC::DispatchToMethod<content::RenderFrameHostImpl, void (content::RenderFrameHostImpl::*)(bool, base::TimeTicks
const&, base::TimeTicks const&), void, std::__1::tuple<bool, base::TimeTicks, base::TimeTicks>>>(con
tent::RenderFrameHostImpl*, void (content::RenderFrameHostImpl::*)(bool, base::TimeTicks const&, base::TimeTicks const&), void*, std::__
1::tuple<bool, base::TimeTicks, base::TimeTicks>&&)(obj=0x3a6284330220, method=(void (content::RenderF
rameHostImpl::*)(content::RenderFrameHostImpl * const, bool, const base::TimeTicks &, const base::TimeTicks &)) 0x7ffff40e9720
<content::RenderFrameHostImpl::OnBeforeUnloadACK(bool, base::TimeTicks const&, base::TimeTicks const&)>, tuple=
..) at ../../src/ipc/ipc_message_templates.h:51
#21 0x00007ffff4116e8a in IPC::MessageT<FrameHostMsg_BeforeUnload_ACK_Meta, std::__1::tuple<bool, base::TimeTicks, base::TimeTicks>,
void>::Dispatch<content::RenderFrameHostImpl, content::RenderFrameHostImpl, void, void (content::RenderFra
meHostImpl::*)(bool, base::TimeTicks const&, base::TimeTicks const&)>>(IPC::Message const*, content::RenderFrameHostImpl*,
content::RenderFrameHostImpl*, void*, void (content::RenderFrameHostImpl::*)(bool, base::TimeTicks const&, base::Time
Ticks const&))(msg=0x3a6287cd0658, obj=0x3a6284330220, sender=0x3a6284330220, parameter=0x0, func=(void (content::RenderFrameHostImpl::*)(
content::RenderFrameHostImpl * const, bool, const base::TimeTicks &, const base::TimeTicks &)) 0x7fff
ff40e9720 <content::RenderFrameHostImpl::OnBeforeUnloadACK(bool, base::TimeTicks const&, base::TimeTicks const&)>))
at ../../src/ipc/ipc_message_templates.h:146
#22 0x00007ffff40e760b in content::RenderFrameHostImpl::OnMessageReceived(IPC::Message const&)(this=0x3a6284330220, msg=...)
at ../../src/content/browser/frame_host/render_frame_host_impl.cc:1231
#23 0x00007ffff4729b1 in content::RenderProcessHostImpl::OnMessageReceived(IPC::Message const&)(this=0x3a6283b43020, msg=...)
at ../../src/content/browser/renderer_host/render_process_host_impl.cc:3373
#24 0x00007ffff7f955c5 in IPC::ChannelProxy::Context::OnDispatchMessage(IPC::Message const&)(this=0x3a6283f76020, message=...)
at ../../src/ipc/ipc_channel_proxy.cc:320
```

运行在“Chrome_IOThread”线程，在这里通过IPC发送读取的资源数据到其它线程。这里并没有直接发送payload，而是通过mojo将datapipe发送出去，发过去之后可以一直使用该datapipe进行数据读取。mojo的DataPipe机制是使用共享内存的方式实现的。

