



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?

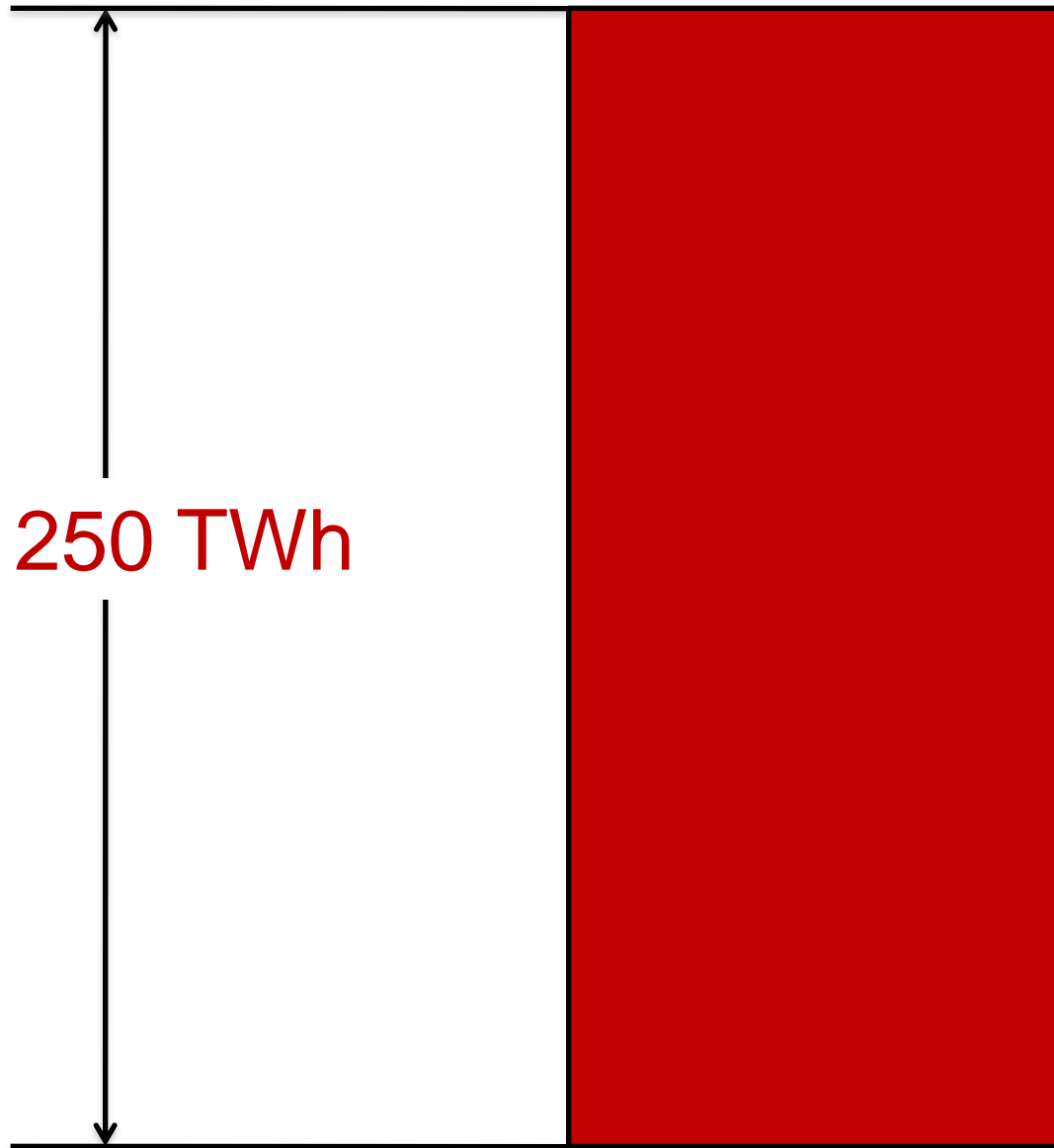


Key observation

Users do not like **disruption**
Automated energy saving is needed

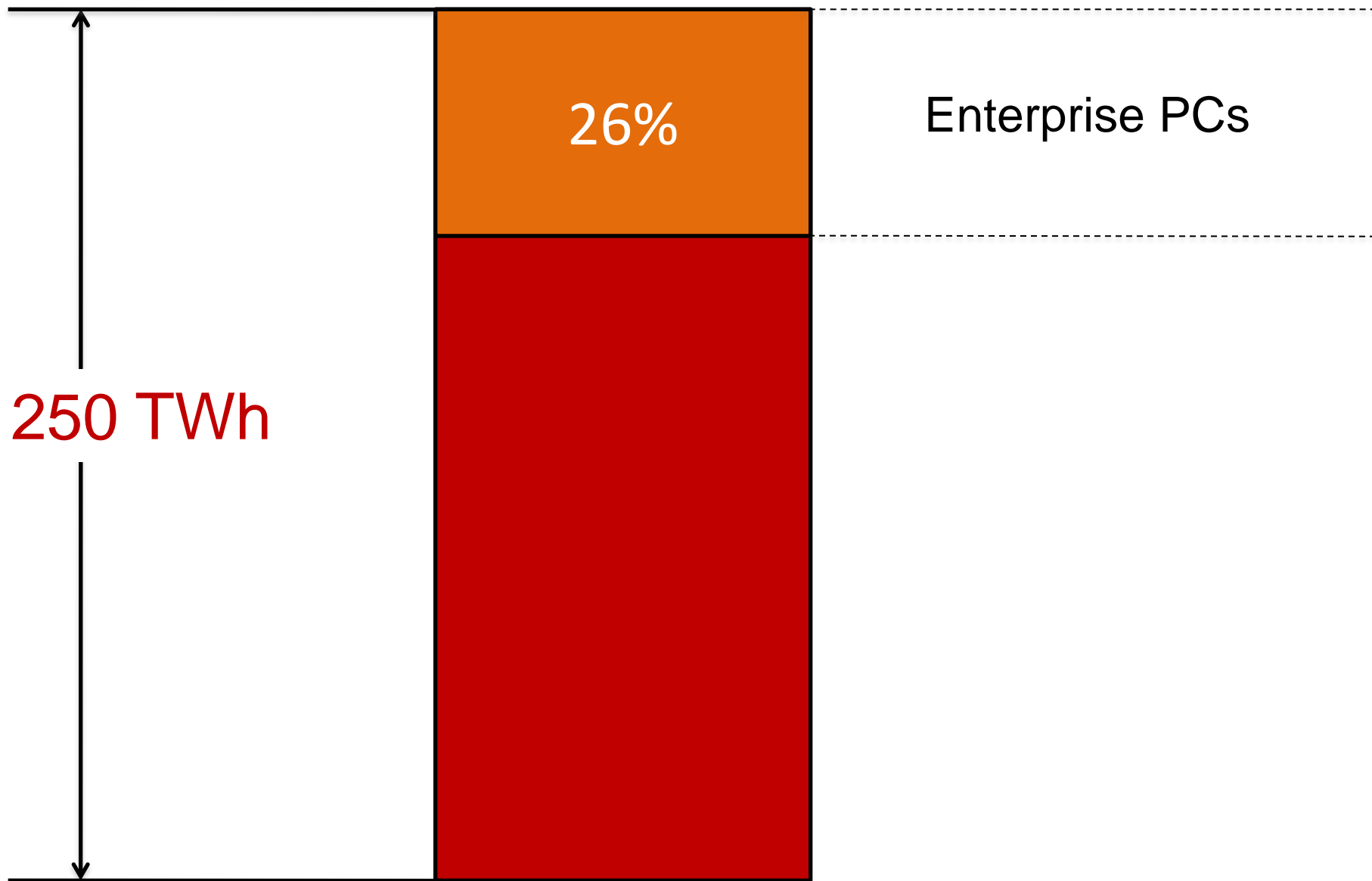
Energy Usage of Electronics in the US (2006)

