۱) :

HTTP/2چیست و چه تفاوت‌هایی با HTTP/1 دارا است؟

بیش از دو دهه است که از استاندارد شدن پروتکل HTTP/1.1 می‌گذرد و این در حالی است که HTTP/2آخرین مرحلهٔ تکاملی این پروتکل انتقال ابرمتن است که در نهایت این تکنولوژی جدید جایگزین HTTP/1.1 خواهد شد HTTPپروتکلی شبکه‌ای است که کاربرد آن ارسال ریکوئست و دریافت ریسپانس در شبکهٔ جهانی وب است .نسخه HTTP/2 در مقایسه با نسخه قدیمی‌تر این پروتکل دارای مزایای بسیاری است که در این مقاله قصد داریم به بررسی آن‌ها بپردازیم .

HTTP/2با HTTP/1.1 چه تفاوت‌هایی دارا است؟

واقعیت امر آن است که نام اصلی دومین نسخه از پروتکل اچ‌تی‌تی‌پی HTTP/2 است که ایده آن برگرفته از فناوری SPDY می‌باشد ، لازم به ذکر است که پروتکل SPDY در سال ۲۰۰۹ توسط گوگل ابداع شد.

HTTP/1.1 دارای یکسری مشکلات است و این در حالی است که این پروتکل در زمانی ساخته شد که حجم صفحات وب به ندرت از ۱۰۰ کیلو بایت تجاوز می‌کرد، زبان CSS هنوز فراگیر نشده بود اما زبان JavaScript تازه پایش به وب باز شده بود ولی صرفاً برای یکسری اِفکت ساده و یا وَلیدیشن فرم استفاده می‌شد. زمانی که این پروتکل عرضه شد، پهنای باند بالا، فناوری ای‌جکس، وب اپلیکیشن‌های تک‌صفحه‌ای و فریمورک‌های سمت کاربر هنوز وجود خارجی پیدا نکرده بودند و در واقع تحت هیچ عنوان امکان ارسال بیش از چند ریکوئست به سرور برای اجرای کامل یک صفحهٔ وب در مرورگر کاربران وجود نداشت!

اهداف اصلی شکل‌گیری HTTP/2

با این تفاسیر، اهداف اصلی پیاده‌سازی HTTP/2 **کاهش دادن زمان لود صفحات** وب بود؛ گرچه توضیح پیرامون نحوه پیاده‌سازی دومین نسخه از این پروتکل بسیار فنی و پیچیده می‌شود، اما در ادامه سعی می‌کنیم به زبانی گویا و ساده این مسئله را تشریح کنیم.

1. **نسخه HTTP/2 حاوی داده‌های باینری (دودویی) است** HTTP/1.1 : از داده‌های متنی استفاده می‌کند و این در حالی است که داده‌های متنی به طور کلی در سراسر شبکه از بازدهی کمتری نسبت به داده‌های باینری برخوردارند.

2. **هِدِرهای HTTP/2 فشرده شده هستند** : به طور کلی منظور از Header اطلاعاتی است که در پاسخ به یک ریکوئست ارسال می‌شود که شامل دیتا، مبدا، نوع، حجم، مدت زمان کَش و موارد

دیگر است. برخلاف HTTP/1.1 ، این داده‌ها در نسخه HTTP/2 فشرده‌سازی می‌شوند تا پرفورمنس ارتقاء یابد.

3. نسخه HTTP/2 اصطلاحاً Asynchronous است :در HTTP/1.1 ، سرور باید به همان ترتیبی که ریکوئست‌ها را دریافت کرده است، ریسپانس‌ها را ارسال کند اما نسخهٔ HTTP/2 اصطلاحاً Asynchronousاست؛ بنابراین پاسخ‌های سریع‌تر و در عین حال با حجم کمتری می‌تواند در زمان کوتاه‌تری از سمت سرور ارسال شود.

4. نسخه HTTP/2 مولتی‌پلکس است :در HTTP/1.1 ، فقط یک درخواست روی یک کانکشن اینترنتی TCP در آن واحد می‌تواند به کار گرفته شود و مرورگرها به طور عادی قادر به ایجاد ٤ تا ٨ کانکشن با سرور هستند و این در حالی است که ریکوئست‌هایی با حجم زیاد می‌توانند سرعت دانلود فایل‌های دیگر را به تاخیر بیندازند HTTP/2 !اجازهٔ ارسال چندین ریکوئست (درخواست) و دریافت ریسپانس (پاسخ) از سمت سرور را به طور هم‌زمان بر روی یک کانکشن امکان‌پذیر می‌سازد.

