

## Latency Optimization Recommendations:

### 1. Decouple metric ingestion from your GET:

- Move fetching from agents and inserting into server\_metrics into a scheduled background process (cron).
- Simplify GET /api/servers to only query Supabase for latest metrics, avoiding HTTP fetches and inserts.

### 2. Index & aggregate in Supabase:

- Add an index on (server\_id, recorded\_at) in server\_metrics table for fast lookups.
- Consider a materialized view or separate table holding only the most recent metrics per server.

### 3. Optimize client-side fetching with SWR:

```
const { data, error } = useSWR('/api/servers', fetcher, {  
  refreshInterval: 60000, // refetch once per minute  
  dedupingInterval: 30000, // dedupe requests within 30s  
});
```

### 4. Leverage Next.js caching (ISR):

```
// in app/api/servers/route.ts  
export const revalidate = 60 // cache route for 60s
```

### Putting it all together:

- Create a cron job or scheduled function for metrics ingestion.
- Simplify API GET to only query Supabase.
- Index your metrics table.
- Use SWR refresh/dedup options.
- Optionally, use ISR with export const revalidate = 60.