Energy Usage of Electronics in the US (2006)



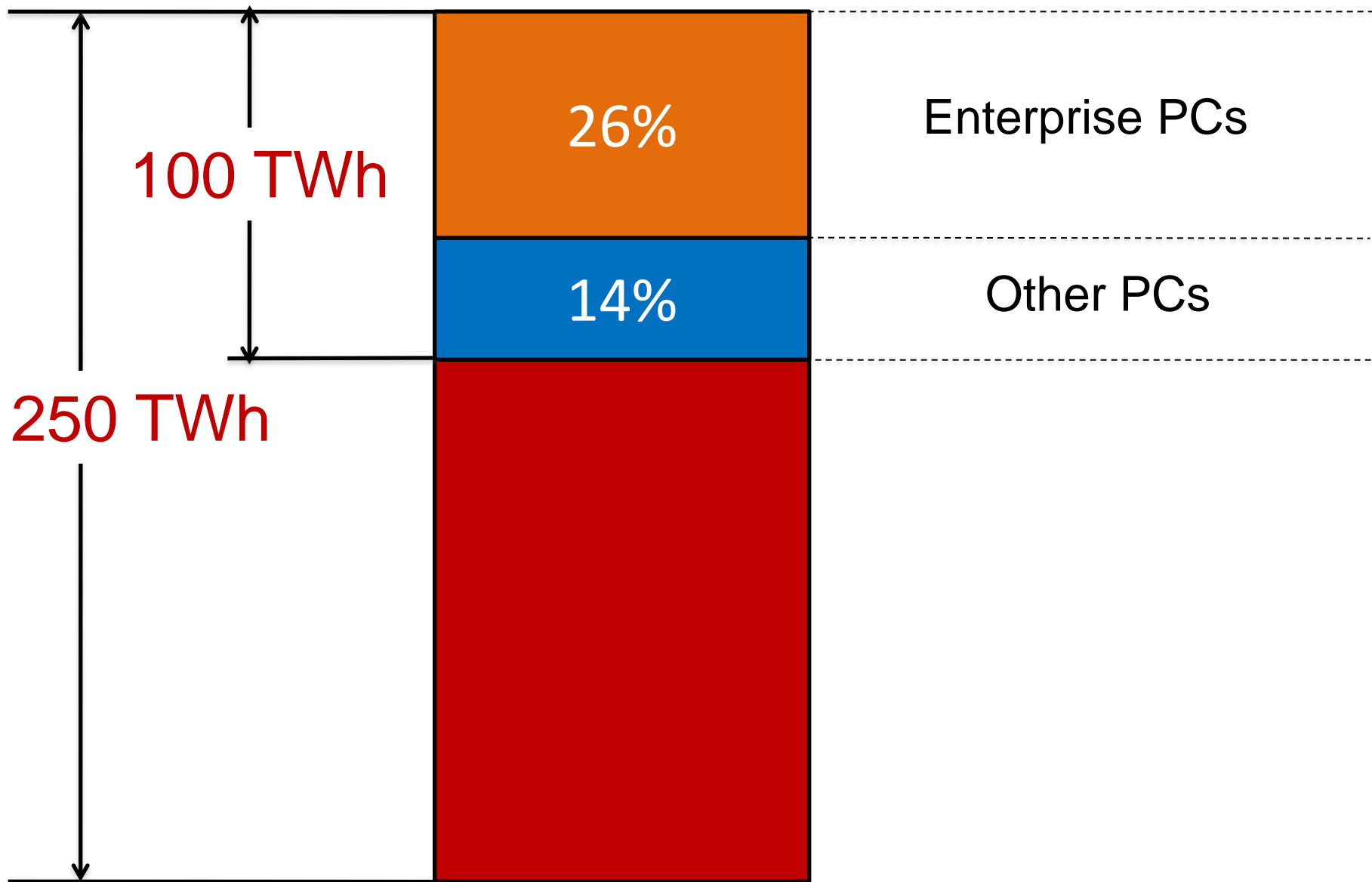
[Greener PCs for the Enterprise, Nordman'06]

Energy Usage of Electronics in the US (2006)