```
#0 0x00007ffff2b45928 in network::mojom::URLLoaderClientProxy::OnStartLoadingResponseBody_Message::Build(bool, bool, bool,
mojom::ScopedHandleBase<mojom::DataPipeConsumerHandle>)(serialize=false, expects_response=false, is_sync=false, param
_body=...) at gen/services/network/public/mojom/url_loader.mojom.cc:1506
#1 0x00007ffff2b3c44c in
network::mojom::URLLoaderClientProxy::OnStartLoadingResponseBody(mojom::ScopedHandleBase<mojom::DataPipeConsumerHandle>)(this=0x3a6287cc37a0,
in_body=...) at gen/services/network/public/mojom/url_loader.mojom.cc:15
73
#2 0x00007ffff4380513 in content::MojoAsyncResourceHandler::OnReadCompleted(int, std::__1::unique_ptr<content::ResourceController, std::__
1::default_delete<content::ResourceController>>)(this=0x3a62866f6da0, bytes_read=32768, controller
=...) at ../../src/content/browser/loader/mojo_async_resource_handler.cc:361
#3 0x00007ffff436a057 in content::InterceptingResourceHandler::OnReadCompleted(int, std::__1::unique_ptr<content::ResourceController, std::__
1::default_delete<content::ResourceController>>)(this=0x3a6286698520, bytes_read=32768, control
ler=...) at ../../src/content/browser/loader/intercepting_resource_handler.cc:137
#4 0x00007ffff43714d4 in content::LayeredResourceHandler::OnReadCompleted(int, std::__1::unique_ptr<content::ResourceController, std::__
1::default_delete<content::ResourceController>>)(this=0x3a6283f45ca0, bytes_read=32768, controller=
..) at ../../src/content/browser/loader/layered_resource_handler.cc:62
#5 0x00007ffff435b919 in content::CrossSiteDocumentResourceHandler::OnReadCompleted(int, std::__1::unique_ptr<content::ResourceController,
std::__1::default_delete<content::ResourceController>>)(this=0x3a6287040020, bytes_read=32768, co
ntroller=...) at ../../src/content/browser/loader/cross_site_document_resource_handler.cc:358
#6 0x00007ffff4377607 in content::MimeSniffingResourceHandler::OnReadCompleted(int, std::__1::unique_ptr<content::ResourceController, std::__
1::default_delete<content::ResourceController>>)(this=0x3a6287e98800, bytes_read=32768, control
ler=...) at ../../src/content/browser/loader/mime_sniffing_resource_handler.cc:232
#7 0x00007ffff43714d4 in content::LayeredResourceHandler::OnReadCompleted(int, std::__1::unique_ptr<content::ResourceController, std::__
1::default_delete<content::ResourceController>>)(this=0x3a62868bc2a0, bytes_read=32768, controller=
..) at ../../src/content/browser/loader/layered_resource_handler.cc:62
#8 0x00007ffff43e5698 in content::ResourceLoader::CompleteRead(int)(this=0x3a6284d33220, bytes_read=32768)
at ../../src/content/browser/loader/resource_loader.cc:807
#9 0x00007ffff43e5021 in content::ResourceLoader::OnReadCompleted(net::URLRequest*, int)(this=0x3a6284d33220, unused=0x3a6287c3e020,
bytes_read=32768) at ../../src/content/browser/loader/resource_loader.cc:485
#10 0x00007ffffd3bf0fb in net::URLRequest::NotifyReadCompleted(int)(this=0x3a6287c3e020, bytes_read=32768)
at ../../src/net/url_request/url_request.cc:1126
#11 0x00007ffffd3f84c7 in net::URLRequestJob::SourceStreamReadComplete(bool, int)(this=0x3a6284ab3ba0, synchronous=false, result=32768)
at ../../src/net/url_request/url_request_job.cc:660
```

运行在“Chrome_InProcRe”线程，通过URLLoaderClient这个mojo接口接收数据，然后通过blink::WebURLLoaderClient将数据传入blink，blink会根据配置选择是否缓存数据在blink中，然后通过blink::WebAssociatedURLLoaderClient通知其它Client已经收到数据，content在收到通知后，通过ResourceMultiBufferDataProvider将数据存储存在DataMap中。

```
#0 0x00007ffffd07493b5 in media::MultiBuffer::OnDataProviderEvent(media::MultiBuffer::DataProvider*)(this=0x3a6284718a48, provider_tmp=
0x3a62847268e0) at ../../src/media/blink/multibuffer.cc:393
#1 0x00007ffffd0779913 in media::ResourceMultiBufferDataProvider::DidReceiveData(char const*, int)(this=0x3a62847268e0, data=0x7ffff9ce8d000
"sc", data_length=0) at ../../src/media/blink/resource_multibuffer_data_provider.cc:419
#2 0x00007ffffd2a82130 in blink::WebAssociatedURLLoaderImpl::ClientAdapter::DidReceiveData(char const*, unsigned int)(this=0x3a62870541a0,
data=0x7ffff9ce7d000 "\366\037H\325u\336*T\343\254\063u\332\v\354\177\002\370Nlj\262\3611Z\265\212
\345\367\067\005\277\353\373M\265\206.7\206\240\265\226\353b\253\307=\321\324s\005\351s\273", data_length=65536)
at ../../src/third_party/blink/renderer/core/exported/web_associated_url_loader_impl.cc:278
#3 0x00007ffffd34ef091 in blink::ThreadableLoader::DataReceived(blink::Resource*, char const*, unsigned long)(this=0x2f9772801840, resource=
0x2f9772801fb8, data=0x7ffff9ce7d000 "\366\037H\325u\336*T\343\254\063u\332\v\354\177\002\370Nlj\
```