5. نسخه HTTP/2 امکان استفاده از Server Push را فراهم می‌سازد :با استفاده از این نسخه از پروتکل اچ‌تی‌تی‌پی، سرور می‌تواند فایل‌ها -و به طور کلی هر نوع داده‌ای- را قبل از آنکه ریکوئستی ارسال شود، برای مرورگر بفرستد که به این فناوری اصطلاحاً Server Pushگفته می‌شود. برای مثال، ممکن است شما در پایین صفحهٔ خود به یک اسکریپت لینک دهید. در HTTP/1.1، مرورگر کدهای HTML را دانلود می‌کند، تجزیه می‌کند و سپس فایل جاوااسکریپت را بارگذاری می‌کند (این بارگذاری هنگامی است که با تگ <script>روبه‌رو شویم). سروری که HTTP/2را ساپورت کند، می‌تواند چنین فایلی را قبل از اینکه نیاز آن را تشخیص دهد، برای مرورگر ارسال کند که در نتیجه در صورت نیاز، کاربر معطل دانلود شدن فایل‌های جی‌اس نخواهد شد که این به معنی UX بهتر است.

آیا HTTP/2 خیلی بهتر عمل می‌کند؟

این موضوع از یک سرور به سرور دیگری متفاوت است، اما HTTP/2 در مقایسه با HTTP/1.1 و منوط به اینکه از پروتکل امن HTTPS استفاده شده باشد، تا چند برابر **سریع‌تر** عمل می‌کند.

آیا هم‌اکنون زمان مهاجرت به HTTP/2 رسیده است؟

در یک کلام، **بله** .با این حال، HTTP/2تنها هنگامی فعال است که نرم‌افزار وب سروری همچون آپاچی و مرورگر کاربران این پروتکل را ساپورت کنند. هنگامی که هر یک از این دو مورد نتواند با موفقیت پروتکل را اجرا کند، کانکشن دوباره به حالت HTTP/1.1 برمی‌گردد. در پایان سال ۲۰۱٦، تقریباً ۱۱٪ از ده میلیون وب‌سایت HTTP/2 را ساپورت می‌کردند لازم به ذکر است که تمام

ورژن‌های فایرفاکس، سافاری، مایکروسافت اج، کروم و دیگر مرورگرهای مبتنی بر Blink این پروتکل را ساپورت می‌کنند.

چگونه تست کنیم که ببینیم وب‌سرورمان HTTP/2 را ساپورت می‌کند؟

ابزار HTTP/2 Test این امکان را در اختیار شما قرار می‌دهد تا تست کنید ببینید آیا سرور شما قادر به پشتیبانی از HTTP/2 می‌باشد یا خیر. اگر شما از سرورهای اختصاصی یا مجازی استفاده می‌کنید، می‌توانید HTTP/2 را به سادگی فعال سازید؛ اکثر وب‌سرورها -شامل Apache ، Nginx ، LiteSpeedو -Microsoft IIS به طور مستقیم از این پروتکل پشتیبانی می‌کنند یا دارای ماژول‌های از پیش نصب‌شده‌ای برای این کار هستند (البته توجه داشته باشیم که این وب‌سرورها تفاوت‌هایی هم با یکدیگر دارند).

آیا باید سورس‌کد وب‌سایت خود را آپدیت کرد؟

در یک کلام، **خیر** .هر کدی که روی HTTP/1.1 اجرا شود، روی HTTP/2 نیز به سادگی اجرا خواهد شد) البته شاید در آینده‌ای نه‌چندان دور، راه‌کارهایی که امروزه در کدنویسی استفاده می‌کنیم در فضای HTTP/2 کارایی چندانی نداشته باشند.(

توجه داشته باشیم که کاهش تعداد ریکوئست‌ها در زمانی که از پروتکل HTTP/1.1 استفاده می‌کنیم خوب است اما پس از مهاجرت به2HTTP/ ، می‌شود خیلی نگران تعداد ریکوئست‌ها نبود چرا که بار زیادی به سرورمان تحمیل نخواهند کرد حتی شاید بهتر باشد که مثلاً به جای یک فایل سی‌اس‌اس اصلی، از چندین فایل سی‌اس‌اس مجزا استفاده کرد تا در صورت نیاز به اعمال تغییر در یکی از آن‌ها، صرفاً نیاز به آپدیت یک فایل باشد.

در گذشته زمانی که وب‌مسترها می‌خواستند تا با استفاده از پروتکل HTTP/1.1 از قابلیت ارسال چندین ریکوئست هم‌زمان استفاده کنند، از چندین دامین یا CDN استفاده می‌کردند) مثلاً یکسری فایل‌های جاوااسکریپت را روی sub1.example.com قرار می‌دادند و یکسری دیگر را روی (sub2.example.com که این کار را به‌منظور برقراری کانکشن‌های بیشتری روی بستر HTTP/1.1 انجام می‌دادند اما خبر خوب اینکه انجام این کار برای HTTP/2 اصلاً لازم نیست چرا که در HTTP/2 شما می‌توانید در یک کانکشن، تعداد ریکوئست‌های زیادی ارسال کنید .

نتیجه‌گیری

به نظر می‌رسد باتوجه به اینکه امروزه بسیاری از موتورهای جستجو همچون گوگل اهمیت زیادی به زمان لود سایت می‌دهند و روزبه‌روز هم به تعداد کاربران دیوایس‌های هوشمندی همچون موبایل و تبلت افزوده می‌شود که بالتبع زمان لود سایت برای ایشان حائز اهمیت است، زمان مناسبی برای این مهاجرت باشد.

