LiteGreen
Saving Energy in Networked Desktops using Virtualization

Tathagata Das, Pradeep Padala, Venkat Padamanabhan, Ram Ramjee, Kang G. Shin
Have you switched off your desktop?
Have you switched off your desktop?

Key observation
Users do not like disruption
Automated energy saving is needed

[Greener PCs for the Enterprise, Nordman’06]

250 TWh

[Greener PCs for the Enterprise, Nordman’06]

- 26% of 250 TWh is used by Enterprise PCs.

[Greener PCs for the Enterprise, Nordman’06]

- Enterprise PCs: 26% of 250 TWh = 100 TWh
- Other PCs: 14% of 250 TWh = 35 TWh

[Greener PCs for the Enterprise, Nordman’06]

- 100 TWh
  - Enterprise PCs: 26%
  - Other PCs: 14%

250 TWh
- Other Electronics

[Greener PCs for the Enterprise, Nordman’06]
How much energy can we save?

- Data gathering for ~120 users, MSRI, Summer’09
- ~65000hrs worth of data
How much energy can we save?

- Data gathering for ~120 users, MSRI, Summer’09
- ~65000hrs worth of data
How much energy can we save?

- Data gathering for ~120 users, MSRI, Summer’09
- ~65000hrs worth of data
How much energy can we save?

- Data gathering for ~120 users, MSRI, Summer’09
- ~65000hrs worth of data
How much energy can we save?

- Data gathering for ~120 users, MSRI, Summer’09
- ~65000hrs worth of data

Key observation

Significant energy savings by exploiting short idle periods!
Existing Approaches and Related Work
Existing Approaches and Related Work

- **Windows Power Management**
  + Trivial inactivity easily handled
  - Disruption to existing applications, cannot wakeup remotely
  - Confused by spikes, night time traffic, background processes

- **Protocol agnostic proxy** [uses Wake-on-LAN mechanism]
  + Supports remote wakeup, stateless
  - Manual, works only in subnet, disruption to existing applications

- **Protocol proxy** [Nedevschi et.al. NSDI’09]
  + Automatic: Protocol Specific Rules (ICMP, ARP, RDP, …)
  - Disruption to existing applications, deployment problems

- **Application proxy** [Agarwal et.al. NSDI’09]
  + Reduces disruption to applications that have proxies
  - Increased complexity in writing application specific proxies
#### Existing Approaches and Related Work

- **Windows Power Management**
  - + Trivial inactivity easily handled
  - – Disruption to existing applications, cannot wakeup remotely
  - – Confused by spikes, night time traffic, background processes

- **Protocol agnostic proxy** [uses Wake-on-LAN mechanism]
  - + Supports remote wakeup, stateless
  - – Manual, works only in subnet, disruption to existing applications

- **Protocol proxy** [Nedevschi et.al. NSDI’09]
  - + Automatic: Protocol Specific Rules (ICMP, ARP, RDP, …)
  - – Disruption to existing applications, deployment problems

- **Application proxy** [Agarwal et.al. NSDI’09]
  - + Reduces disruption to applications that have proxies
  - – Increased complexity in writing application specific proxies
Existing Approaches and Related Work

• **Windows Power Management**
  + Trivial inactivity easily handled
  – Disruption to existing applications, cannot wakeup remotely
  – Confused by spikes, night time traffic, background processes

• **Protocol agnostic proxy** [uses Wake-on-LAN mechanism]
  + Supports remote wakeup, stateless
  – Manual, works only in subnet, disruption to existing applications

• **Protocol proxy** [Nedevschi et.al. NSDI’09]
  + Automatic: Protocol Specific Rules (ICMP, ARP, RDP, …)
  – Disruption to existing applications, deployment problems

• **Application proxy** [Agarwal et.al. NSDI’09]
  + Reduces disruption to applications that have proxies
  – Increased complexity in writing application specific proxies
Existing Approaches and Related Work

• **Windows Power Management**
  + Trivial inactivity easily handled
    – Disruption to existing applications, cannot wakeup remotely
    – Confused by spikes, night time traffic, background processes

• **Protocol agnostic proxy** [uses Wake-on-LAN mechanism]
  + Supports remote wakeup, stateless
    – Manual, works only in subnet, disruption to existing applications

• **Protocol proxy** [Nedevschi et.al. NSDI’09]
  + Automatic: Protocol Specific Rules (ICMP, ARP, RDP, ...)
    – Disruption to existing applications, deployment problems

• **Application proxy** [Agarwal et.al. NSDI’09]
  + Reduces disruption to applications that have proxies
    – Increased complexity in writing application specific proxies
Existing Approaches and Related Work

- **Windows Power Management**
  - + Trivial inactivity easily handled
  - – Disruption to existing applications, cannot wakeup remotely
  - – Confused by spikes, night time traffic, background processes

- **Protocol agnostic proxy** [uses Wake-on-LAN mechanism]
  - + Supports remote wakeup, stateless
  - – Manual, works only in subnet, disruption to existing applications

- **Protocol proxy** [Nedevschii et.al. NSDI'09]
  - + Automatic: Protocol Specific Rules (ICMP, ARP, RDP, …)
  - – Disruption to existing applications, deployment problems

- **Application proxy** [Agarwal et.al. NSDI'09]
  - + Reduces disruption to applications that have proxies
  - – Increased complexity in writing application specific proxies

**Key observation**
Disruption to applications and lack of generality are major limitations
Power management spectrum

Maintenance overhead

Energy savings
Power management spectrum

- Windows PM
  - Automated
  - Low savings
Power management spectrum

- **Windows PM**
  - Automated
  - Low savings

- **WoL**
  - Manual
  - User disruption
Power management spectrum

- **Windows PM**
  - Automated
  - Low savings

- **WoL**
  - Manual
  - User disruption

- **Proxying**
  - Automated
  - App stubs

- Maintenance overhead vs. Energy savings
Power management spectrum

- Holy Grail
- Proxying
  - Automated
  - App stubs
- WoL
  - Manual
  - User disruption
- Windows PM
  - Automated
  - Low savings

Energy savings vs. Maintenance overhead
**Key research question and goals**

How to **automatically** save energy?

- Minimal performance disruption
- Minimal maintenance
LiteGreen: virtualization and consolidation
LiteGreen: virtualization and consolidation

Hardware
LiteGreen: virtualization and consolidation

Virtual Machine

Virtualization

Hardware
LiteGreen: virtualization and consolidation

Virtualization

Hardware

Server

- Active VM
- Idle VM
- Machine off
LiteGreen: virtualization and consolidation

Virtual Machine

Virtualization

Hardware

Server

Active VM

Idle VM

Machine off
LiteGreen: virtualization and consolidation

- Virtual Machine
  - Virtualization
  - Hardware

- Server
  - Active VM
  - Idle VM
  - Machine off
1. Provide normal, undisrupted desktop experience
   – How to make the machine usable instantly?
   – How to mask the effect of VMs and their migration?
   – **Solution**: Indirection (RDP) + live migration

2. Automated Energy Savings
   – When to migrate?
   – Which VMs to migrate?
   – **Solution**: Savings algorithm driven by data analysis

3. Evaluation
   – Hyper-V prototype and small-scale deployment
   – Trace-driven analysis
LiteGreen Contributions

1. Provide normal, undisrupted desktop experience
   – How to make the machine usable instantly?
   – How to mask the effect of VMs and their migration?
   – **Solution:** Indirection (RDP) + live migration

2. Automated Energy Savings
   – When to migrate?
   – Which VMs to migrate?
   – **Solution:** Savings algorithm driven by data analysis

3. Evaluation
   – Hyper-V prototype and small-scale deployment
   – Trace-driven analysis
LiteGreen Contributions

1. Provide normal, undisrupted desktop experience
   – How to make the machine usable instantly?
   – How to mask the effect of VMs and their migration?
   – **Solution**: Indirection (RDP) + live migration

2. Automated Energy Savings
   – When to migrate?
   – Which VMs to migrate?
   – **Solution**: Savings algorithm driven by data analysis

3. Evaluation
   – Hyper-V prototype and small-scale deployment
   – Trace-driven analysis
LiteGreen Contributions

1. Provide normal, undisrupted desktop experience
   – How to make the machine usable instantly?
   – How to mask the effect of VMs and their migration?
   – **Solution**: Indirection (RDP) + live migration

2. Automated Energy Savings
   – When to migrate?
   – Which VMs to migrate?
   – **Solution**: Savings algorithm driven by data analysis

3. Evaluation
   – Hyper-V prototype and small-scale deployment
   – Trace-driven analysis
1. Provide normal, undisrupted desktop experience
   – How to make the machine usable instantly?
   – How to mask the effect of VMs and their migration?
   – **Solution**: Indirection (RDP) + live migration

2. Automated Energy Savings
   – When to migrate?
   – Which VMs to migrate?
   – **Solution**: Savings algorithm driven by data analysis

3. Evaluation
   – Hyper-V prototype and small-scale deployment
   – Trace-driven analysis
Talk outline

• Background and Motivation
• What is LiteGreen?
• Design and Implementation (demo)
• Evaluation
  – Trace-driven Analysis
  – Hyper-V Deployment
• Limitations and Future Work
• Conclusion
Problem: How to mask the effect of migration?
Problem: How to mask the effect of migration?
Solution: RDP client + Live migration
Server

Solution: RDP client + Live migration

Virtual Machine

Virtualization

Hardware

Server

Active VM

Idle VM
Server LiteGreen Operation

Solution: RDP client + Live migration

Virtualization

Hardware

Active VM

Idle VM

Server
LiteGreen Operation

Solution: RDP client + Live migration

RDP client

Virtual Machine

Virtualization

Hardware

Server

Active VM

Idle VM
**Solution**: RDP client + Live migration

- **RDP client**
- **Virtual Machine**

---

- **Virtualization**
- **Hardware**

- **Server**

- **Active VM**
- **Idle VM**
Solution: RDP client + Live migration

RDP client

Virtual Machine

Virtualization

Hardware

Server

Active VM

Idle VM
Server

LiteGreen Operation

Virtualization

Solution: RDP client + Live migration

Hardware

RDP client

Virtual Machine

Server

Active VM

Idle VM
Solution: RDP client + Live migration Few seconds
Solution: RDP client + Live migration

Few seconds

Live Migration

Virtualization

Hardware

Server

Active VM

Idle VM
**Solution:** RDP client + Live migration

- **RDP client**
- **Virtual Machine**
- **Virtualization**
- **Hardware**
- **Server**
- **Active VM**
- **Idle VM**
LiteGreen in action (demo)
Savings algorithm – Intuition

Active VM = (User is interactive) OR (Resource usage > ResThr)
Savings algorithm – Intuition

Active VM = (User is interactive) OR (Resource usage > ResThr)
Savings algorithm – Intuition

Active VM = (User is interactive) **OR** (Resource usage > **ResThr**)

- **ACTIVE**
- **MIGRATING**

VM not active for **IdleThr** minutes
Savings algorithm – Intuition

Active VM = (User is interactive) OR (Resource usage > ResThr)

VM is active
migration is cancelled

VM not active for
IdleThr minutes

ACTIVE → MIGRATING
Savings algorithm – Intuition

Active VM = (User is interactive) OR (Resource usage > ResThr)

VM is active migration is cancelled

VM not active for IdleThr minutes

migration finishes

ACTIVE ➔ MIGRATING ➔ IDLE
Savings algorithm – Intuition

Active VM = (User is interactive) OR (Resource usage > ResThr)
Savings algorithm – Intuition

Active VM = (User is interactive) OR (Resource usage > ResThr)

How to choose ResThr and IdleThr? Statistical analysis
Savings Algorithm - Policies

- idle VMs *could* be pulled into the server, subject to capacity ["eligible for pull" list]
- active VMs *must* be pushed back to the desktop machine ["mandatory to push" list]
Two Policies

1. **Conservative**: 
   Active VM = (User is interactive) OR (Resource usage > ResThr)

2. **MaxSavings**: Active VM = (User is interactive)
Talk outline

• Background and Motivation
• What is LiteGreen?
• Design and Implementation (demo)
• Evaluation
  – Trace-driven Analysis
  – Hyper-V Deployment
• Limitations and Future Work
• Conclusion
Trace driven analysis: Energy savings
Trace driven analysis: Energy savings

- LG MaxSavings: 34.82%
- LG Conservative: 37.32%
- Existing: 14.49%

Sleep Time (%)

0 10 20 30 40 50 60 70 80 90 100
Trace driven analysis: Energy savings

- Data gathered at MSR India, 120 machines
- 65,000 user-hours of data
- Existing power management
  - Manual switching off
  - Windows PM plans
    - Balanced (most users)
    - 30 min IdleThr (unclear what idle means)
    - Few users have high performance plan
- LiteGreen thresholds
  - CPUShr = 10%, IdleThr = 5min
Trace driven analysis: Energy savings

- Data gathered at MSR India, 120 machines
- 65,000 user-hours of data
- Existing power management:
  - Manual switching off
  - Windows PM plans
    - Balanced (most users)
    - 30 min IdleThr (unclear what idle means)
    - Few users have high performance plan
- LiteGreen thresholds:
  - CPUThr = 10%, IdleThr = 5 min

<table>
<thead>
<tr>
<th>Sleep Time (%)</th>
<th>Existing</th>
<th>LiteGreen Conservative</th>
<th>LiteGreen MaxSavings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekend Night</td>
<td>34.82</td>
<td>14.49</td>
<td>17.38</td>
</tr>
<tr>
<td>Weekend Day</td>
<td>37.32</td>
<td>4.66</td>
<td>5.72</td>
</tr>
<tr>
<td>WeekNight</td>
<td>14.49</td>
<td>17.38</td>
<td>6.63</td>
</tr>
<tr>
<td>WeekDay</td>
<td>37.32</td>
<td>15.65</td>
<td>21.37</td>
</tr>
</tbody>
</table>

MaxSavings: 14.49% (Weekend Night), 15.65% (Weekend Day), 17.38% (WeekNight), 21.37% (WeekDay)
User who is always on ...
User who is always on ...

Many short idle periods
Spikes cause problems for existing PM
User who is always on ...

**CPU utilization**

**Sleep Time (%)**

- **Existing**
  - Weekend Night: 2.85
  - Weekend Day: 5.33
  - Week Night: 1.97
  - Week Day: 14.2
  - Total: 23.71

- **LiteGreen**
  - Weekend Night: 8.85
  - Weekend Day: 7.06
  - Week Night: 29.98
  - Week Day: 23.71
  - Total: 80.6
User who manually switches off

Nothing can beat manual PM
User who manually switches off

**CPU utilization**

- **Existing**
  - Weekend Night: 9.03
  - Weekend Day: 9.03
  - Week Night: 34.59
  - Week Day: 5.93

- **LiteGreen**
  - Weekend Night: 0.78
  - Weekend Day: 3.05

**Sleep Time (%):**

- **Weekend Night:** 9.03%
- **Weekend Day:** 9.03%
- **Week Night:** 34.59%
- **Week Day:** 5.93%
Prototype with Hyper-V and Xen

• A 2-node server cluster
  – One server node (32GB RAM + 8 cores CPU) running Windows 2008 R2 + Hyper-V R2
  – One storage node (2x100GB disks) accessed through iSCSI

• 10 user desktops
  – Mixture of researchers and non-researchers
  – USENIX paper submission done using two of the desktops

• All nodes connected on 1Gbps LAN

• Memory ballooning and consolidation experiments with Xen
Deployment results - # of migrations

- **Holiday Night**
- **Holiday Day**
- **Weekday Night**
- **WeekDay Day**

### Legend
- Purple (Holiday Night)
- Light Green (Holiday Day)
- Red (Weekday Night)
- Blue (WeekDay Day)

### Days in Deployment

<table>
<thead>
<tr>
<th>Days in Deployment</th>
<th>No. of Migrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
</tr>
</tbody>
</table>

- **No. of Migrations**
  - 0
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8
  - 9
  - 10
  - 11
  - 12
  - 13
  - 14
  - 15
  - 16
  - 17
  - 18
  - 19
  - 20
  - 21
  - 22
  - 23
  - 24
  - 25
  - 26
  - 27
  - 28
  - 29
  - 30
  - 31
  - 32
  - 33
  - 34
  - 35
  - 36
  - 37
  - 38
  - 39
  - 40
  - 41
  - 42
  - 43
  - 44
  - 45
  - 46
  - 47
  - 48
  - 49
  - 50

- **Days in Deployment**
  - 1
  - 4
  - 7
  - 10
  - 13
  - 16
  - 19
  - 22
  - 25
  - 28
Deployment results - # of migrations

- LiteGreen can exploit idle periods during both **day and night**
Deployment results - # of migrations

- LiteGreen can exploit idle periods during both day and night

Total Energy Savings: 71%
• Every 30 minutes memory of desktop VM reduced by 128MB
• Adaptive mechanism based on page faults can be developed
Talk outline

• Background and Motivation
• What is LiteGreen?
• Design and Implementation (demo)
• Evaluation
  – Trace-driven Analysis
  – Hyper-V Deployment
• Limitations and Future Work
• Conclusion
Limitations and Future work

• Virtualization
  – GPU virtualization. Client hypervisors can help.
  – Shared storage based live migration. Live migration with local disk state can help.

• Evaluating consolidation of idle VMs
  – How much does memory sharing help?
  – What metrics to be used for performance disruption?

• Combining with
  – Software only approaches. SleepServer: A Software-Only Approach for Reducing the Energy Consumption of PCs within Enterprise Environments [Agarwal et.al. USENIX ATC’10]
  – Proxying. Sleepless in Seattle No Longer [Reich et. al. USENIX ATC’10]
Summary: LiteGreen in one slide

- RDP client
- VM
- Live Migration
- Virtualization
- Hardware
- Server
1. Significant energy savings exist in short idle periods
1. Significant energy savings exist in **short idle periods**

2. Holygrail = High manageability + High savings
1. Significant energy savings exist in *short idle periods*
2. Holygrail = High manageability + High savings
3. LiteGreen exploits *live migration* and *RDP* to gain short idle period savings
1. Significant energy savings exist in short idle periods
2. Holygrail = High manageability + High savings
3. LiteGreen exploits live migration and RDP to gain short idle period savings
4. LiteGreen can save 72-74% more energy on top of existing savings (32%) for collected traces
1. Significant energy savings exist in short idle periods
2. Holygrail = High manageability + High savings
3. LiteGreen exploits live migration and RDP to gain short idle period savings
4. LiteGreen can save 72-74% more energy on top of existing savings (32%) for collected traces
5. Hyper-V prototype shows total of 71% savings for 10 users
1. Significant energy savings exist in **short idle periods**
2. Holygrail = High manageability + High savings
3. LiteGreen exploits **live migration** and **RDP** to gain short idle period savings
4. LiteGreen can save **72-74%** more energy on top of existing savings (32%) for collected traces
5. Hyper-V prototype shows total of **71%** savings for 10 users

**LiteGreen = Saving Desktop Energy using Virtualization**