```
262\3611Z\265\212\345\367\067\005\277:\353\373M\265\206.7\206\240\265\226\353b\253\307=\321\324s>\005\351s\273", data_length=65536)
at ../../src/third_party/blink/renderer/core/loader/threadable_loader.cc:939
#4 0x00007ffffdfcd3c6f in blink::Resource::AppendData(char const*, unsigned long) (this=0x2f9772801fb8, data=0x7fff9ce7d000 "\366\037H\325u\336
*T\343\254\063u\332\v\354\177\F\002\370NLj\262\3611Z\265\212\345\367\067\005\277:\353\373M\265\2
06.7\206\240\265\226\353b\253\307=\321\324s>\005\351s\273", length=65536)
at ../../src/third_party/blink/renderer/platform/loader/fetch/resource.cc:304
#5 0x00007ffffdfcd8618 in blink::RawResource::AppendData(char const*, unsigned long) (this=0x2f9772801fb8, data=0x7fff9ce7d000 "\366\037H\325u
\336*T\343\254\063u\332\v\354\177\F\002\370NLj\262\3611Z\265\212\345\367\067\005\277:\353\373M\26
5\206.7\206\240\265\226\353b\253\307=\321\324s>\005\351s\273", length=65536)
at ../../src/third_party/blink/renderer/platform/loader/fetch/raw_resource.cc:144
#6 0x00007ffffdfcd214bd in blink::ResourceLoader::DidReceiveData(char const*, int) (this=0x2f9772802728, data=0x7fff9ce7d000 "\366\037H\325u\336
*T\343\254\063u\332\v\354\177\F\002\370NLj\262\3611Z\265\212\345\367\067\005\277:\353\373M\265\2
06.7\206\240\265\226\353b\253\307=\321\324s>\005\351s\273", length=65536)
at ../../src/third_party/blink/renderer/platform/loader/fetch/resource_loader.cc:915
#7 0x00007ffff5686519 in content::WebURLLoaderImpl::Context::OnReceivedData(std::__1::unique_ptr<content::RequestPeer::ReceivedData, std::__
1::default_delete<content::RequestPeer::ReceivedData> >) (this=0x3a62876321a0, data=...) at ../../
src/content/renderer/loader/web_url_loader_impl.cc:928
#8 0x00007ffff56877e1 in content::WebURLLoaderImpl::RequestPeerImpl::OnReceivedData(std::__1::unique_ptr<content::RequestPeer::ReceivedData,
std::__1::default_delete<content::RequestPeer::ReceivedData> >) (this=0x3a6283edded0, data=...)
at ../../src/content/renderer/loader/web_url_loader_impl.cc:1124
#9 0x00007ffff5678f59 in content::URLResponseBodyConsumer::OnReadable(unsigned int) (this=0x3a62849d3a20, unused=0)
at ../../src/content/renderer/loader/url_response_body_consumer.cc:149
在这里通过由URLLoaderClient::OnStartLoadingResponseBody()这个mojo接口传过来mojo::ScopedDataPipeConsumerHandle读取数据，这个类包装了用于传输数据的datapipe。
#10 0x00007ffff56762a6 in content::URLLoaderClientImpl::OnStartLoadingResponseBody(mojo::ScopedHandleBase<mojo::DataPipeConsumerHandle>) (this=
0x3a6283b94020, body=...) at ../../src/content/renderer/loader/url_loader_client_impl.cc:319
#11 0x00007ffff2d285d2 in content::ThrottlingURLLoader::OnStartLoadingResponseBody(mojo::ScopedHandleBase<mojo::DataPipeConsumerHandle>) (this=
0x3a62849d3f20, body=...) at ../../src/content/common/throttling_url_loader.cc:563
#12 0x00007ffff2b47e90 in network::mojom::URLLoaderClientProxy::OnStartLoadingResponseBody::Message::Dispatch(network::mojom::URLLoaderClient*)
(this=0x3a6287bac720, impl=0x3a62849d3f20) at gen/services/network/public/mojom/url_loader.mojom.cc:1539
#13 0x00007ffff2b3de36 in network::mojom::URLLoaderClientStubDispatch::Accept(network::mojom::URLLoaderClient*, mojo::Message*) (impl=
0x3a62849d3f20, message=0x7fffa265d240) at gen/services/network/public/mojom/url_loader.mojom.cc:2096
```

运行在“TaskSchedulerFo”线程中，读取数据到ffmpeg分配的内存中，该数据是已经从ResourceMultiBufferDataProvider中读取之后存储在DataMap中的数据，这里只是做一个拷贝。ResourceMultiBufferDataProvider继承自blink::WebAssociatedURLLoaderClient接口。