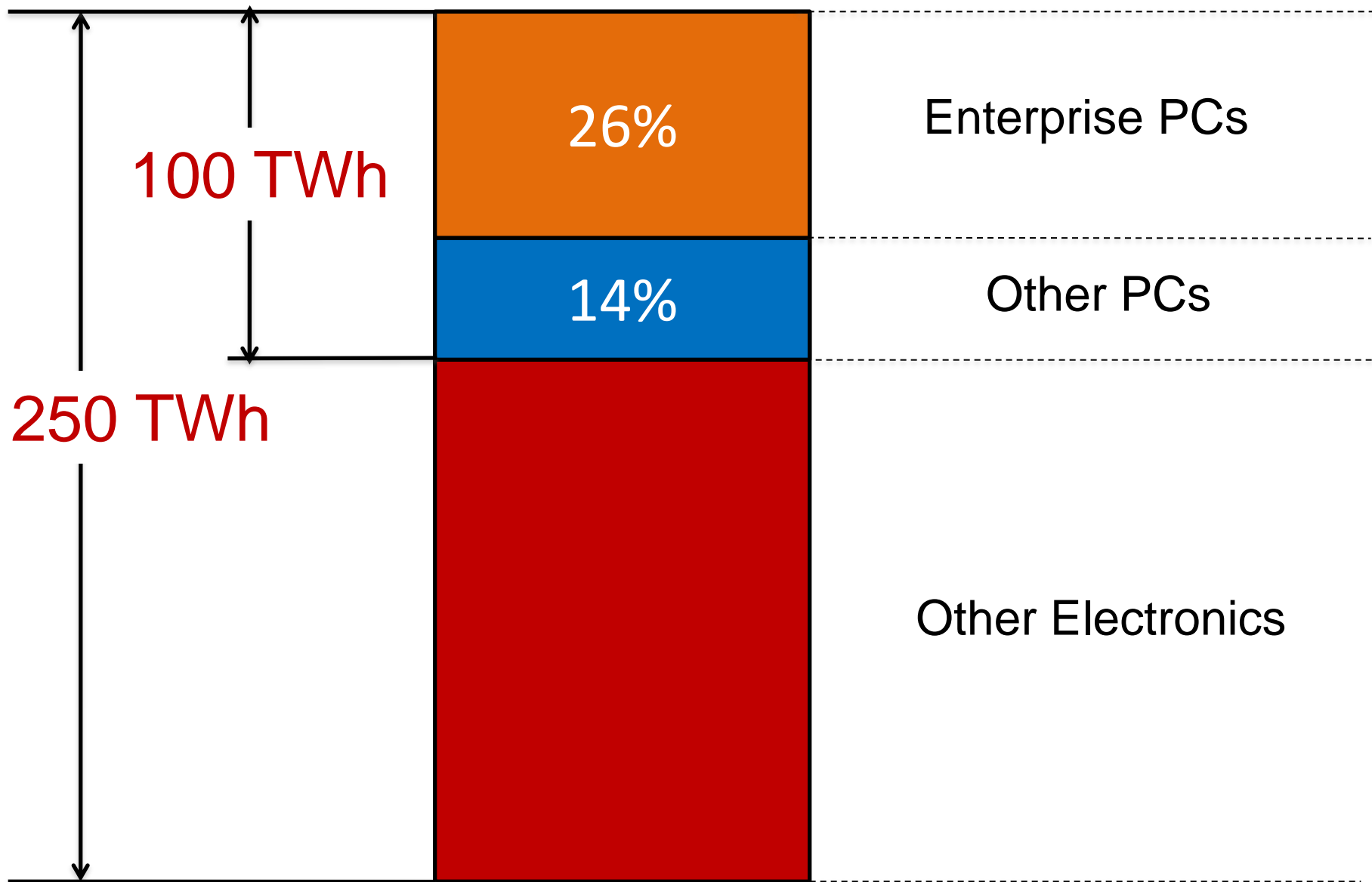


[Greener PCs for the Enterprise, Nordman'06]

Energy Usage of Electronics in the US (2006)



Energy Usage of Electronics in the US (2006)



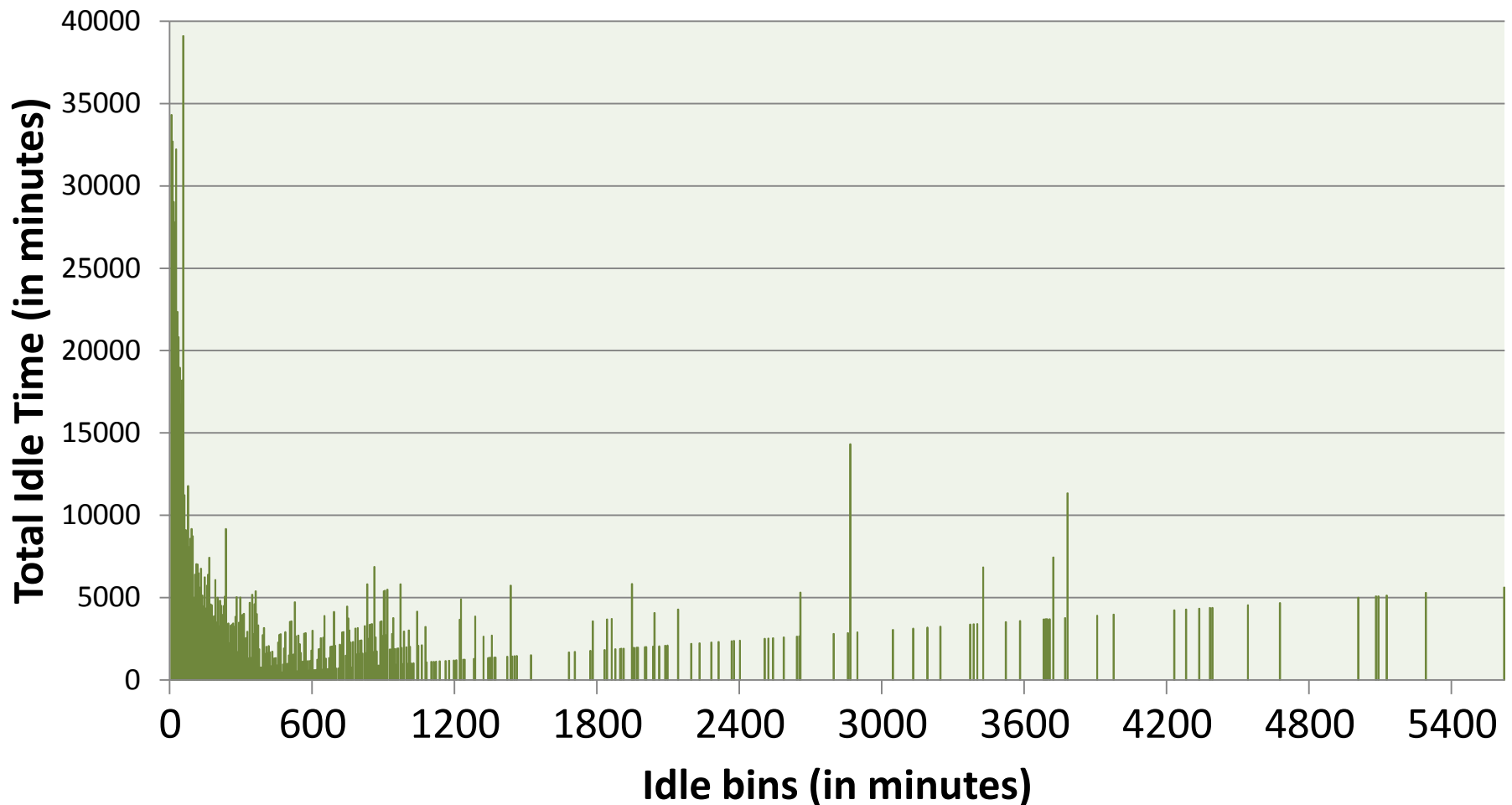
[Greener PCs for the Enterprise, Nordman'06]

How much energy can we save?

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

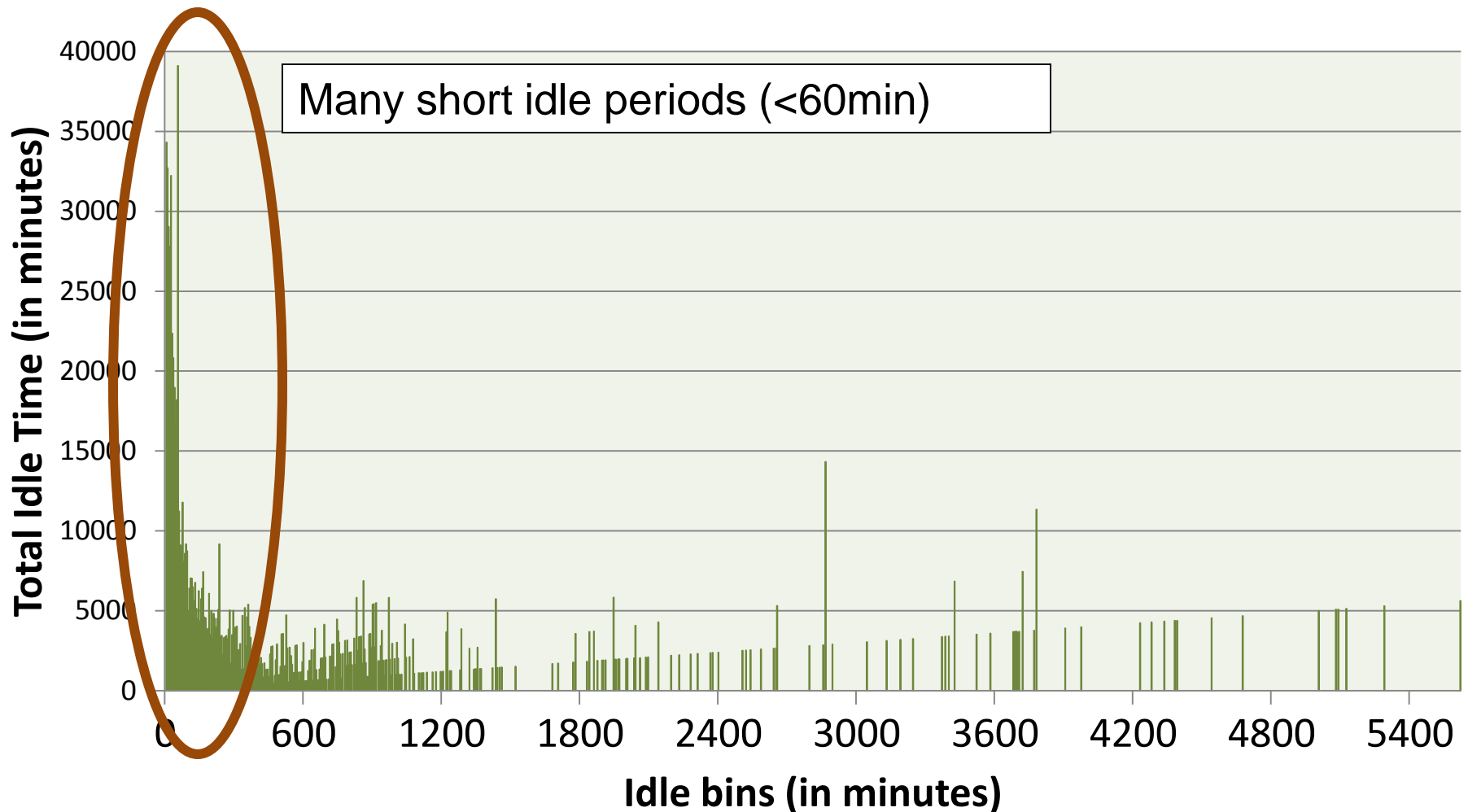
How much energy can we save?

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



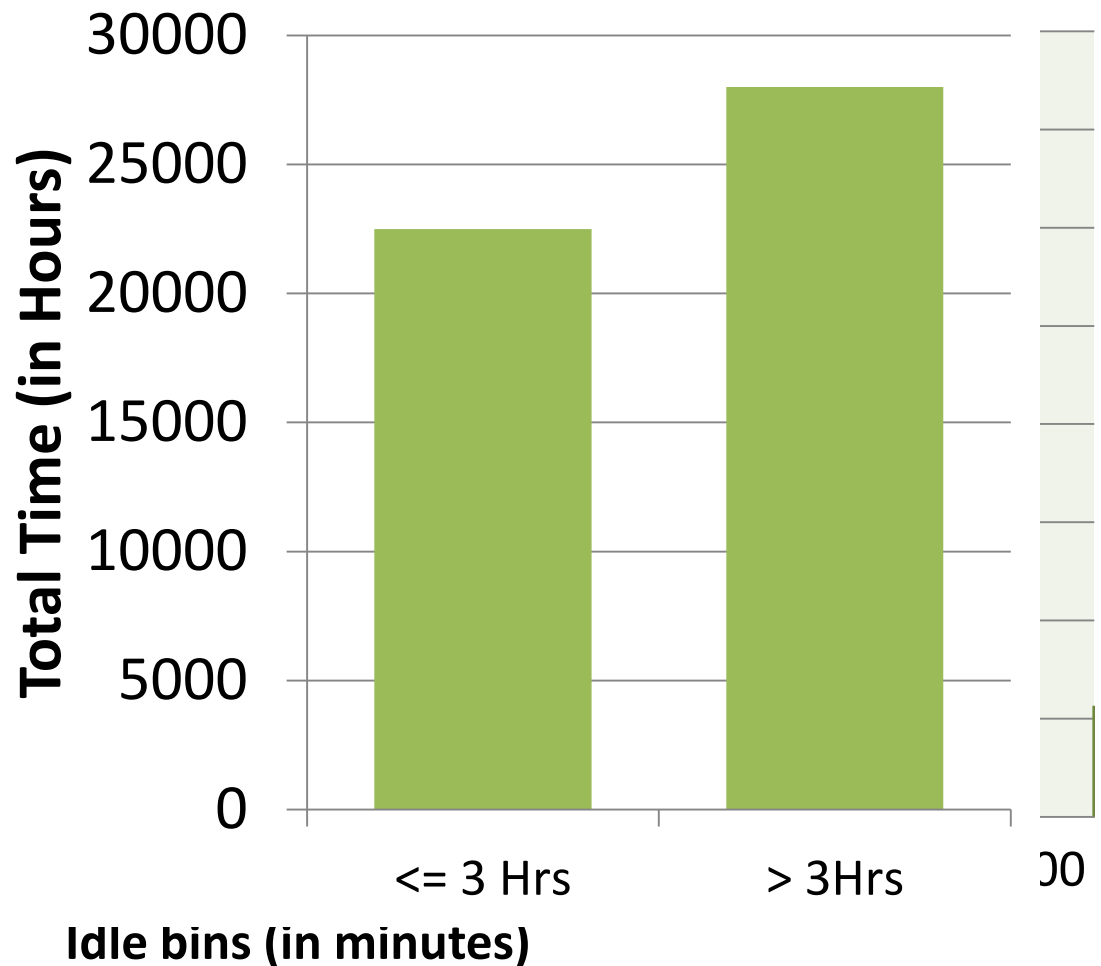
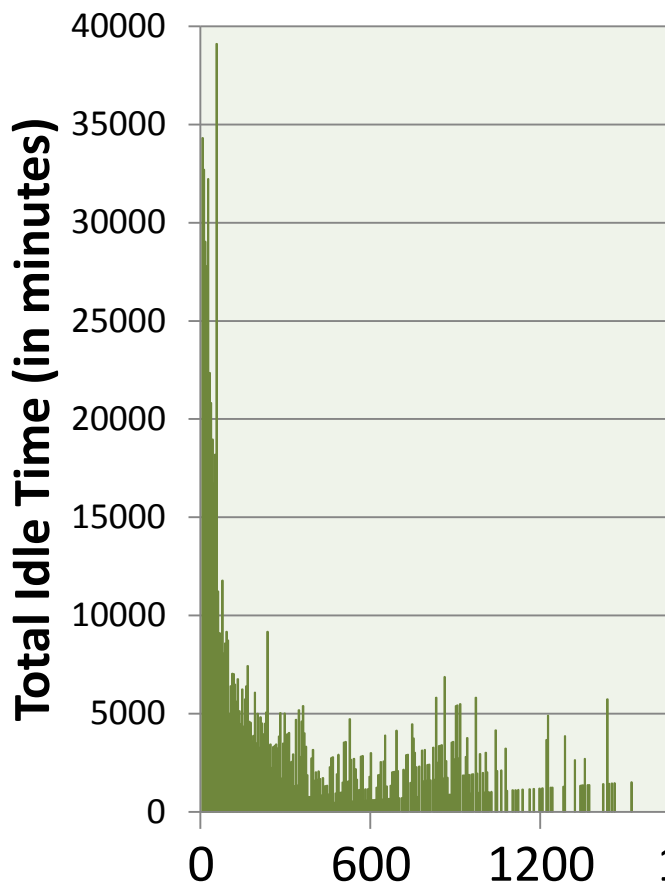
How much energy can we save?

