astrophysics data system

User Interface Report

Tim Hostetler, Jennifer Chen
and the ADS Team

ADS Users Group Meeting, 05-06 Dec. 2024

CENTER FOR ASTROPHYSICS HARVARD & SMITHSONIAN

NASA Partner

Outline

SciX UI Improvements Since Last Meeting

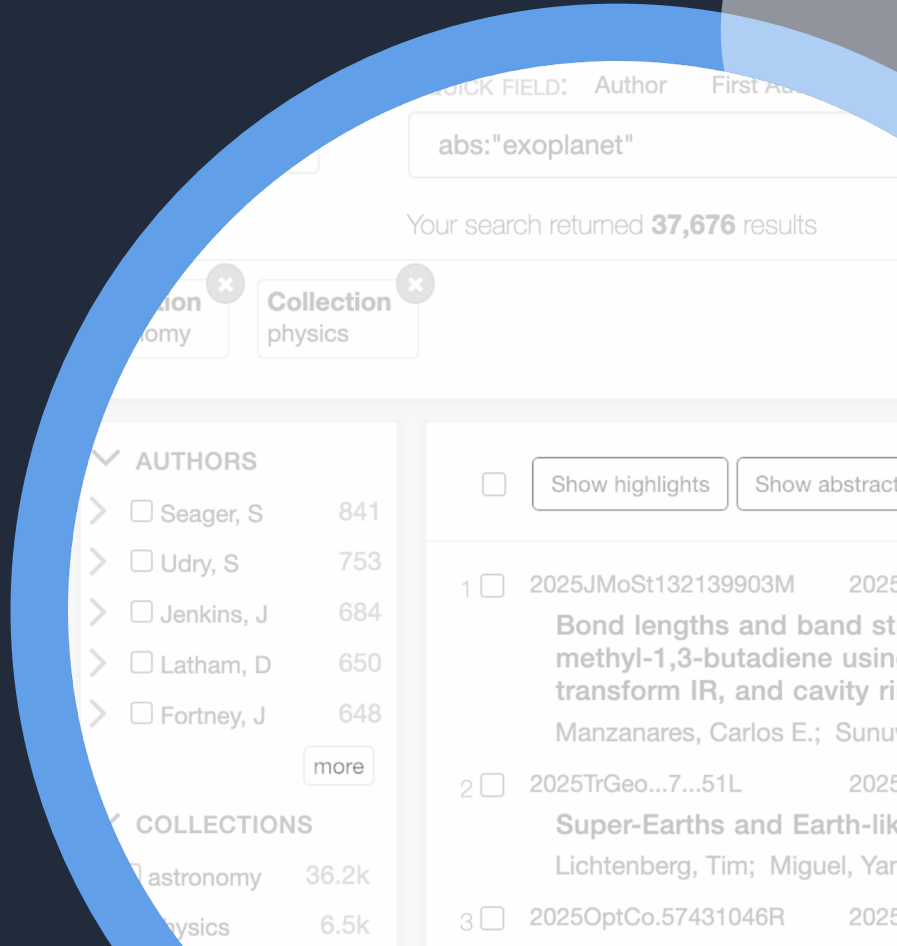
Performance Monitoring

Core Metrics

Comparing Competitors

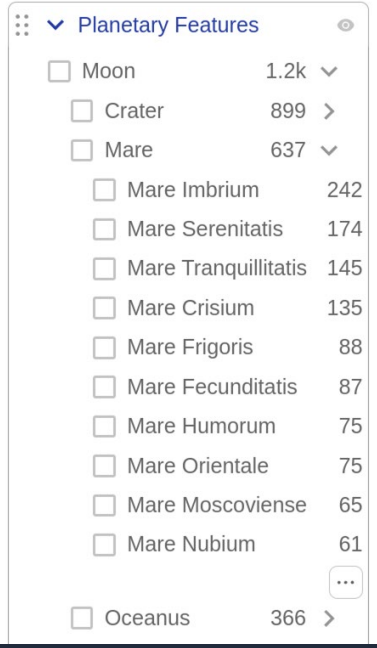
Integration Testing

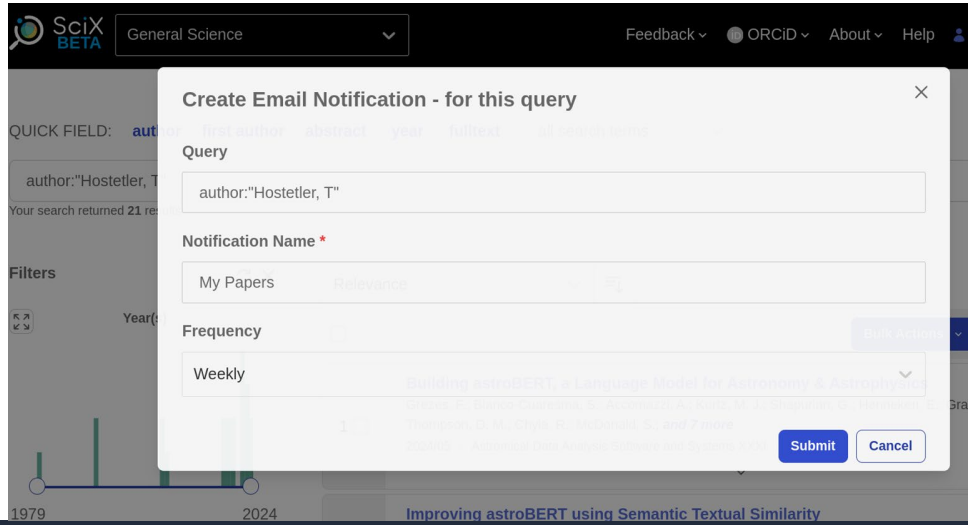
Wrapping Up and Next Steps



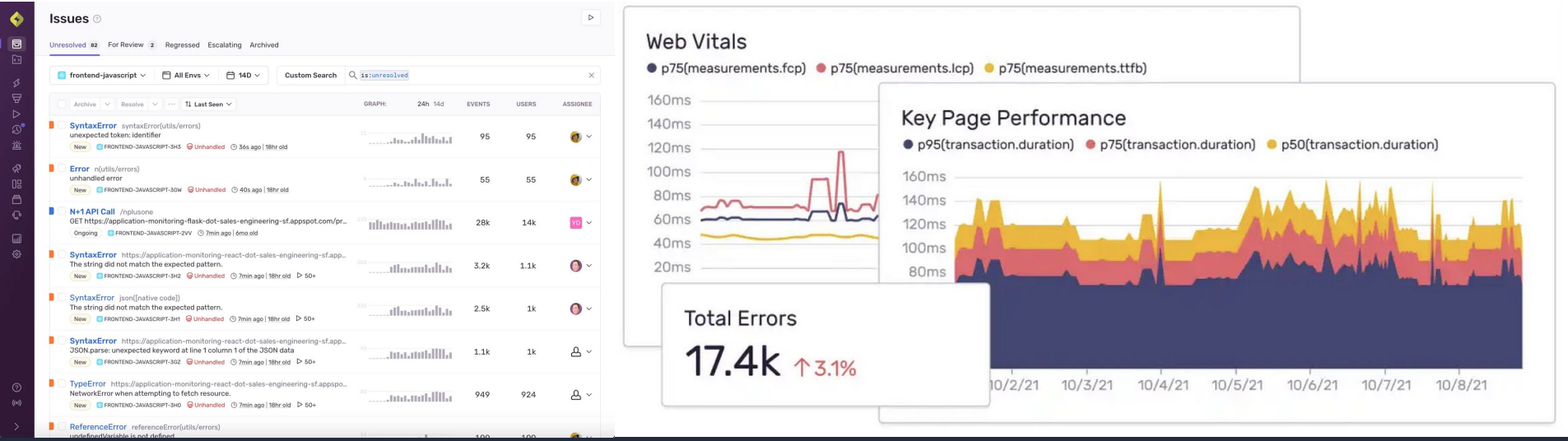
SciX UI Improvements Since Last Meeting

- **Email Notifications (myADS) Fully Implemented in SciX**
  - Users can make email notifications for their queries from results or directly in the Email Notifications dashboard
- **New Planetary Features Facet**
  - First 3-level facet
- **Accessibility Improvements**
- **Improvements to landing page**
  - Carousel, introductory video
- **Improved Error Handling**





- **Sentry.io**
  - Provides custom metrics/timings for our apps (SciX and ADS)
  - Records anonymized user replays with events (errors, web vitals, timings, network requests, etc.)
- **How we use it**
  - Track and prioritize unhandled errors, using replays we can find out if it affect a user negatively and analyze trends in web vitals and timings
  - Continuing to understand how to utilize efficiently
  - We plan on creating dashboards to see user experience trends across both apps at a glance

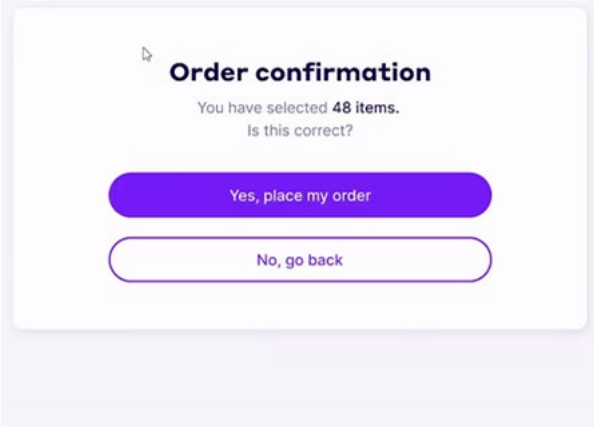


Core Metrics

- **Goals**
  - Determine core metrics to be used when measuring and comparing apps
  - Benchmark to determine the baseline performance
  - Compare with competitors
    - Semantic Scholar
    - Google Scholar
    - Geoscience World
    - PubMed
    - INSPIRE-HEP
  - Create a convenient way to analyze trends and identify regressions

When comparing web applications, the industry standard performance metrics are the **Core Web Vitals**

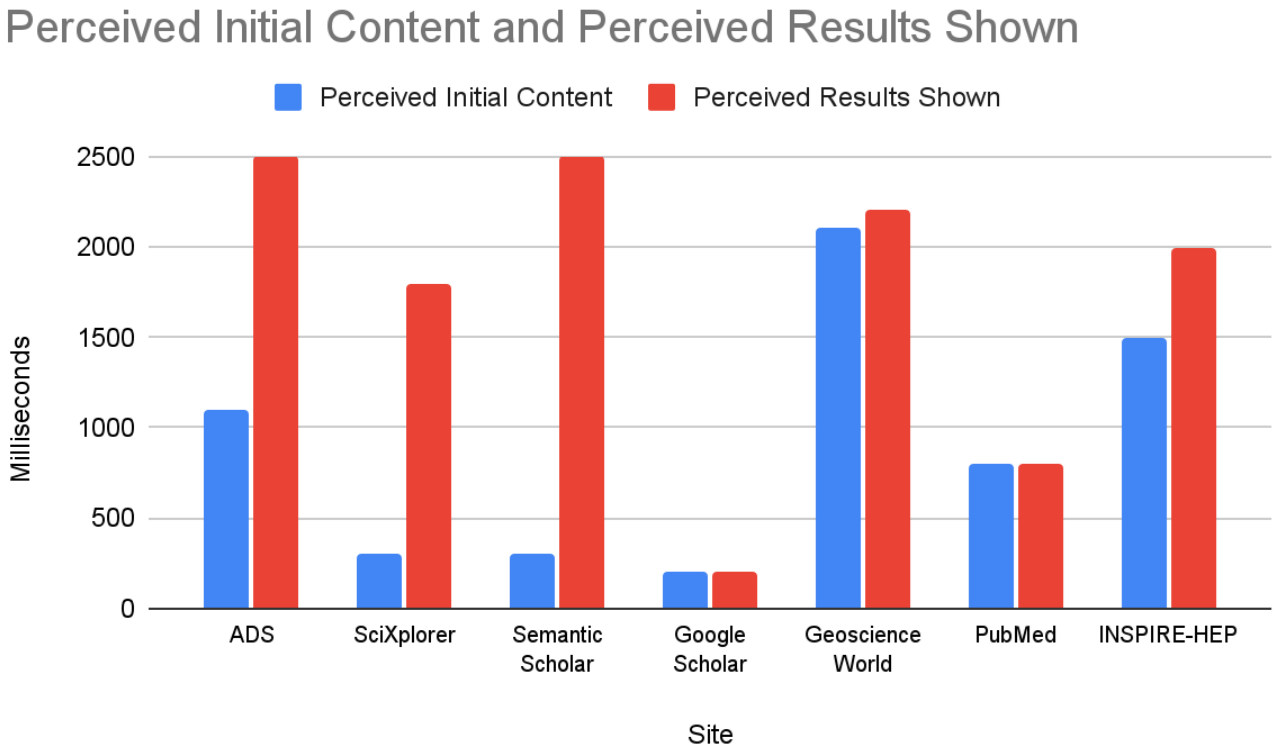
- **Largest Contentful Paint (LCP)**: Measures loading performance. How long does it take until the largest (in sq. pixels) paint is completed.
- **Interaction To Next Paint (INP)**: Measures responsiveness. When clicking on an interactive element (i.e. button) how long does the app take to have some visual response, disregarding page transitions.
- **Cumulative Layout Shift (CLS)**: Measures visual stability. How much extra “jumping” around the page does after content is loaded, sometimes causing users to accidentally click on something they didn’t mean to



Core Metrics

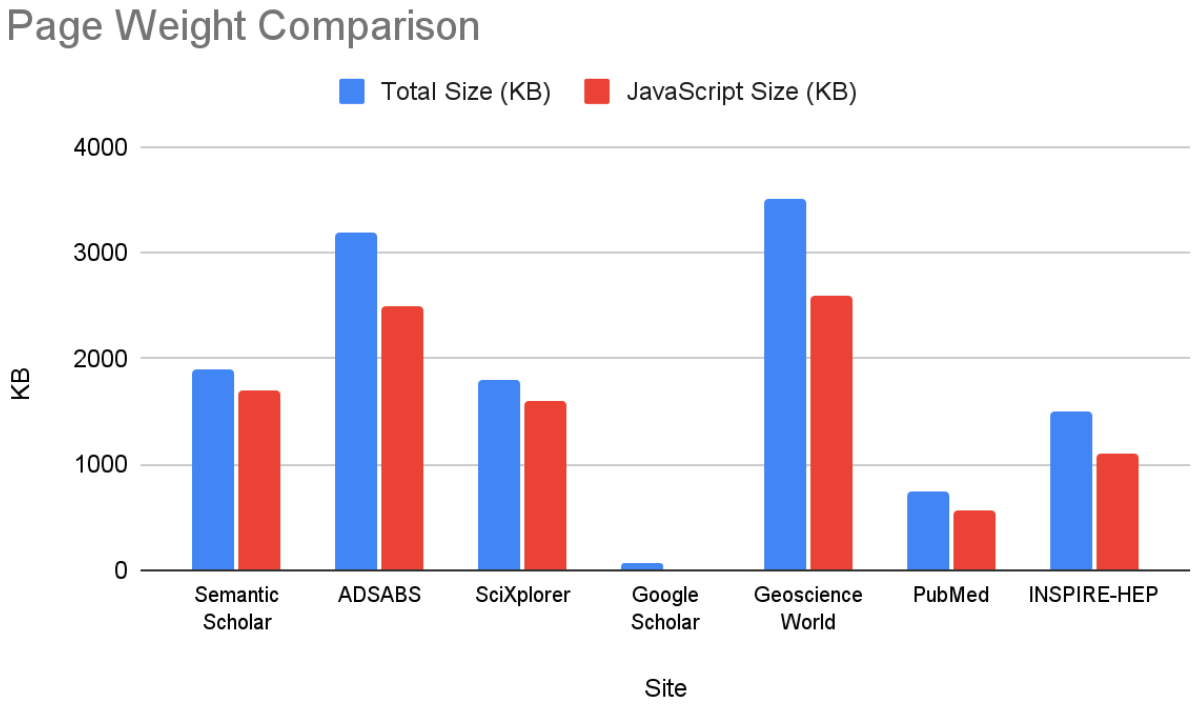
- **Beyond Core Metrics**
  - First Contentful Paint – How long until app does very first paint (might be a loading indicator, for example)
  - Total Blocking Time – Time spent on doing CPU heavy tasks (loading JavaScript, for example), when the user would be blocked from interacting with the app.
  - First Input Delay – How long it takes for the app to react to a user's interaction
- **How to use these**
  - These measurements can help let us know if generally our users are having a positive experience
  - We have to confirm all of them to make sure that they are measuring what we care about