```
#0 0x00007ffffd076b7c9 in media::MultibufferDataSource::Read(long, int, unsigned char*, base::RepeatingCallback<void (int)> const&) (this=
0x3a6283b94020, position=0, size=32768, data=0x3a6287bb0040 '\253' <repeats 200 times>..., read_cb=..
.) at ../../src/media/blink/multibuffer_data_source.cc:401
#1 0x00007ffffea57a82 in media::BlockingUrlProtocol::Read(int, unsigned char*) (this=0x3a62871047b0, size=32768, data=0x3a6287bb0040 '\253'
<repeats 200 times>...) at ../../src/media/filters/blocking_url_protocol.cc:58
#2 0x00007ffffea7c1a8 in media::AVIOReadOperation(void*, unsigned char*, int) (opaque=0x3a62871047b0, buf=0x3a6287bb0040 '\253' <repeats 200
times>..., buf_size=32768) at ../../src/media/filters/ffmpeg_glue.cc:23
#3 0x00007ffffd7990296 in read_packet_wrapper(s=<optimized out>, buf=<optimized out>, size=<optimized out>)
at ../../src/third_party/ffmpeg/libavformat/aviobuf.c:535
#4 0x00007ffffd7990296 in fill_buffer(s=0x3a6287bd1e40) at ../../src/third_party/ffmpeg/libavformat/aviobuf.c:585
#5 0x00007ffffd7990bcb in avio_read(s=0x3a6287bd1e40, buf=0x3a6287bbc620 '\253' <repeats 200 times>..., size=2048)
at ../../src/third_party/ffmpeg/libavformat/aviobuf.c:678
#6 0x00007ffffd79961a8 in av_probe_input_buffer2(pb=0x3a6287bd1e40, fmt=0x3a6287b7d848, filename=<optimized out>, logctx=0x3a6287b7d840,
offset=0, max_probe_size=1048576) at ../../src/third_party/ffmpeg/libavformat/format.c:262
#7 0x00007ffffd798060d in init_input(s=<optimized out>, filename=<optimized out>, options=<optimized out>)
at ../../src/third_party/ffmpeg/libavformat/utils.c:434
#8 0x00007ffffd798060d in avformat_open_input(ps=0x3a6283e01508, filename=0x0, fmt=<optimized out>, options=0x0)
at ../../src/third_party/ffmpeg/libavformat/utils.c:573
#9 0x00007ffffea7c569 in media::FFmpegGlue::OpenContext(bool) (this=0x3a6283e01500, is_local_file=true)
at ../../src/media/filters/ffmpeg_glue.cc:110
```

FFmpegGlue的构造函数中会初始化AVIO接口。

在FFmpegDemuxer初始化的时候触发该异步调用，这属于Demuxer初始化的一步，这一步会读取文件前一部分用于获取当前视频文件的容器类型等信息。FFmpegGlue用于和ffmpeg对接，因为chromium通过ffmpeg的AVIO的接口使用ffmpeg，包括Context创建，回调注册（Read和Seek回调），buffer的创建等。FFmpegDemuxer中还创建了BlockingUrlProcol对象，该对象实现了对读取逻辑的控制，具体的读取逻辑通过调用DataSource实现。

运行在“Chrome_InProcRe”线程，创建FFmpegDemuxer，然后Start Pipeline，创建RendererImpl（GpuVideoAcceleratorFactories，AudioRendererImpl，GpuMemoryBufferVideoFramePool，VideoRendererImpl）。Start Pipeline的时候会向SerialRunner中依次放入InitializeDemuxer，ReportMetadata，CreateRender，InitializeRenderer这些回调，当这个流程执行完之后会调用CompleteSeek，它会调用RenderImpl::StartPlayingFrom开始进行数据读取及解码流程，最终输出一个VideoFrame对象，该VF被存入VideoFrameAlgorithm中。CreateRender的逻辑在新版本进行了改动，在新版本中是在SerialRunner中创建Renderer的，在旧版本中并不是。

```
#0 0x00007ffffee858401 in media::PipelineImpl::Start(media::Pipeline::StartType, media::Demuxer*, std::__1::unique_ptr<media::Renderer, std::__1:
default_delete<media::Renderer> >, media::Pipeline::Client*, base::RepeatingCallback<void (me
dia::PipelineStatus)> const&) (this=0x3a628608d3e0, start_type=media::Pipeline::StartType::kNormal, demuxer=0x3a628704e020, renderer=...,
client=0x3a62860c0830, seek_cb=...) at ../../src/media/base/pipeline_impl.cc:1016
#1 0x00007ffffee9bb284 in media::PipelineController::Start(media::Pipeline::StartType, media::Demuxer*, media::Pipeline::Client*, bool, bool)
(this=0x3a62860c0898, start_type=media::Pipeline::StartType::kNormal, demuxer=0x3a628704e020, cli
ent=0x3a62860c0830, is_streaming=false, is_static=true) at ../../src/media/filters/pipeline_controller.cc:59
#2 0x00007ffffd07d7a37 in media::WebMediaPlayerImpl::StartPipeline() (this=0x3a62860c0820) at ../../src/media/blink/webmediaplayer_impl.cc:2554
#3 0x00007ffffd07d7faa in media::WebMediaPlayerImpl::DataSourceInitialized(bool) (this=0x3a62860c0820, success=true)
at ../../src/media/blink/webmediaplayer_impl.cc:2349
```

从下面注册以上回调，blink解析资源来源，创建用于读取资源的MultibufferDataSource

```
#0 0x00007ffffd07d4cb4 in media::WebMediaPlayerImpl::DoLoad(blink::WebMediaPlayer::LoadType, blink::WebURL const&,
blink::WebMediaPlayer::CORSMode) (this=0x3a62860c0820, load_type=blink::WebMediaPlayer::kLoadTypeURL, url=..., cors_mode=bli
nk::WebMediaPlayer::kCORSModeUnspecified) at ../../src/media/blink/webmediaplayer_impl.cc:631
...base...
```

```
#6 0x00007fffff2b72743 in content::ContentRendererClient::DeferMediaLoad(content::RenderFrame*, bool, base::OnceCallback<void (>)>) (this=
0x3a6283a88bb0, render_frame=0x3a628431d420, has_played_media_before=false, closure=Python Exception <
class 'gdb.MemoryError'> Cannot access memory at address 0x38df5b4754d5fd00:
...base...
```

```
#12 0x00007ffffd07d43ce in media::WebMediaPlayerImpl::Load(blink::WebMediaPlayer::LoadType, blink::WebMediaPlayerSource const&,
blink::WebMediaPlayer::CORSMode) (this=0x3a62860c0820, load_type=blink::WebMediaPlayer::kLoadTypeURL, source=...
, cors_mode=blink::WebMediaPlayer::kCORSModeUnspecified) at ../../src/media/blink/webmediaplayer_impl.cc:428
#13 0x00007ffffe2e77859 in blink::HTMLMediaElement::StartPlayerLoad() (this=0x3571c9be3910)
at ../../src/third_party/blink/renderer/core/html/media/html_media_element.cc:1319
```

```
#14 0x00007ffffe2e74c31 in blink::HTMLMediaElement::LoadResource(blink::WebMediaPlayerSource const&, WTF::String const&) (this=0x3571c9be3910, source=..., content_type=...) at ../../src/third_party/blink/renderer/core/html/media/html_media_element.cc:1235
#15 0x00007ffffe2e71bde in blink::HTMLMediaElement::LoadNextSourceChild() (this=0x3571c9be3910) at ../../src/third_party/blink/renderer/core/html/media/html_media_element.cc:1161
#16 0x00007ffffe2e73c7d in blink::HTMLMediaElement::SelectMediaResource() (this=0x3571c9be3910) at ../../src/third_party/blink/renderer/core/html/media/html_media_element.cc:1104
#17 0x00007ffffe2e71d4a in blink::HTMLMediaElement::LoadInternal() (this=0x3571c9be3910) at ../../src/third_party/blink/renderer/core/html/media/html_media_element.cc:1033
#18 0x00007ffffe2e6ec41 in blink::HTMLMediaElement::LoadTimerFired(blink::TimerBase*) (this=0x3571c9be3910) at ../../src/third_party/blink/renderer/core/html/media/html_media_element.cc:793
#19 0x00007ffffe2178447 in blink::TaskRunnerTimer<blink::V8GCForContextDispose>::Fired() (this=0x3571c9be3a10) at ../../src/third_party/blink/renderer/platform/timer.h:141
#20 0x00007ffffdfc1acba in blink::TimerBase::RunInternal() (this=0x3571c9be3a10) at ../../src/third_party/blink/renderer/platform/timer.cc:156
```

运行于Media线程，初始化FfmpegDemuxer：创建FfmpegGlue，初始化AVIO接口，解析media的metadata数据，获取视频流中的媒体流，初始化DemuxerStream。