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



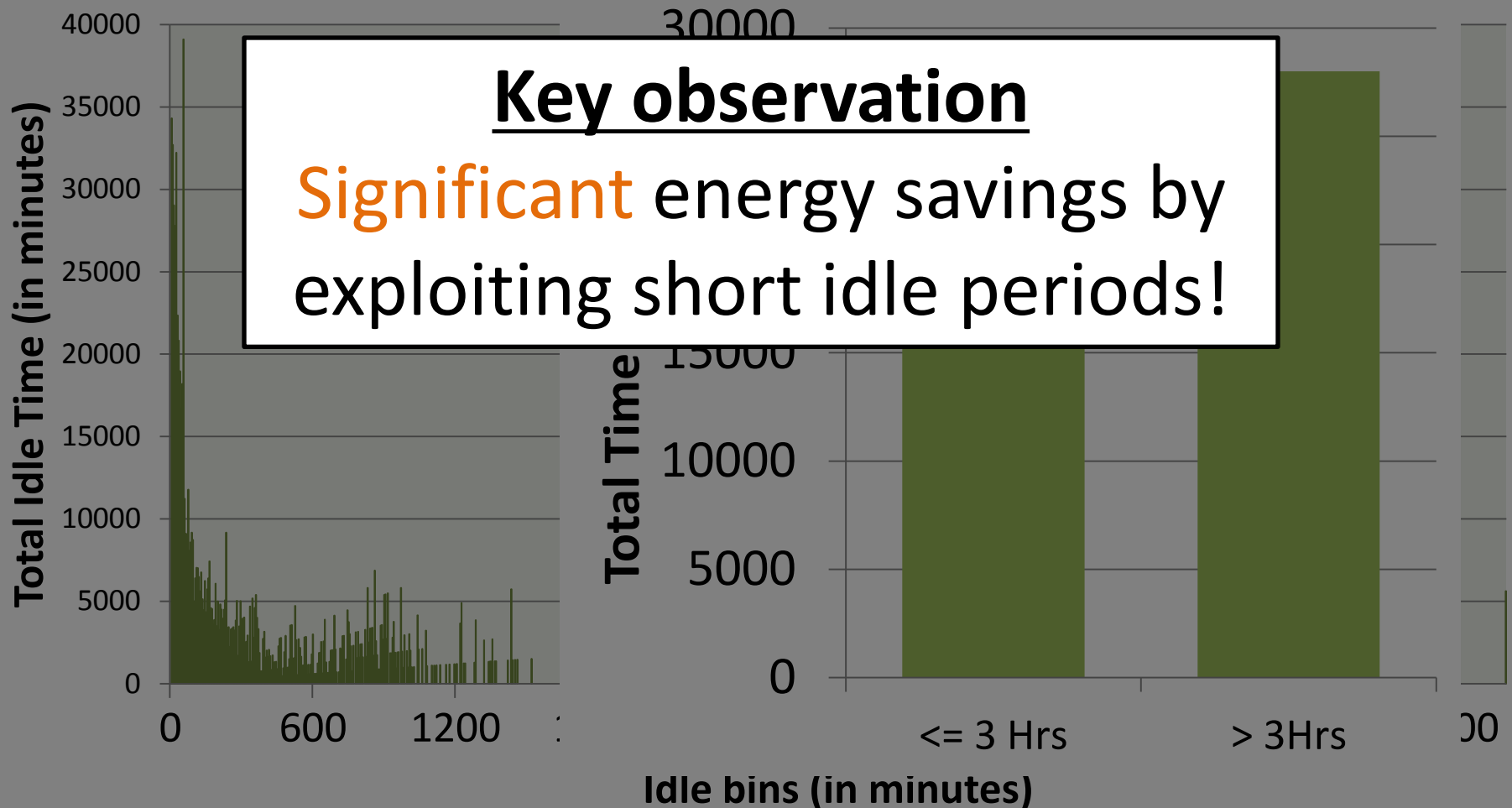
How much energy can we save?

- 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



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]**

- + S
- N
- Prot
- + A

Key observation

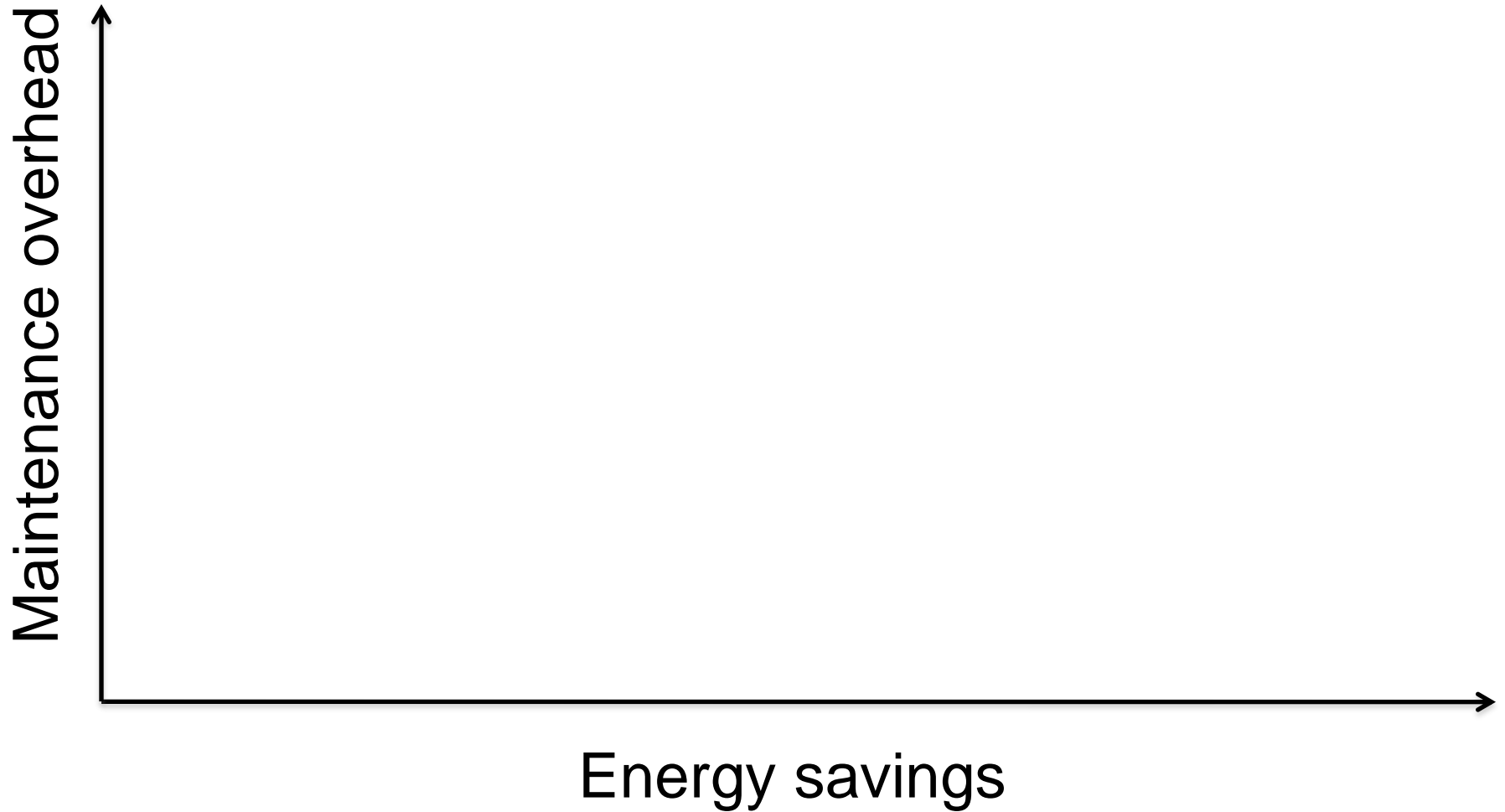
Disruption to applications and **lack of generality** are major limitations

