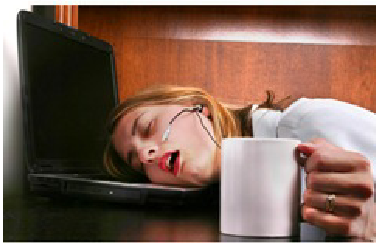
&lsquo;Drowsy&rsquo; Networking

Lori MacVittie, 2009-11-11

No, not the kind you do on Facebook when you're really, really tired but the kind defined as a means to reduce power consumption without affecting application performance or availability by eliminating non-essential processing and networking whenever possible.



An article on "Drowsy" computing as a means to reduce power consumption in data centers got me thinking about how such concepts might be applied to networking.

To summarize the concept of "drowsy" computing its basic premise is that when applications aren't being heavily used some mechanism is used to reduce the power consumption on the physical server to its lowest levels, thereby saving costs associated with drawing power. The CEO of 1e, which offers a product to provide the mechanism by which power consumption levels on servers are manipulated, says the concept "*can make a significant dent in what is currently more than $4 billion in wasted energy use every year."*

The trick is apparently differentiating between "useful" and "housekeeping" computing. Thus the trick to accomplishing similar behavior in networking would be to distinguish between the "useful" and "housekeeping" networking. But we also need to bear in mind that changes in the network or application network architecture as a means to reduce resource and power consumption should be automated lest the financial gains end up negated by the cost of manually carrying out such tasks.

Turns out there are some interesting applications of this concept, especially in the application network management arena, that fulfill both.

NETWORK HOUSEKEEPING

In application delivery, at least, there are a couple of tasks that might be good candidates for "slowing down" during periods of less frequent activity. These tasks generally consume resources on the application network components, the network and its associated components, *and* the web/application/database servers. For applications that are primarily business-related and for which usage is highly predictable, i.e. used only during business hours or has known patterns of use based on time of day or other events, these housekeeping tasks can be better regulated to reduce the power and usage of all the components involved across the data center.

- **Application Health Monitoring** This is the cornerstone block in the application availability house so tread lightly in this area but definitely venture inside. For applications known to be in use only during business hours health checks should be as frequent as necessary to ensure availability and fault-tolerance during usage but after hours can be relaxed somewhat. Health checks, regardless of whether they're simple ICMP or (as is proper) at the application layer, make use of application network, network, server, and application resources. By reducing the frequency with which these checks are made during off hours from an interval measured in seconds to one measured in perhaps minutes you can reduce the consumption of resources across the entire data center. It's hard to measure the amount of power saved across all components and on a per-health check basis is probably minute, but over time that all adds up.
- **Available Server Reduction** If the usage patterns of an application are fairly predictable then you know that during certain time intervals you'll need X servers and at other times you'll need Y. If those intervals can be measured in hours then it might be advantageous to remove servers from the pool (cluster or farm) of available servers during light usage periods. This effectively removes the health monitoring checks and ensures that no matter what load balancing algorithm is being used that requests will not be distributed across all servers. This reduces resource utilization in general and it's a fairly simple task to add those servers back into the pool before usage increases

usage increases.

- **Caching** Caching requires that you better understand the update frequency of data being requested by applications, but if it’s the case that content is updated only once every X hours or even not at all during the evening hours then caching can drastically reduce the consumption of resources across the data center. By employing the caching capabilities of application delivery solutions you can offload requests from the network and servers. If the application delivery solution is flexible you can further modify caching policies on-demand to dynamically adapt to a schedule of caching for shorter or longer periods of time, depending on how often content is updated. You’re removing the housekeeping task of checking with the origin server to see if content “might be updated” because you know already that it hasn’t.

I’m certain there are more housekeeping tasks that can be evaluated for potential modification as a means to reduce consumption of resources. I’m also fairly certain that you won’t see a huge reduction in costs associated with such actions, but combined with other cost and power savings measures it’s a step in the right direction.

- Putting a Price on Uptime
- Web Application Security at the Edge is More Efficient Than In the Application
- Hardware Acceleration Critical Component for Cost-Conscious Data Centers
- Telecommute your way to a greener bottom line
- Saving the world, one server at a time
- Green IT: 404 Blacklisting
- Green IT: More is Less
- Green IT: You don't have to sacrifice speed
- Green IT: Reduce, Reuse, Recycle