```
#0 0x00007ffffeeaa65d9a in media::FfmpegDemuxer::Initialize(media::DemuxerHost*, base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a6287d75160, host=0x3a6286d7c020, init_cb=...) at ../../src/media/filters/ffmpeg_demuxer.cc:979
#1 0x00007ffffee84ba6f in media::PipelineImpl::RendererWrapper::InitializeDemuxer(base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a6286d7c020, done_cb=...) at ../../src/media/base/pipeline_impl.cc:881
#7 0x00007ffffee86e2e5 in media::SerialRunner::RunNextInSeries(media::PipelineStatus) (this=0x3a6284dc0720, last_status=media::PIPELINE_OK) at ../../src/media/base/serial_runner.cc:109
#8 0x00007ffffee86e06b in media::SerialRunner::SerialRunner(media::SerialRunner::Queue const&, base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a6284dc0720, bound_fns=..., done_cb=...) at ../../src/media/base/serial_runner.cc:86
#9 0x00007ffffee86e3fb in media::SerialRunner::Run(media::SerialRunner::Queue const&, base::RepeatingCallback<void (media::PipelineStatus)> const&) (bound_fns=..., done_cb=...) at ../../src/media/base/serial_runner.cc:95
#10 0x00007ffffee84b32d in media::PipelineImpl::RendererWrapper::Start(media::Pipeline::StartType, media::Demuxer*, std::__1::unique_ptr<media::Renderer, std::__1::default_delete<media::Renderer>>, base::WeakPtr<media::PipelineImpl>) (this=0x3a6286d7c020, start_type=media::Pipeline::StartType::kNormal, demuxer=0x3a6287d75160, renderer=..., weak_pipeline=...) at ../../src/media/base/pipeline_impl.cc:263
```

运行于Media线程，创建并设置音频解码器。

```
#0 0x00007ffffeeb2d52a in media::DefaultRendererFactory::CreateAudioDecoders(scoped_refptr<base::SingleThreadTaskRunner> const&) (this=0x3a62879ed9e0, media_task_runner=...) at ../../src/media/renderers/default_renderer_factory.cc:36
#7 0x00007ffffee968a12 in media::DecoderStream<media::DemuxerStream::Type>1::SelectDecoder() (this=0x3a6284710020) at ../../src/media/filters/decoder_stream.cc:322
#8 0x00007ffffee968926 in media::DecoderStream<media::DemuxerStream::Type>1::Initialize(media::DemuxerStream*, base::OnceCallback<void (bool)>, media::CdmContext*, base::RepeatingCallback<void (media::PipelineStatistics const&)>, base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a6284710020, stream=0x3a62870622e0, init_cb=..., cdm_context=0x0, statistics_cb=..., waiting_for_decryption_key_cb=...) at ../../src/media/filters/decoder_stream.cc:173
#9 0x00007ffffeeb224d6 in media::AudioRendererImpl::Initialize(media::DemuxerStream*, media::CdmContext*, media::RendererClient*, base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a6284946220, stream=0x3a62870622e0, cdm_context=0x0, client=0x3a6287f088e0, init_cb=...) at ../../src/media/renderers/audio_renderer_impl.cc:538
#10 0x00007ffffeeb3cc30 in media::RendererImpl::InitializeAudioRenderer() (this=0x3a6287646c60) at ../../src/media/renderers/renderer_impl.cc:382
#11 0x00007ffffeeb3c15f in media::RendererImpl::Initialize(media::MediaResource*, media::RendererClient*, base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a6287646c60, media_resource=0x3a628704e020, client=0x3a62850348e8, init_cb=...) at ../../src/media/renderers/renderer_impl.cc:164
#12 0x00007ffffee84c8c7 in media::PipelineImpl::RendererWrapper::InitializeRenderer(base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a62850348e0, done_cb=...) at ../../src/media/base/pipeline_impl.cc:911
#18 0x00007ffffee86e2e5 in media::SerialRunner::RunNextInSeries(media::PipelineStatus) (this=0x3a62871fac20, last_status=media::PIPELINE_OK) at ../../src/media/base/serial_runner.cc:109
```

运行在Media线程，在音频解码器设置成功后创建，初始化，并设置视频解码器，在DecoderSelector中依次尝试初始化解码器，第一个初始化成功的会被选中。

