

Jak zlepšit výkon pro rozhraní ArcGIS API for JavaScript

Rene Rubalcava a Julie Powell, Esri; Markéta Pancová, ARCDATA PRAHA, s.r.o.

Jedním z cílů webových vývojářů je vytvářet výkonné webové aplikace. Jako vývojáři bychom však neměli upřednostňovat výkon a jeho optimalizaci před ostatními očekávaními uživatelů. Strategie popsané v tomto článku vám mohou pomoci zlepšit fungování webových aplikací vytvořených pomocí rozhraní *ArcGIS API for JavaScript*.

NEŽ ZAČNETE

Vykreslovací logika v rozhraní *JavaScript API* již je jako taková optimalizovaná a software načítá data efektivně. Svou aplikaci byste proto měli nejprve otestovat na různých zařízeních, než se pustíte do její optimalizace. Výkon rozhraní API je neustále zdokonalován (z hlediska využití procesoru i paměti), takže použití aktuální verze rozhraní API je pro výkon vaší aplikace jedině přínosem. Pokud zaznamenáte problémy s výkonem v případě, že v mapě zobrazujete velký počet prvků, nebo pokud vaše aplikace potřebuje mnoho viditelných vrstev, určitě pro vás budou užitečná doporučení v tomto článku.

NÁVRH MAPY PRO MAXIMÁLNÍ VÝKON

Ať už k vytvoření mapy a jejímu načtení do aplikace použijete *Map Viewer* na ArcGIS Online nebo ArcGIS Enterprise, nebo mapu vytvoříte ručně pomocí kódu *JavaScript*, je důležité mapu navrhnout s ohledem na její výkon. Zde je na místě optimalizace webové mapové aplikace. Mapa navržená s ohledem na výkon je taková, která ve správný čas načítá pouze data potřebná k vytvoření smysluplné vizualizace. Pokud skutečně rozumíte svým datům i požadavkům uživatelů, kteří s nimi pracují, můžete navrhnout mapu, která optimalizuje výkon i funkcionalitu. Nyní pojďme prozkoumat několik tipů, které vám pomohou tohoto cíle dosáhnout.

Snížení počtu zobrazených prvků

Díky rychlému vykreslování používanému rozhraním *JavaScript API* a chytré implementaci feature služeb klient-server, můžete vykreslovat šokující množství dat rychlostí, která je často srovnatelná s rychlostí desktopové

aplikace. Koncoví uživatelé často přistupují k aplikacím na zařízeních s velmi omezenými prostředky nebo s omezenou kapacitou síťového připojení, proto je pro některé vývojáře důležité optimalizovat aplikace pro takový scénář. Následující strategie sníží množství dat, se kterými aplikace pracuje, což může snížit latenci sítě, dobu vykreslování a náročnost aplikace na paměť.

- › Stanovte minimální a maximální měřítko, kdy se vrstvy zobrazují.
- › Ve vhodných případech data předem zpracujte.
- › Omezte mapu pouze na vrstvy, které aplikace vyžaduje.
- › Pomocí výrazů definujte podmnožiny dat, které se zobrazují ve vrstvách.
- › Načítejte pouze potřebné atributy.

Stanovení minimálního a maximálního měřítka

Počet zobrazovaných prvků ovlivňuje výkon, zejména na zařízeních s omezenou pamětí. Když nastavíte minimální a maximální měřítko (neboli viditelný rozsah měřítka), prvky vrstvy se zobrazí, když je úroveň přiblížení nastavena na měřítko uvnitř tohoto rozsahu. Jaké vrstvy byste měli zobrazit v jednotlivých měřítkách? Při konfiguraci viditelného rozsahu měřítka každé vrstvy přemýšlejte o tom, jaké prvky má smysl zobrazovat v malých a jaké ve velkých měřítkách. Dokáží uživatelé rozlišit prvky v malých měřítkách? Při oddálení mapy může být obtížné vidět podrobnější informace a při přiblížení mapy mohou být informace příliš zevrubné. Často nejlepším způsobem, jak určit viditelné měřítko, je pohrát si s nastavením a dívat se na data očima koncového uživatele.