- Disruption to existing applications, deployment problems

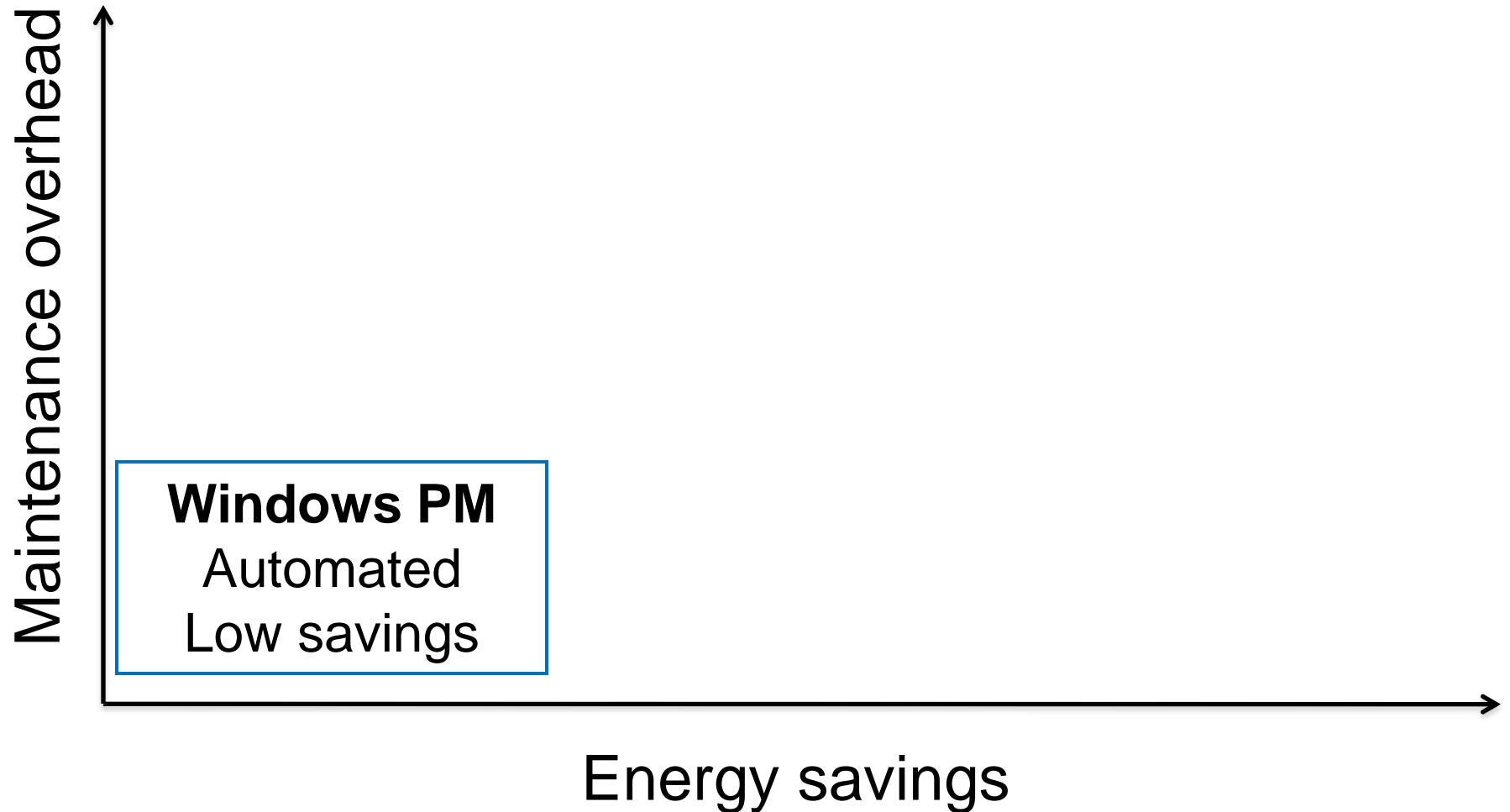
- **Application proxy [Agarwal et.al. NSDI'09]**

- + Reduces disruption to applications that have proxies
- Increased complexity in writing application specific proxies

