Introduction
Network protocols can be designed to enable a reduction in energy use of data servers. We architect a new HTTP timed redirection response:

- GET requests are redirected to another server with a delay.
- Used for load balancing [1].
- Time shifting increases the response time of a GET request.
- For some applications there is a degree of delay tolerance that can be exploited.

Examples: automatic app updates, and large file downloads

System Design for HTTP Timed Redirection
In a hybrid web server (proposed in [2]),
- All HTTP requests are sent to the Assistant
- Assistant uses Timed redirection to delay requests arriving during the time Master is sleeping (duration $T_{notServe}$).
- These requests will be re-arrived at a later time when the Master is not sleeping (duration $T_{serve}$)

Timed Redirection is done by a new Retry-Delay field in seconds similar to the existing Retry-After field in HTTP 3xx responses. The browser or download agent will send the same request again to the new location (Master) in Retry-Delay seconds.

System Design for HTTP Timed Redirection (continued)

In Fig. 2, each epoch of duration $T_{epoch}$ consists of:
- a not serving period ($T_{notServe}$, Master sleeping or transitioning), and
- a serving period ($T_{serve}$, Master awake and serving).

Requests that arrive during a serving period are redirected (to be served by the Master) with no delay. Requests that arrive during $T_l$ to $T_l + T_{notServe}$ are redirected with a delay, $T_{delay}$ calculated such that the redirected request will occur in the immediately following awake period.

\[
T_{delay} = \begin{cases} 
T_{notServe} - t & \text{if } t \leq T_{notServe} \\
\frac{t}{T_{notServe}} \cdot T_{max} & \text{if } t > T_{notServe}
\end{cases}
\]

Distribution of arrivals during $T_l$ to $T_l + T_{notServe}$ is preserved, but scaled to $T_l + T_{notServe}$ to $T_l + T_{notServe} + T_{max}$.

Experimental Evaluation
Some of the parameters used in evaluation:
- **Evaluation Method:** Prototype; Dell OptiPlex 790 (Master), SheevaPlug (Assistant)
- **Modified Apache mod rewrite rules** and wget to implement Timed Redirection
- **Response variables:** Master average power draw and mean wait to serve a request

**Epoch experiment** (Fig. 4): the effects of the duration of an epoch on energy savings and delay
**Duty cycle experiment** (not shown): the effects of duty cycle on energy savings and delay

The average power draw tends to the duty cycle $D = T_{notServe} / T_{epoch}$, multiplied by the average power draw when always on as epoch length increases.

More details on evaluation in the paper

Summary and Next Steps
A new timed redirection protocol can be used to save energy in hybrid servers by coalescing HTTP GET requests in a distributed manner. Next steps:
- Identifying applicable application and their tolerable delays
- Investigating adaptive scheduling of requests

Selected References: