PR #2794 完整报告

radixark/miles

dashboard: scroll the token strip instead of paging through it

合并时间：2026-08-29 15:02

原文链接：http://prhub.com.cn/radixark/miles/pull/2794

执行摘要

本 PR 把 dashboard 样本视图的令牌条从“1024-token 分页窗口”改为“整条连续滚动”。一次全量 /tokens 请求同时喂给图表与令牌条，把面板的接口调用从 3 次降到 1 次；同时移除 ◀ / ▶ 与 window / start 控件，颜色标尺改为全量计算，切换指标只重着色不重建 DOM，并自动滚动到第一个生成 token。第二个提交修复了 review 指出的隐藏面板下 offsetTop 为 0 导致滚动失效的问题，由 claude[bot] 最终 Approving。无后端、schema 与 Python 测试变更。

功能与动机

PR body 点明问题：长回复被切成 1024-token 一页，看到尾部要点击 24 次 ◀ / ▶，而 window / start 输入框是唯一能定位中间位置的方式。同时，令牌条下方的指标图本来就拉全量区间（start=0, end=total），窗口化的令牌条等于重复请求服务器已在发送的字节。作者在 64,006-token 样本上实测：全量 /tokens 拉取约 920 ms（图表已支付）、DOM 构建 64,006 个 span 仅 66 ms、堆约 29 MB，因此分页从未换来过渲染收益。

实现拆解

1. 数据读取简化：在 miles/dashboard/static/views_tokens.js 的 loadTokensPane 中，用一次全量请求替代原 probe + windowed read 两次请求，删除 WINDOW_SIZES 与 start、windowSize 状态；available 统计提前到加载入口，stat 默认值改为从可用列表中选取。原因是图表本来就拉全量区间，窗口化只是重复请求同一批字节。
2. 渲染全量一次构建：paintStrip 根据 payload 生成全部 token span，掩码判断收敛到 isMasked helper，并让初次构建与 color by 重绘两处共用；切换指标时只重着色、不重建 DOM，因此滚动位置不再丢失。
3. 滚动盒布局：style.css 中 .tokens 增加 max-height: 480px、overflow-y: auto、position: relative，使 .tokens 成为其子 token 的 offsetParent；这样 token 的 offsetTop 就是把它滚到盒顶所需的 scrollTop，可以直接定位。
4. 面板生命周期修正：两个 pane 始终挂载、用 hidden 切换，避免 detach 导致 scrollTop 重置；tokensPane._onShown 一次性回调负责“有布局后再滚动到首个生成 token 并渲染图表”，select() 取消隐藏后重新触发，这正是 review 提出的 offsetTop 时序问题的修复方案。
5. 配套与验证：README.md 更新 sample view 描述；仓库没有 JS 测试 harness，作者用 Playwright 驱动真实 Chrome 对 64,006-token 样本回归，覆盖 eval 视图、无 conversation sidecar、全 masked、10-token 等场景。

miles/dashboard/static/views_tokens.js

核心改动文件：移除分页状态，一次全量读取，新增 isMasked / paintStrip / _onShown 逻辑，token 条从窗口化改为整条滚动。

```
// 关键实现：令牌条从“分页窗口”改为“整条滚动”，并处理好隐藏面板的布局时机。
// 注释按 PR 描述整理，省略无关分支。

// 掩码判定：loss 掩码非 0 的位置在绘制时调暗，且不参与按指标着色
function isMasked(payload, index) {
  const mask = payload.loss_mask?.[index];
  return mask != null && mask !== 0; // 0 表示无掩码
}

// 在 loadTokensPane 内部：创建滚动盒并一次性构建全部 span
const strip = el('div', { class: 'tokens' });
const firstResponse = paintStrip(strip, payload, stat); // 返回首个生成 token 的 span

// 两个 pane 始终挂载、用 hidden 切换：一旦把 tokensPane 从文档摘除，
// 浏览器会丢弃它的布局并把 scrollTop 重置为 0
const select = (name) => {
  conversationPane.hidden = name !== 'conversation';
  tokensPane.hidden = name !== 'tokens';
  if (name === 'tokens') {
    startTokens(); // 懒加载只执行一次
    tokensPane._onShown?.(); // 补偿面板隐藏期间完成的延迟加载
  }
};

// 打开时直接滚到第一个生成 token；agentic prompt 可能长达数千 token，
// 从位置 0 出发会让读者离值得看的内容太远
tokensPane._onShown = () => {
  // hidden 状态下 getClientRects() 为空且 offsetTop 恒为 0，
  // 此刻什么都不做，等 select() 取消隐藏后再触发一次
  if (!tokensPane.getClientRects().length || tokensPane._scrollApplied) return;
  tokensPane._scrollApplied = true; // 一次性：后续切标签不再重滚
  strip.scrollTop = firstResponse.offsetTop; // .tokens 是 offsetParent
};
```

评论区精华

> claude[bot]：自动滚动依赖 firstResponse.offsetTop，而 tokens 面板在加载期间若被切到 Conversation 标签就会从文档 detach，offsetTop 为 0；加载完成后切回 Tokens，整条序列显示但永久停在 scrollTop 0。
>
> claude[bot]（修复后）：Approving — 时序问题已修复：滚动与图表渲染进入一次性 _onShown 回调，由 root.getClientRects().length 守卫，select() 在取消隐藏后重新触发，所以它只会在面板真正拥有布局后运行。

Zhichenzzz 在 PR 评论中表示“Great fix!”，并认为 Claude Code 的 review 值得一看，随后给出 APPROVED。

风险与影响

- DOM 规模：每个 token 一个 span，64k 样本约 29 MB 堆；几十万 token 的超长样本可能推高内存与样式重算开销。
- 布局时序：滚动定位依赖 offsetTop，hidden 场景已由 _onShown 修复；未来若绕过 select() 直接挂载 tokensPane，回调可能不触发。
- 测试覆盖：仓库无 JS 测试 harness（无 package.json），本次验证靠 Playwright 手工回归，后续回归缺少自动防线。
- 事件性能：tooltip 仍为 per-span 监听，mousemove 在 64k span 上约 66 ms；作者评估委托方案只省 6 ms，不值得承担回归风险，但超长样本仍可能卡顿。

关联脉络

历史 PR 中没有直接改动 miles/dashboard 静态文件的记录，本 PR 属于 dashboard 前端独立演进；但它与 rollout 指标观测链路同向：PR#2710、#2743、#2765 持续在 rollout 侧完善指标数据（compaction-aware metrics、非数值奖励跳过、per-rollout 响应长度），本 PR 让样本视图能一次拿到全量指标序列并稳定着色、快速定位，二者合起来构成“指标生产 → 指标可视化”的完整闭环。