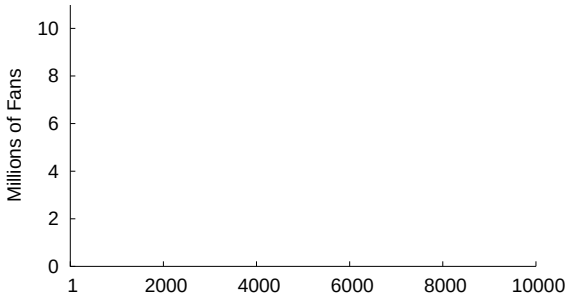
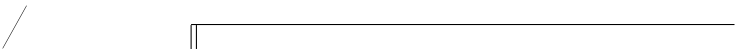


Figure 1: The popularity of web data often exhibits a long heavy tail of equally unpopular objects. This graph ranks the popularity of Facebook group pages by their number of fans.

1.1 Example: BlinkCache

To demonstrate how blinking impacts a common data center application, we explore the design of BlinkCache—a blinking version of *memcached* that gracefully handles intermittent power constraints—as a proof-of-concept example. Memcached is a distributed memory cache for storing key-value pairs that many promi-



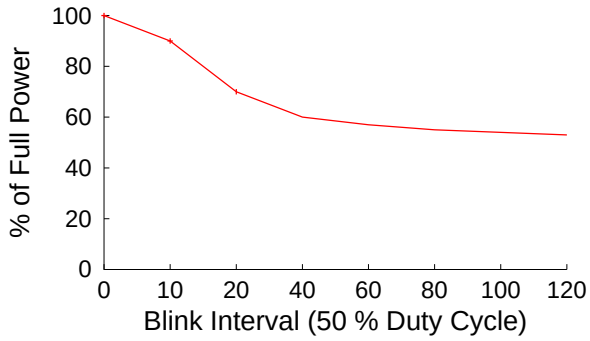
Type

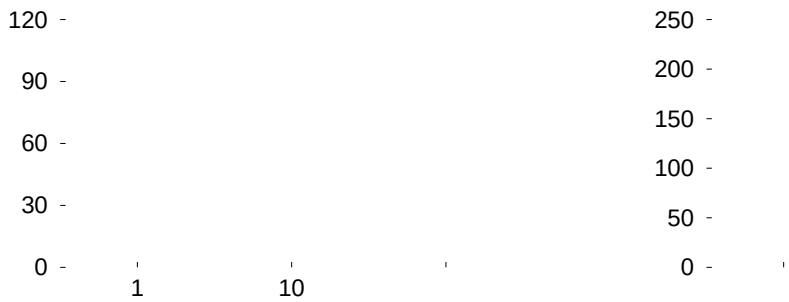
Figure 4: The popularity rank, by number of fans, for all 20 million public group pages on Facebook follows a Zipf-like distribution with $\alpha = 0.6$.

a corresponding update in persistent storage. Each memcached server uses the Least Recently Used (LRU) replacement policy to evict objects. One common example of a cached object is an HTML fragment generated from the results of multiple queries

Figure 5: To explicitly control the mapping of keys to servers, we interpose always-active request proxies between memcached clients and servers. The proxies are able to monitor per-key hit rates and migrate similarly popular objects to the same nodes.

server. One way to correct the problem is to dynamically change the built-in client mapping function to only reflect the current set


100
80
60
40
20
0
% of Full Power
0
10
20
40
60
80
100
120
Blink Interval (50 % Duty Cycle)


120
90
60
30
0
1
10

250
200
150
100
50
0

storage research, such as migrating popular objects to more active nodes [28, 41]. Additionally, power-aware caching algorithms focus on maximizing the idle time between disk accesses to reduce disk power consumption, while our work focus on controlling the power consumption of the cache itself [42].

Blinking introduces regulated churn into data center applications as nodes switch from the active to inactive state. Churn has been well-studied in decentralized, self-organizing distributed hash tables [34]. However, the type of churn experienced by DHTs is different than the churn caused by blinking, which motivates our different approach to the problem. In the former case, nodes arrive and depart unexpectedly based on autonomous user behavior and