Perceived Results Shown is the same as Largest Contentful Paint (web vital) timing in each app
* With the exception of ADS, which was verified using screenshots



Comparing Competitors

Another potential point of comparison we can make is **Page Weight**.
This takes into account the full size of the downloaded assets, not counting other payloads (like results, user settings, etc.)



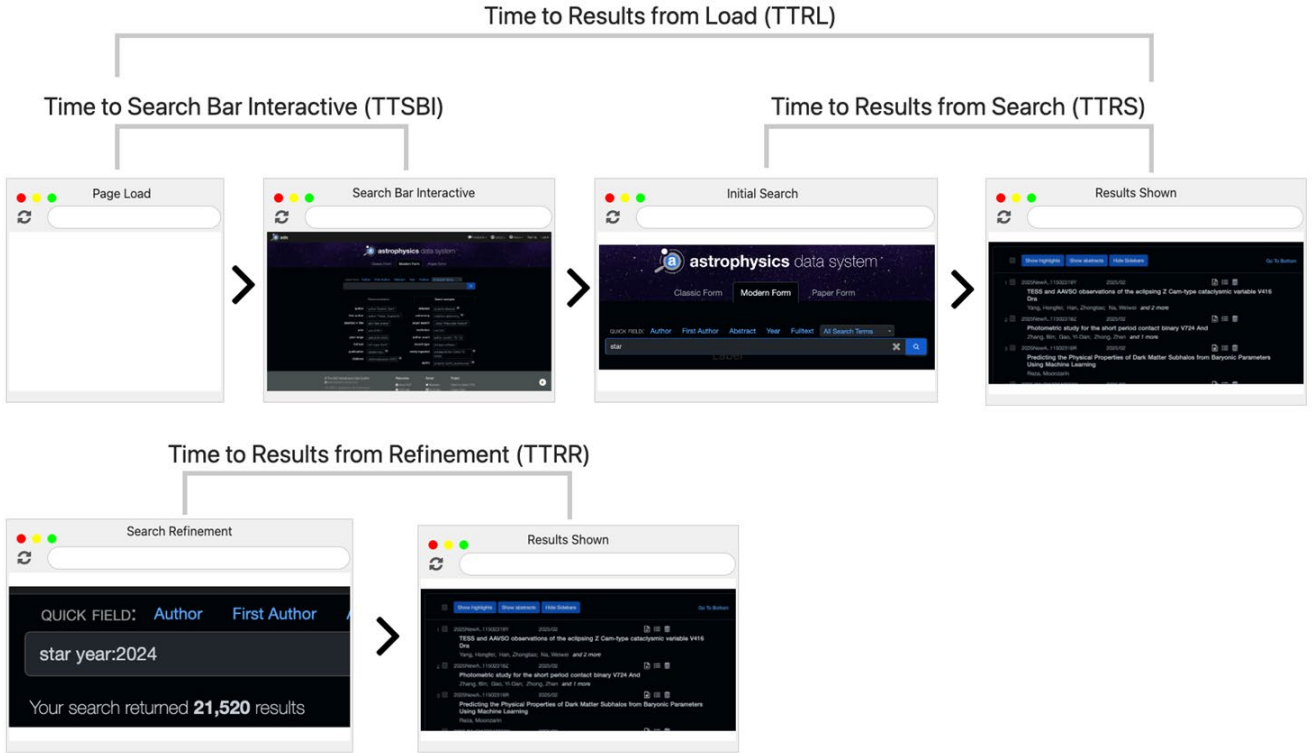
Integration Testing

Goals

- Create a set of core timings we care about and that are representative of app performance
- Create a performance test suite that can run on a variety of queries

Custom Timings:

- `TTSBI` – Time taken for the search bar to become ready for input.
- `TTRL` – Time taken from page load to initial results display.

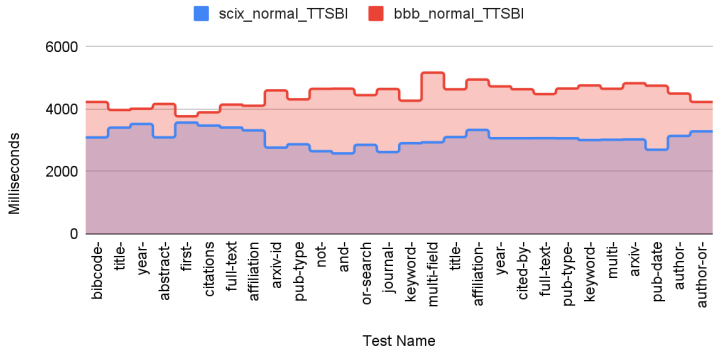


For more information, see the full Integration testing report

Integration Testing Results

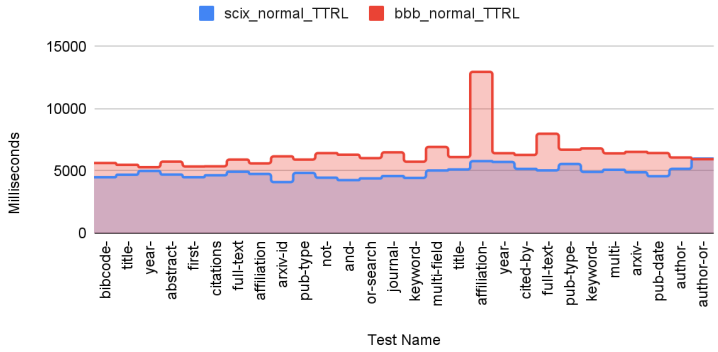
Time To Search Bar Interactive

Starting from pageload, how long until we see search bar and can enter search terms