```
#0 0x00007ffffeeb2a51a in media::DefaultDecoderFactory::CreateVideoDecoders(scoped_refptr<base::SingleThreadTaskRunner>, media::GpuVideoAcceleratorFactories*, media::MediaLog*, base::RepeatingCallback<void (bool, base::RepeatingCallback<void (media::OverlayInfo const&)> const&)> const&, gfx::ColorSpace const&, std::__1::vector<std::__1::unique_ptr<media::VideoDecoder, std::__1::default_delete<media::VideoDecoder>>, std::__1::allocator<std::__1::unique_ptr<media::VideoDecoder, std::__1::default_delete<media::VideoDecoder>>>> >> >) (this=0x3a6284626890, task_runner=..., gpu_factories=0x3a628736e1e0, media_log=0x3a6284706c20, request_overlay_info_cb=..., target_color_space=..., video_decoders=0x7ffff9bb2cd68) at ../../src/media/renderers/default_decoder_factory.cc:96
```

在这里创建FfmpegVideoDecoder等多种decoder，在windows上会默认创建GpuVideoDecoder。

```
#0 0x00007ffffeeb2d679 in media::DefaultRendererFactory::CreateVideoDecoders(scoped_refptr<base::SingleThreadTaskRunner> const&, base::RepeatingCallback<void (bool, base::RepeatingCallback<void (media::OverlayInfo const&)> const&)> const&, gfx::ColorSpace const&, media::GpuVideoAcceleratorFactories*) (this=0x3a62879ed9e0, media_task_runner=..., request_overlay_info_cb=..., target_color_space=..., gpu_factories=0x3a628736e1e0) at ../../src/media/renderers/default_renderer_factory.cc:50
#6 0x00007ffffee955cef in media::DecoderSelector<media::DemuxerStream::Type>2::SelectDecoder(base::OnceCallback<void (std::__1::unique_ptr<media::VideoDecoder, std::__1::default_delete<media::VideoDecoder>>, std::__1::unique_ptr<media::DecryptedDemuxerStream, std::__1::default_delete<media::DecryptedDemuxerStream>>>, base::RepeatingCallback<void (scoped_refptr<media::VideoFrame> const&)>)> (this=0x3a628482b798, select_decoder_cb=..., output_cb=...) at ../../src/media/filters/decoder_selector.cc:90
```

在这里调用各种decoder的Initialize方法进行初始化，第一个初始化成功的decoder会被选用。在FfmpegVideoDecoder的初始化中会调用avcodec_find_decoder和avcodec_open2初始化codecs。

```
#7 0x00007ffffee95e192 in media::DecoderStream<media::DemuxerStream::Type>2::SelectDecoder() (this=0x3a628482b720) at ../../src/media/filters/decoder_stream.cc:322
#8 0x00007ffffee95e0a6 in media::DecoderStream<media::DemuxerStream::Type>2::Initialize(media::DemuxerStream*, base::OnceCallback<void (bool)>, media::CdmContext*, base::RepeatingCallback<void (media::PipelineStatistics const&)>, base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a628482b720, stream=0x3a6284faeb20, init_cb=..., cdm_context=0x0, statistics_cb=..., waiting_for_decryption_key_cb=...) at ../../src/media/filters/decoder_stream.cc:173
#9 0x00007ffffeeb4be0 in media::VideoRendererImpl::Initialize(media::DemuxerStream*, media::CdmContext*, media::RendererClient*, base::RepeatingCallback<bool (std::__1::vector<base::TimeDelta, std::__1::allocator<base::TimeDelta>> const& >, std::__1::vector<base::TimeTicks, std::__1::allocator<base::TimeTicks>> const&, base::RepeatingCallback<void (media::PipelineStatus)> const&) (this=0x3a62876537a0, stream=0x3a6284faeb20, cdm_context=0x0, client=0x3a6287f27a20, wall_c
```