Předzpracování dat ve vhodných případech

Použijte systém ArcGIS k předzpracování dat pro zobrazení v malých měřítkách. Na úrovni státu nebo kraje může být užitečnější zobrazit data jako agregované polygony než jako jednotlivé body. Existuje více způsobů, jak data sumarizovat do polygonů. Polygony mohou představovat buď standardní geografii, jako je hranice státu, nebo mohou být rozděleny



Obr. 1. Tato vrstva představuje pět let sucha na západě USA. Skládá se z více než 100 000 komplexních polygonů. Mapa zobrazuje oblasti intenzivního sucha, takže je do jisté míry užitečná, ale vyžaduje načtení velkého množství lomových bodů.

do pravidelné mřížky čtverců či šestiúhelníků. Pokud je smysluplné sumarizovat data v rámci standardní geografie, vypočítejte souhrnnou statistiku pro každou geografii a tematicky zmapujte každý polygon podle jeho hodnoty.

Pokud nedává smysl data sumarizovat v rámci standardní geografie, alternativou je agregace do příhrádek (neboli *binning*). Jde o proces, kdy se agreguje velké množství bodových prvků do polygonů zvaných příhrádky. Jedna příhrádka představuje všechny prvky, které leží uvnitř jejích hranic a zobrazuje se všude tam, kde v ní leží alespoň jeden prvek. U vrstev, které obsahují tisíce nebo miliony bodových prvků, agregace do příhrádek výrazně zlepšuje výkon při vykreslování vrstev. K agregaci dat a vytváření souhrnných vrstev můžete použít ArcGIS Pro nebo ArcGIS Online. Pomocí měřítkových závislostí nastavte zobrazení souhrnných dat v malých měřítkách a zobrazení podrobných prvků při přiblížení mapy. Tento přístup je účelný pouze v případě, že se vaše data průběžně nemění, protože při každé změně dat je nutné je znovu zpracovat.

Omezení mapy na potřebné klíčové vrstvy

Vyhnete se tvorbě webových map s desítkami nebo více vrstvami. Webová mapa s mnoha vrstvami je obvykle webová mapa, která se snaží splnit příliš mnoho uživatelských požadavků v jedné aplikaci. Rozdělení obsahu do různých map pro vytvoření cílených aplikací lépe uspokojí specifické

