**开平主应用CRP优化 - prefetch chunk**

**业务背景**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 因为主应用加载路由前，依赖这些接口的加载，会阻塞路由和子应用的加载。在请求接口等待服务器响应这段时间，用matchPath匹配到当前路由提前执行资源加载。   * 主应用路由，根据路由配置执行router的import方法。 * 子应用，每次菜单接口返回都把结果存储到localstroage，覆盖更新。下次打开页面的时候如果匹配到路由则执行garfish preload。      * **优化前**  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | 主应用chunk下载 |  |  |  | |  | Prefetch API |  |  | |  |  | 主应用路由chunk加载 |  | |  |  |  | 子应用chunk下载 | |  |  * **优化后**  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | 主应用chunk下载 |  | |  | Prefetch API | |  | 主应用路由chunk加载 | |  | 子应用chunk下载 | |  | |

**实现**

**子应用chunk加载**

使用react-router-dom匹配到当前路由配置执行router的import方法，提前加载路由chunk。

|  |
| --- |
| TypeScript function prefetchMainAppRoute(appRoutes: any[]) {  appRoutes.forEach(el => {  // https://github.com/remix-run/react-router/blob/fbb6358dd7f47eedd55d0b63e0725ac335d28bde/packages/react-router/modules/Route.js#L41  const match = matchPath(location.pathname, el);   if (match && el.routeMatchPrefetch) {  try {  console.log('[prefetch main route chunk]', el.path);  el?.component?.();  } catch (e) {  console.log('[prefetch main route chunk error]', el.path);  }  }  }); } |

**主应用路由chunk加载**

每次菜单接口返回都把结果存储到localstroage，覆盖更新。下次打开页面的时候如果匹配到路由则执行garfish preload。

|  |
| --- |
| TypeScript function getMenu() {  // 获取上次保存的路由配置  const gameMenu = safeParseJson(  localStorage.getItem(`${saveMenuKeyPrefix}game`) || '',  [],  );  const miniGameMenu = safeParseJson(  localStorage.getItem(`${saveMenuKeyPrefix}mini\_game`) || '',  [],  );   return [...gameMenu, ...miniGameMenu]; }  function prefetchSubAppRoute() {  getMenu().forEach(el => {  const match = matchPath(location.pathname, el as any);  if (match) {  console.log('[prefetch subApp route chunk]', el);  prefetchloadAppResource({  entry: el.appEntry,  name: el.appName,  });  }  }); } |

**结果**

**PPE**

|  |
| --- |
| ppe浮动比较大 |

|  |  |
| --- | --- |
|  |  |

**ONLINE**

显著降低LCP指标，但是对整体的性能有劣化（有劣化的MR合入），因为FMP时间段执行的js和加载的资源增加了。

