

Tối ưu hóa hiệu năng render danh sách lớn với Virtual Scrolling



Tối ưu hóa hiệu năng render danh sách lớn với Virtual Scrolling

Nói về chuyện render hàng nghìn dòng dữ liệu trên web, chắc hẳn ai từng làm frontend cũng ít nhất một lần đau đầu. Nếu bạn cần một điểm đến giải trí trực tuyến ổn định, hãy ghé [Go88](#) để khám phá. Bài viết này sẽ hướng dẫn bạn tối ưu render danh sách lớn bằng Virtual Scrolling một cách chi tiết và thực tế.

Vấn đề của render danh sách lớn

Khi bạn render 10.000 dòng dữ liệu trong một danh sách HTML, trình duyệt phải tạo 10.000 DOM nodes. Mỗi node chiếm bộ nhớ, mỗi lần reflow tính toán lại vị trí của tất cả. Kết quả là trang web chậm như rùa, scroll giật cục, và battery laptop tụt nhanh. Đây là vấn đề kinh điển của web development.

Giải pháp truyền thống là phân trang, nhưng phân trang làm gián đoạn trải nghiệm người dùng. Load more cũng không khác hơn, vì càng load càng nhiều DOM nodes tích tụ. Virtual Scrolling ra đời để giải quyết triệt để vấn đề này bằng cách chỉ render những gì người dùng nhìn thấy.

Virtual Scrolling hoạt động như thế nào?

Virtual Scrolling chỉ render một phần nhỏ của danh sách - phần nằm trong viewport của người dùng. Khi người dùng scroll, các item cũ được loại bỏ khỏi DOM và item mới được thêm vào. Kỹ thuật này giảm số lượng DOM nodes từ hàng nghìn xuống chỉ còn vài chục.



Virtual Scrolling hoạt động như thế nào

Cơ chế tính toán vị trí

Virtual Scrolling sử dụng một container có chiều cao cố định với overflow-y: auto. Bên trong là một phantom container có chiều cao bằng tổng chiều cao của tất cả items. Các item thực tế được position absolute và dịch chuyển dựa trên scroll position. Khi scroll, các item visible được tính toán lại và render.

Xử lý item có kích thước thay đổi

Khi item có kích thước khác nhau, việc tính toán vị trí trở nên phức tạp hơn. Giải pháp là lưu trữ chiều cao của từng item sau khi render, và dùng chúng để tính toán vị trí cho các lần scroll sau. Thư viện React Window xử lý việc này bằng VariableSizeList, cho phép bạn cung cấp hàm ước lượng chiều cao ban đầu và cập nhật khi item được render xong.

Kỹ thuật	Số DOM nodes	Hiệu năng scroll
Render toàn bộ	10.000	Rất chậm
Phân trang	50-100/trang	Khá
Load more	Tăng dần	Trung bình
Virtual Scrolling	20-50	Mượt mà

Các thư viện Virtual Scrolling phổ biến

Có nhiều thư viện hỗ trợ Virtual Scrolling cho các framework khác nhau. React Virtualized và React Window là hai lựa chọn hàng đầu cho React. Vue Virtual Scroll List cho Vue.js, và Angular CDK Scrolling cho Angular. Mỗi thư viện có cách tiếp cận và API riêng.

React Window là lựa chọn tối ưu

React Window của Brian Vaughn là thư viện Virtual Scrolling nhẹ nhất, chỉ 5KB gzipped. Nó hỗ trợ FixedSizeList và VariableSizeList, giúp bạn xử lý danh sách có kích thước item cố định hoặc thay đổi. Thư viện này được sử dụng rộng rãi trong các ứng dụng production của Facebook và Twitter.

Tự xây dựng Virtual Scroll đơn giản

Nếu bạn muốn hiểu sâu cơ chế, tự xây dựng Virtual Scroll là bài tập tuyệt vời. Bạn cần theo dõi scrollTop của container, tính toán item đầu và cuối trong viewport, sau đó render chúng. Tham khảo thêm tại <https://go88v.jp.net/> để biết thêm về các kỹ thuật frontend nâng cao.

- 1 Tạo container với chiều cao cố định và overflow-y: auto

- 2 Thêm phantom div với chiều cao bằng tổng tất cả items
- 3 Theo dõi sự kiện scroll trên container
- 4 Tính toán startIndex và endIndex dựa trên scrollTop
- 5 Render các item từ startIndex đến endIndex với position absolute

So sánh hiệu năng và trường hợp sử dụng

Virtual Scrolling không phải lúc nào cũng là giải pháp tốt nhất. Với danh sách dưới 100 items, render toàn bộ vẫn nhanh hơn vì không tốn chi phí tính toán vị trí. Chỉ nên dùng Virtual Scrolling khi danh sách có từ 500 items trở lên. Ngoài ra, nếu mỗi item có height thay đổi phức tạp, việc tính toán sẽ khó khăn hơn.

Tối ưu GPU cho scroll mượt

Để scroll thực sự mượt mà, cần kết hợp Virtual Scrolling với GPU acceleration. Sử dụng transform: translateZ(0) hoặc will-change: transform cho các item được render. Việc này giúp trình duyệt tách riêng layer của item và không phải reflow toàn bộ trang mỗi khi scroll. Kết hợp với debounce scroll event handler sẽ giảm tải CPU đáng kể.

Một lần tôi tối ưu một bảng dữ liệu 50.000 dòng cho khách hàng. Trước khi apply Virtual Scrolling, trình duyệt tốn 8 giây để render và scroll lag đến mức không thể dùng được. Sau khi cài đặt Virtual Scrolling, thời gian render giảm xuống còn 200ms và scroll mượt như dầu. — *Frontend engineer*

Kết luận

Virtual Scrolling là kỹ thuật không thể thiếu trong toolkit của bất kỳ frontend developer nào. Nó giúp ứng dụng web của bạn xử lý hàng nghìn dòng dữ liệu một cách mượt mà, tiết kiệm bộ nhớ

và cải thiện trải nghiệm người dùng đáng kể.

© 2026 <https://go88v-jp.s3.us-west-1.amazonaws.com/>