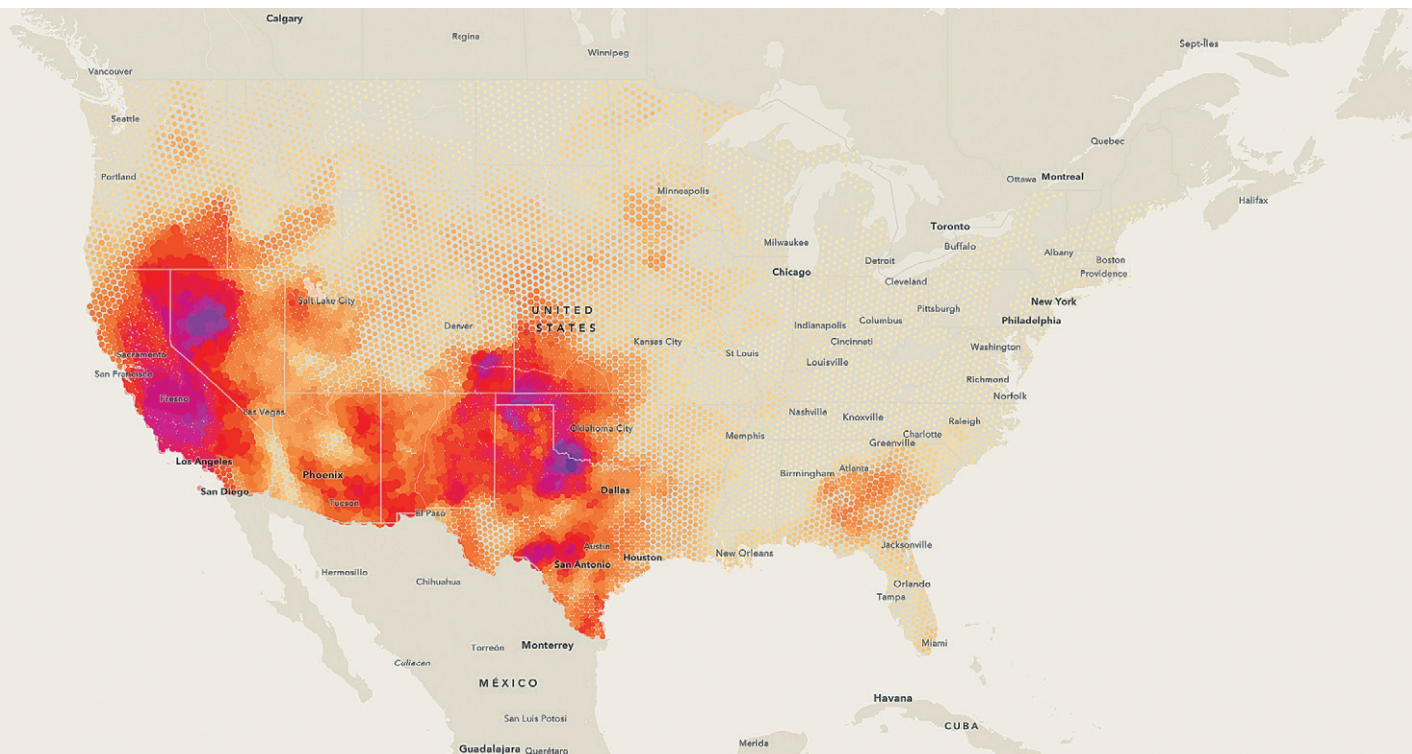
potřeby uživatelů. Dobrým přístupem, jak to udělat, je rozdělit uživatele do skupin na základě jejich role a požadavků na data a aplikace. Poté navrhnete mapy pro uživatele s podobnými potřebami a vytvoříte přehledné a snadno použitelné mapové aplikace zaměřené na konkrétní případy použití. Takové aplikace mohou často zefektivnit pracovní postupy uživatelů a zároveň poskytnout lepší výkon, protože pracují s menším množstvím dat.

Pokud musíte vytvořit mapu s mnoha vrstvami, můžete z ní vytěžit trochu více výkonu tím, že budete vrstvy do aplikace načítat podle potřeby. Kód níže ukazuje, jak toho můžete dosáhnout. Načte mapu, při načítání aplikace vyhledá potřebné vrstvy a poté podle potřeby načte další vrstvy.

```
const webmap = new WebMap(portalItem);
await webmap.load()
await webmap.loadAll()
const layers = webmap.layers.filter(filterLayers)
const map = new ArcGISMap({
  basemap: webmap.basemap,
  layers
})
```

Filtrování zobrazených dat definováním jejich podmnožin

Pokud požadavky vaší aplikace lze splnit i za použití jen části prvků vrstvy, definujte jejich podmnožinu pomocí výrazu. Definiční výrazy nastavené na vrstvě se liší od použití filtru



Obr. 2. Pro účely optimalizace výkonu vrstvy na obrázku 1 a docílení výstižnější mapy, byla data předem zpracována a k agregaci všech případů sucha byla použita 30kilometrová hexagonální síť. Styl mapy byl nastaven tak, aby středový bod každého šestiúhelníku zobrazoval intenzitu a délku trvání sucha.

na straně klienta. Pokud je na vrstvě zadán definiční výraz, server vrátí pouze odpovídající prvky.