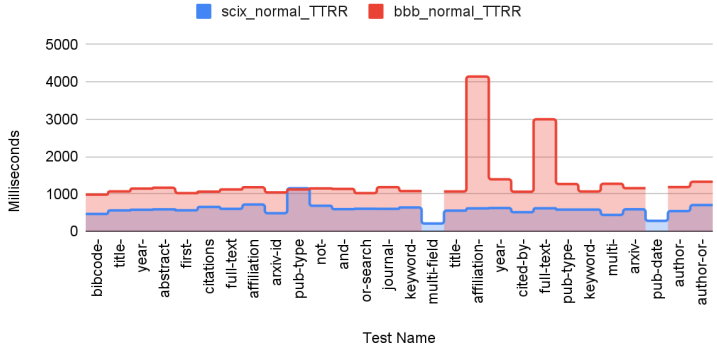
Time To Results from Load

Starting from pageload, how long until we finally see results for the first time



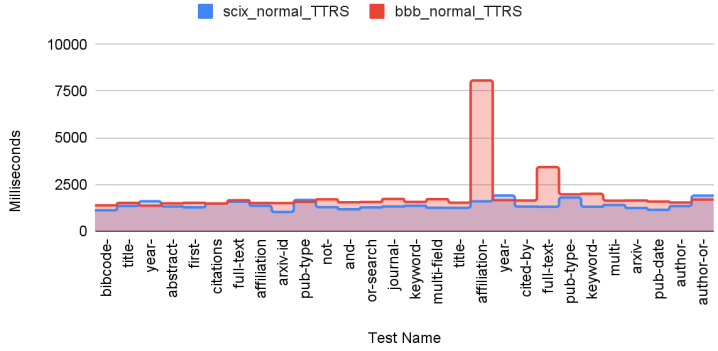
Time To Results from Refinement

Starting from refined search submitted, how long until we see results on screen



Time To Results from Search

From search submitted, how long until we see results on screen



Results & Conclusions

- **Results**
  - **Landing Page Performance**
    - SciX outperforms ADS from a full app load by **500 ms on average** due to its lighter initial payload.
    - ADS's heavier client-side application leads to slower early performance.
  - **Interactivity**
    - Both apps perform similarly in interactivity after the page is loaded.
    - SciX's key opportunity lies in optimizing initial interactivity, such as making the search bar usable faster (~2.5 seconds delay).
  - **Competitive Comparison**
    - In **Core Web Vitals** and **Page Weight**, both SciX and ADS perform competitively relative to other apps.
    - Both apps have room for improvement to better compete.
  - **Performance Context**
    - SciX edges out ADS in several tests, especially from full page load.
    - Performance post-load depends largely on API response times, where both apps perform relatively closely.
- **Conclusions**
  - **SciX Strengths**
    - Performs well on initial load due to its light payload.
    - Outpaces ADS in key full-load metrics.
  - **SciX Opportunities**
    - Interactivity is delayed until full hydration; optimizing server-side rendering, caching, and prefetching is critical.
  - **ADS Opportunities**
    - Improve early load performance by front-loading critical assets and reducing extraneous assets.
  - **Overall**
    - The main win lies in optimizing the initial load, reducing unnecessary assets (Page Weight), and improving interactivity phases for both apps.

Wrapping Up & Next Steps

- **Opportunities**
  - **Reduce Blocking Time and Improve Interactivity**
    - Minimize JavaScript and CSS usage per page to reduce CPU task length and load times.
    - Use tooling to assess and reduce unused code.
  - **Optimize Key Elements**
    - **ADS**: Focus on loading critical assets early for faster visibility of key content.
    - **SciX**: Ensure search bar is either interactive upon load or appears only when usable (~2.5s current delay).
  - **Extraneous Asset Reduction**
    - ADS should eliminate unnecessary assets to improve load times.
- **Next Steps**
  - **Continuous Integration**
    - Regularly measure Core Web Vitals (e.g., LCP, FID) and custom performance timings.
  - **Sentry.io Instrumentation**
    - Use traces and replays to identify bottlenecks and improve performance.
    - Leverage custom timings to fill knowledge gaps and track user experience trends.
  - **Dashboard Creation**
    - Consolidate metrics such as:
      - Core Web Vitals over time.
      - Lighthouse scores over time.
      - Active user data (Google Analytics).
      - Internal timings (e.g., "time to results").
  - **Focus Areas**
    - Prioritize optimizations in early load and interactivity phases for both apps.
    - Use tooling to streamline asset delivery and improve critical path rendering.