```
lock_time_cb=..., init_cb=...) at ../../src/media/renderers/video_renderer_impl.cc:257
#10 0x00007ffffeb40049 in media::RendererImpl::InitializeVideoRenderer() (this=0x3a6287646c60)
at ../../src/media/renderers/renderer_impl.cc:431
#11 0x00007ffffeb3f935 in media::RendererImpl::OnAudioRendererInitializeDone(media::PipelineStatus) (this=0x3a6287646c60,
status=media::PIPELINE_OK) at ../../src/media/renderers/renderer_impl.cc:404
```

运行在Media线程中，CompleteSeek会在上面的SerialRunner执行完所有流程之后触发，它调用VideoRendererImpl::StartPlayingFrom开始播放，调用VideoRendererImpl::AttemptRead_Locked()获取VideoFrame，如果algorithm中的VF队列有足够的数

据，会通过AudioRenderer和VideoRenderer触发渲染逻辑，如果队列满了，则返回，否则调用DecoderStream::Read()获取VideoFrame并放入队列中，DecoderStream从FfmpegDemuxerStream::Read()获取DecoderBuffer，FfmpegDemuxerStream从内部的buffer队列获取DecoderBuffer。如果该队列为空，则会在FfmpegDemuxer中异步调用Ffmpeg的av_read_frame方法读取一个AVPacket，然后将该AVPacket存入FfmpegDemuxerStream中，然后将数据拷贝到DecoderBuffer中并放入DS中的队列。av_read_frame会通过FfmpegGlue中初始化的AVIO接口来读取数据。

```
#0 0x00007ffffea699e6 in media::FfmpegDemuxer::ReadFrameIfNeeded() (this=0x3a628704e020) at ../../src/media/filters/ffmpeg_demuxer.cc:1797
#1 0x00007ffffea64902 in media::FfmpegDemuxer::NotifyCapacityAvailable() (this=0x3a628704e020)
at ../../src/media/filters/ffmpeg_demuxer.cc:1177
#2 0x00007ffffea62933 in media::FfmpegDemuxerStream::SatisfyPendingRead() (this=0x3a62870622e0)
at ../../src/media/filters/ffmpeg_demuxer.cc:869
#0 0x00007ffffea63257 in media::FfmpegDemuxerStream::Read(base::RepeatingCallback<void (media::DemuxerStream::Status,
scoped_refptr<media::DecoderBuffer>>) const& (this=0x3a62870622e0, read_cb=...) at ../../src/media/filters/ffmpeg_demux
er.cc:729
#0 0x00007ffffe969f54 in media::DecoderStream<media::DemuxerStream::Type>1::ReadFromDemuxerStream() (this=0x3a6284710020)
at ../../src/media/filters/decoder_stream.cc:651
#1 0x00007ffffe969235 in media::DecoderStream<media::DemuxerStream::Type>1::Read(base::OnceCallback<void
(media::DecoderStream<media::DemuxerStream::Type>1::Status, scoped_refptr<media::AudioBuffer> const&)>) (this=0x3a6284710020, rea
d_cb=...) at ../../src/media/filters/decoder_stream.cc:211
#2 0x00007ffffeb2026d in media::AudioRendererImpl::AttemptRead_Locked() (this=0x3a6284946220)
at ../../src/media/renderers/audio_renderer_impl.cc:853
#3 0x00007ffffeb20075 in media::AudioRendererImpl::StartPlaying() (this=0x3a6284946220)
at ../../src/media/renderers/audio_renderer_impl.cc:339
#4 0x00007ffffeb3de74 in media::RendererImpl::StartPlayingFrom(base::TimeDelta) (this=0x3a6287646c60, time=...)
at ../../src/media/renderers/renderer_impl.cc:233
#5 0x00007ffffe84ce4f in media::PipelineImpl::RendererWrapper::CompleteSeek(base::TimeDelta, media::PipelineStatus) (this=0x3a62850348e0,
seek_time=..., status=media::PIPELINE_OK) at ../../src/media/base/pipeline_impl.cc:833
#11 0x00007ffffe86e22c in media::SerialRunner::RunNextInSeries(media::PipelineStatus) (this=0x3a62871fac20, last_status=media::PIPELINE_OK)
at ../../src/media/base/serial_runner.cc:104
```