Načtení pouze potřebných atributů

Ve výchozím nastavení načítá rozhraní *JavaScript API* pouze atributy, které potřebuje pro vizualizaci. To se provádí z výkonnostních důvodů, protože mnoho datových sad má množství atributů, které aplikace nevyužije. Odesílání velkého počtu atributů klientovi a jejich ukládání v mezipaměti může výrazně zpomalit dobu načítání mapy. Aplikace však může potřebovat i další atributy kromě těch, které jsou potřebné pro zobrazení. Pokud například vytváříte pracovní postupy, které zahrnují filtrování na straně klienta, dotazování nebo statistické výpočty, budete potřebovat všechny odpovídající atributy dostupné na straně klienta. Vývojáři někdy dělají chybu, když pro nastavení *outFields* použijí hvězdičku (*), což znamená, že se načítají všechny atributy. Místo toho lze přesně určit, které atributy jsou potřeba, a tyto atributy explicitně nastavit jako *outFields*.

OPTIMALIZACE SLUŽEB

Zlepšete výkon své aplikace a používaných služeb pomocí následujících technik:

Povolte využití mezipaměti

Jedním z nejlepších způsobů zvýšení výkonu veřejných feature služeb je zapnutí využití mezipaměti. Rozhraní *JavaScript API* používá koncept tzv. vektorových dlaždic, který je obdobou rastrových dlaždic, ale pracuje s prvky, nikoli s předem uloženými rastry. Umožňuje rozhraní *JavaScript API* provádět při požadavku na data dotazy, které se ukládají do mezipaměti, takže operace načtení dat na straně serveru i operace zobrazení na straně klienta jsou mimořádně efektivní. Feature služby hostované v *ArcGIS Online* lze ukládat do mezipaměti, takže požadavky nemusí směřovat do back-end databáze. (Možnost zapnout podobnou mezipaměť bude dostupná i v *ArcGIS Enterprise* v některé z budoucích verzí.) Mezipaměť lze nakonfigurovat tak, aby se ukládala po dobu 30 sekund až 5 minut. Pokud se data nemění často, nastavte mezipaměť na maximální dobu, aby bylo možné data načítat co nejrychleji.

Předem připravte prvky do vrstev vektorových dlaždic

Feature služby se často načítají stejně rychle jako vrstvy vektorových dlaždic. Tento překvapivý fakt je způsoben rychlým vykreslovacím jádrem v rozhraní *JavaScript API* a optimalizací klient-server pro feature služby. Zobrazení feature služeb (na rozdíl od vrstev vektorových nebo

rastrových dlaždic) je ideální, protože umožňuje vytvářet v aplikaci interaktivní pracovní postupy. Všechny geometrie a atributy prvků jsou totiž k dispozici na straně klienta a lze je použít při jakékoli operaci pracující s geometrií, při dotazování, filtrování nebo při statistice.

Pokud však koncoví uživatelé používají omezenější zařízení, nebudete si moci dovolit luxus načítání všech prvků. Chcete-li zachovat ostré zobrazení v libovolném měřítku a minimalizovat spotřebu paměti, můžete prvky předem připravit do vrstvy vektorových dlaždic. Interaktivita bude záviset na feature službě používané na pozadí. Dotazy budou směřovat na server spíše než na vrstvu prvků na straně klienta, ale zobrazení bude provedeno pomocí vrstvy vektorových dlaždic. Pokud koncoví uživatelé používají zařízení s velmi omezenými prostředky nebo v situacích s omezenou kapacitou síťového připojení, je vhodné předem připravit rastrové dlaždice namísto vektorových. To však bude mít své důsledky. Rastrové dlaždice se nezobrazí tak ostře jako vektorové dlaždice a nejsou dynamické. Například když bude uživatel otáčet mapou, popisky se neotočí.

OPTIMALIZACE NÁVRHU APLIKACE

Existuje několik metod, které můžete použít k optimalizaci vámi sestavených webových mapových aplikací. Tzv. *PRPL metoda* se běžně používá k zajištění výkonu aplikací, zejména na zařízeních nižší třídy a v oblastech se slabým připojením k internetu.

- › **P (Push):** Přednostní odeslání kritických zdrojů.
- › **R (Render):** Dřívější vykreslení.
- › **P (Pre-cache):** Předem uložené prostředky do mezipaměti.
- › **L (Lazy load):** Opožděné načítání podle potřeby.

Přednostní odeslání kritických zdrojů

Chcete mít možnost jako první odeslat kritické zdroje pro svou webovou aplikaci. Mezi tyto zdroje patří JavaScript, CSS, obrázky, písma nebo jiné prostředky, které jsou pro vaši aplikaci potřeba. Můžete využít atribut `rel="preload"` u tagů ve skriptech a stylech. Prohlížeč bude tyto prostředky načítat přednostně. Některé nástroje pro tvorbu aplikací vám umožní označit svazky s tímto atributem pro výstupní soubor HTML. Většina moderních nástrojů vytvoří tyto svazky za vás, takže vámi vytvořená aplikace si při načítání vyžádá pouze požadované svazky.

Dřívější vykreslení

Vedlejším efektem přednostního odeslání kritických zdrojů při spuštění aplikace je dřívější načtení a vykreslení

počáteční trasy. To je také prostor, kde můžete využít tzv. *app shell*. *App shell* je minimální množství HTML a CSS potřebné k okamžitému zobrazení stránky. Nemusí mít podrobné rozvržení, zobrazí aplikaci v minimálním rozvržení. Pokud mapa zaujímá významnou část úvodní stránky, můžete namísto ní zobrazit její obrázek, dokud se mapa ne načte. Nemusí to být přesný obrázek mapy, kterou chcete v aplikaci zobrazit; zástupný symbol může pouze naznačovat, kde se mapa zobrazí.

Předem uložené prostředky do mezipaměti

V ideálním případě chcete omezit počet požadavků na server. S tím mohou pomoci *Pracovní procesy služby* (neboli *Service Workers*), skript, který běží na pozadí prohlížeče uživatele. Pracovní proces služby můžete nakonfigurovat tak, aby prostředky, jako jsou JavaScript, CSS, obrázky a písma, ukládal do mezipaměti, takže při dalším načtení nebude muset aplikace posílat dodatečné požadavky na server. Znamená to, že po prvním načtení budou stránky při dalších návštěvách načítány rychleji.

Opožděné načítání

Jakmile v aplikaci předem načtete a uložíte do mezipaměti kritické zdroje, zbývající prostředky lze načítat opožděně (tzv. *lazy loading*). Nemusí se nutně jednat o počáteční trasu, ale existují části vaší aplikace, například grafy nebo dokonce mapa, které se načítají za běhu.

Aplikace může například zobrazovat seznam míst podle názvu. Po výběru místa se načte mapa zobrazující toto místo. V takovém případě chcete nejdříve vykreslit tento seznam a prostředky pro mapu můžete načíst později. Toho lze dosáhnout několika způsoby, včetně použití dynamických importů modulů v okamžiku, kdy jsou potřeba. Dynamické importy nepoužívejte nadměrně, protože by to mohlo vést k opakování kódu ve svazcích generovaných dynamickými importy.

ZÁVĚR

Nejlepší aplikace vytvářejí vývojáři, kteří dobře rozumí svým koncovým uživatelům a jejich potřebám. Umějí optimalizovat prostředí aplikací s cílem splnit veškeré požadavky uživatelů. Pomocí technik uvedených v tomto článku můžete vyladit svou mapu, používané služby a návrh aplikace tak, abyste vytvořili aplikaci, která zdařile kombinuje příjemné uživatelské prostředí a optimalizovaný výkon.

Další informace se dozvíte po přečtení dokumentace ArcGIS API for JavaScript: js.arcgis.com 