نظر شما چیست؟ آیا تاکنون امکان استفاده از پروتکل HTTP/2 را داشته‌اید و آیا توانسته‌اید از لحاظ پرفورمنسی مقایسه‌ای مابین این نسخه و HTTP/1.1 انجام دهید؟ نظرات و دیدگاه‌های خود را با سایر کاربران سکان آکادمی به اشتراک بگذارید.

۲) :

تفاوت‌ها

تفاوت‌ها در جزئیات و عمدتاً در این حیطه هستند به لطف استفاده‌ی HTTP/3 از :QUIC

- پروتکل HTTP/3 به لطف دست‌دهی ۰-RTT پروتکلِ QUIC از داده اولیه پشتیبانی بهتری می‌کند، هنگامیکه TCP Fast Open و TLS هماره داده کمتری ارسال می‌کنند و با مشکل مواجه می‌شوند.
- پروتکل HTTP/3 به لطف QUIC در مقایسه با TLS + TCP از دست‌دهی‌های به مراتب سریع‌تری برخوردار است.
- پروتکل HTTP/3 در نسخه‌ی نا-امن و بدون رمزگذاری وجود ندارد. پروتکل HTTP/2 می‌تواند بدون HTTPS پیاده‌سازی و استفاده شود - اگرچه در اینترنت کمتر بدین شکل دیده می‌شود.
- پروتکل HTTP/2 می‌تواند مستقیم داخل یک دست‌دهی TLS با افزونه ALPN قرار بگیرد، حال آنکه HTTP/3 بر روی QUIC است و ازینرو ابتدا به یک پاسخ سرایند :Alt-Svc نیاز دارد تا کارخواه را از این عامل آگاه سازد.
- پروتکل HTTP/3 اولویت‌بندی ندارد. رویکرد HTTP/2 در اولویت‌بندی پییچده تلقی می‌شود، و یا حتی صرفاً یک شکست، لذا کار بر روی ساخت موردی ساده‌تر در جریان است. لین طرح ساده‌تر هم برنامه‌ریزی شده تا پیش‌انتقال بتواند با استفاده از مکانیزم پسوند HTTP/2 بر روی HTTP/2 اجرا شود.

۳) :

کدهای سری ۳xxx (تغییر مسیر)

300 **انتخاب چندگانه :** سرور لیستی از لینک‌ها را نمایش داده و کاربر می‌تواند آنها را انتخاب کرده و به آن مکان برود.

301 **انتقال دائم :** صفحه درخواست داده شده به صورت دائمی به صفحه دیگری انتقال و یا ارجاع داده خواهد شد. در این حالت سرور به صورت خودکار کاربر را به صفحه جدید هدایت می‌کند.

302 **انتقال موقت یا پیدایش :** درخواست صفحه مورد نظر شما به صورت موقت به صفحه جدیدی ریدایرکت خواهد شد.

303 **مراجعه به قسمت دیگری :** این خطا در هنگامی رخ می‌دهد که کاربر باید برای دریافت درخواست خود به محل دیگری مراجعه کند.

304 **اصلاح نشده :** هنگامی که سرور این وضعیت را نمایش می‌دهد بدین معناست که از آخرین باری که درخواست ارسال شده ، صحفه مورد نظر اصلاح نشده است و در این حالت سرور به اجبار یک صفحه خالی به شما نمایش خواهد داد.

305 **استفاده از پروکسی :** کاربر زمانی می‌تواند از صفحه مورد درخواست استفاده کند که از پروکسی استفاده کرده باشد.

307 **انتقال موقت :** همانند کد 302 عمل کرده اما نوع ریدایرکت موقت آن کمی متفاوت خواهد بود.

۴) :

Http Request Headers

Accept-Encoding : لیست کدگذاری‌ها و فشرده‌سازی‌های قابل قبول در پاسخ.

Host : مشخصی کننده نام هاستی که به آن درخواست ارسال می شود .

Referrer : از کجا به اینجا آمده اید. این معادل همان دکمه Back در مرورگر می باشد.

Http Response Headers

Cache-Control : آیا این محتوا قابل cache می باشد یا خیر، مثلا این محتوا private است یا public که بتوان آن را cache کرد. اگر این محتوا public باشد قابل cache است یا خیر، و اگر است به مدت زمان یا life time .

Content-Type : نوع Entity Body که در Response ارسال می گردد.

Content-Length : حجم بدنه منبع (بدون در نظر گرفتن سرآیندها) به واحد بایت (بایت‌های ۸ بیتی).

Location : آدرس url که کاربر باید به آن ارسال شود. به طور کل از این هدر برای ارسال کاربر به مکان مختلف پس از انجام پردازش استفاده می کنیم.

Content-Range : نشون میده که توی یک پیام، تمام بدنه یک پیام، جزئی از یک کل هست.

Last-Modified : وجود تاریخ اصلاح خیلی مهمه. از آخرین تاریخ اصلاح منبع برای مقایسه چند نسخه از همون منبع استفاده میشه.