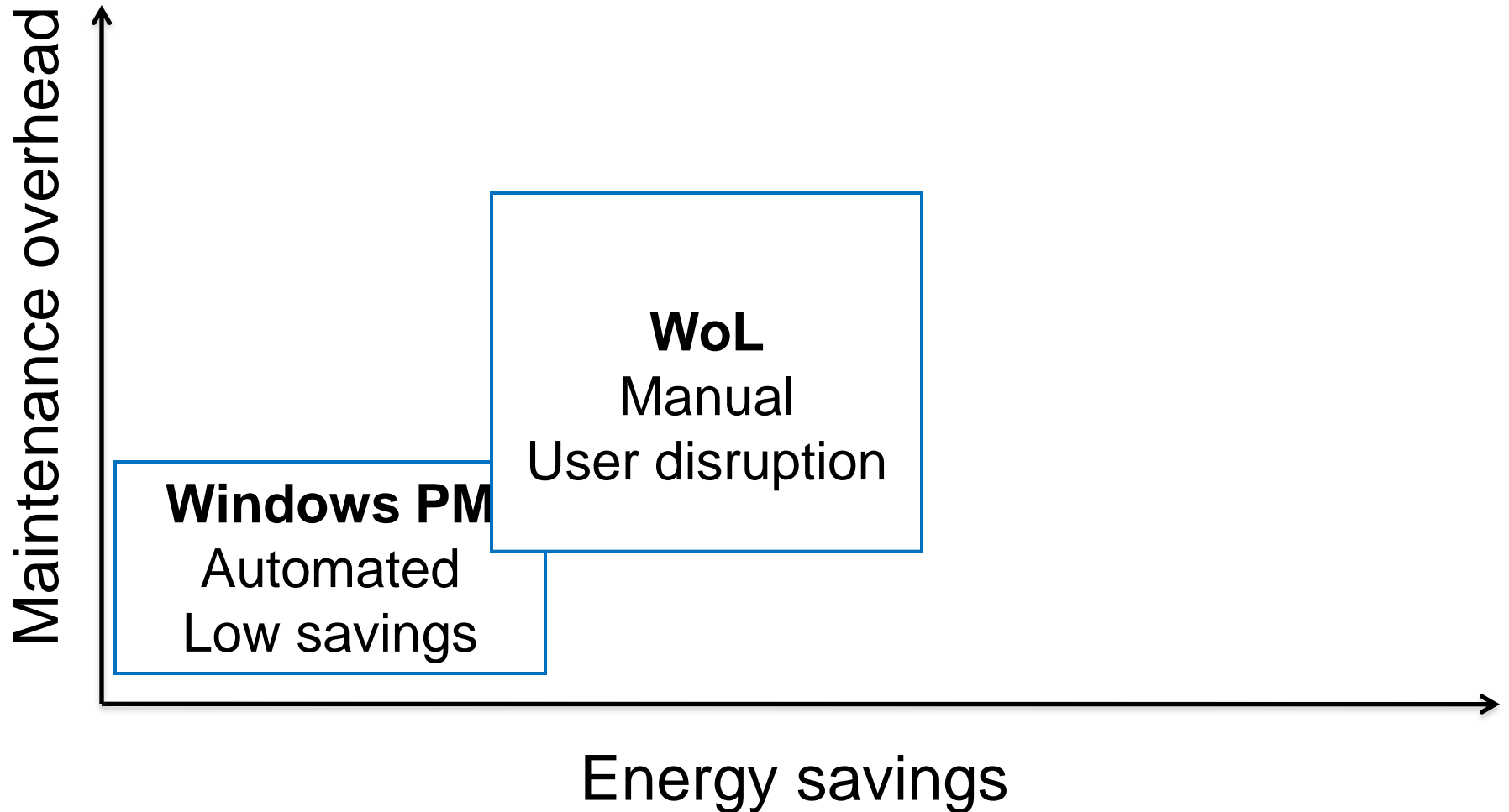
Power management spectrum



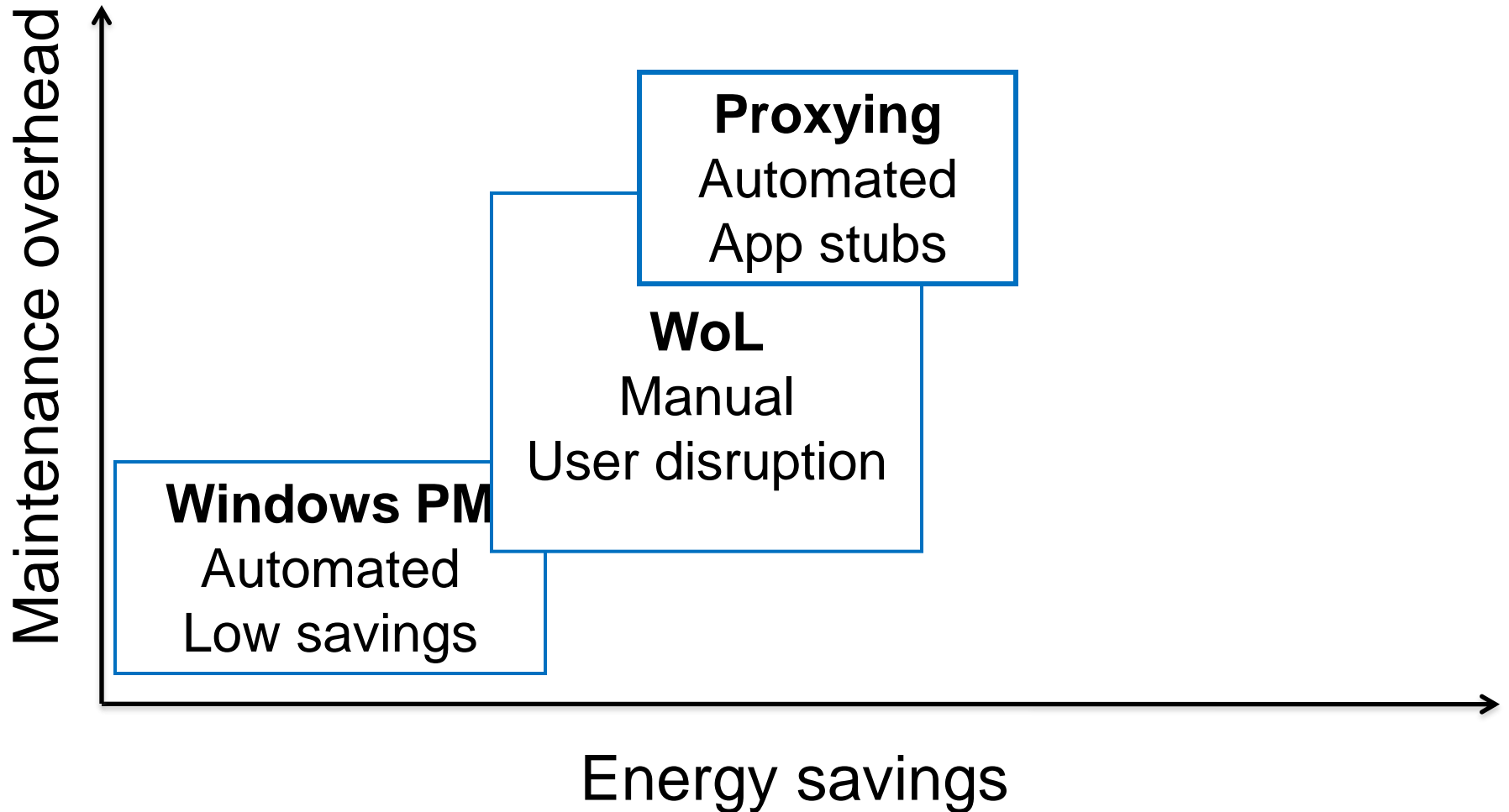
Power management spectrum



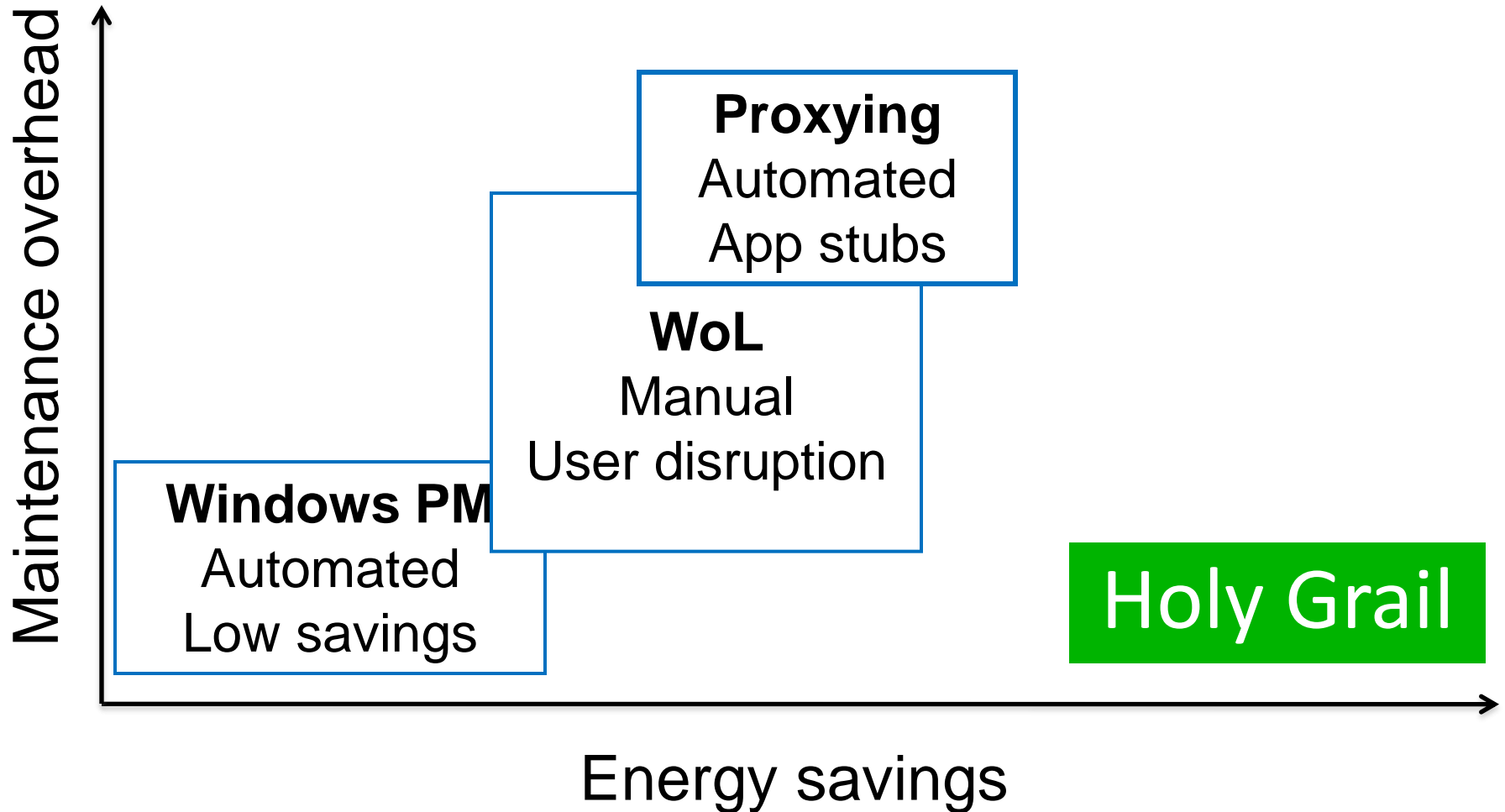
Power management spectrum



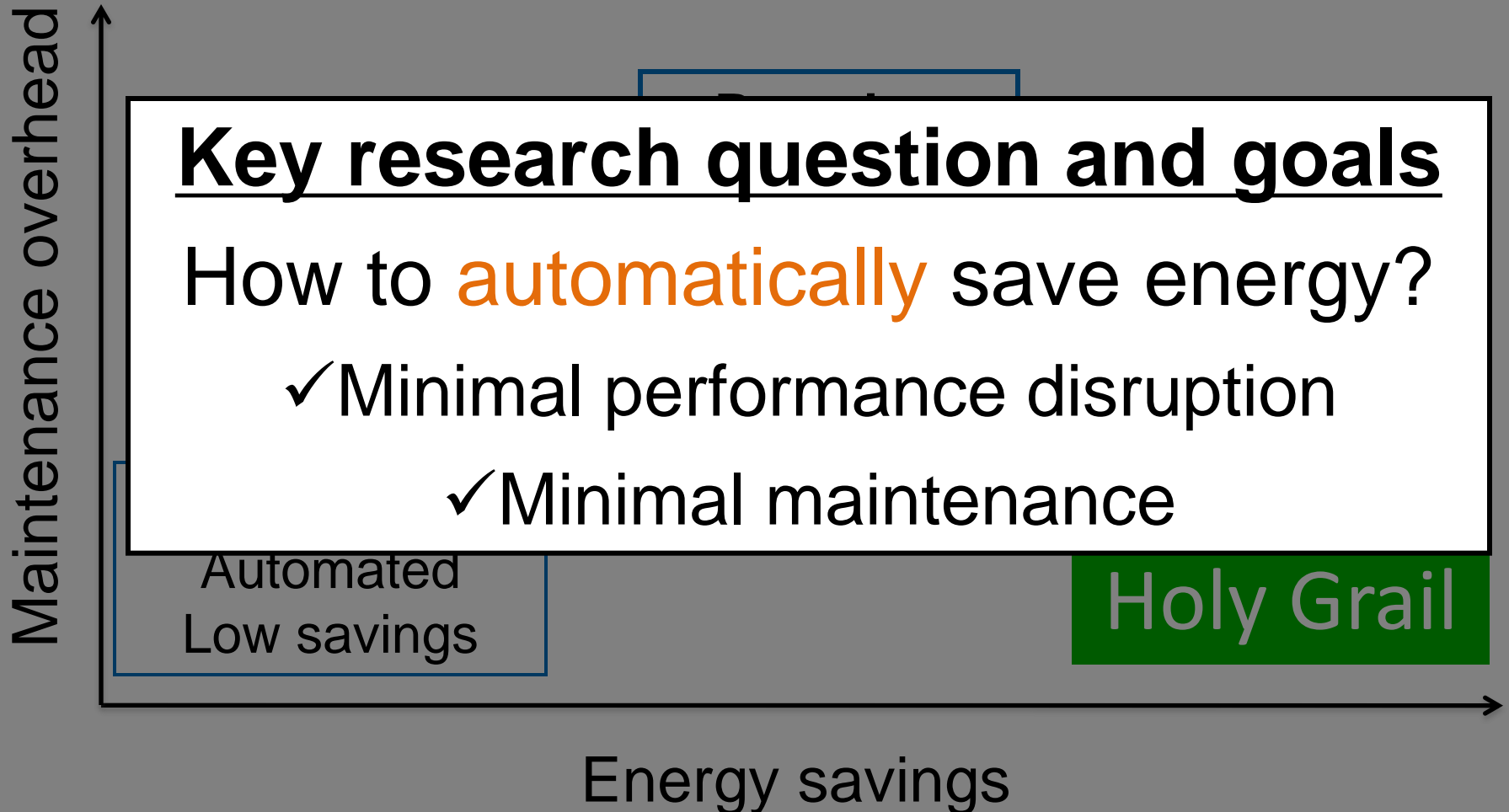
Power management spectrum



Power management spectrum



Power management spectrum



LiteGreen: virtualization and consolidation

LiteGreen: virtualization and consolidation



Hardware

LiteGreen: virtualization and consolidation



Virtual Machine

Virtualization

Hardware

LiteGreen: virtualization and consolidation



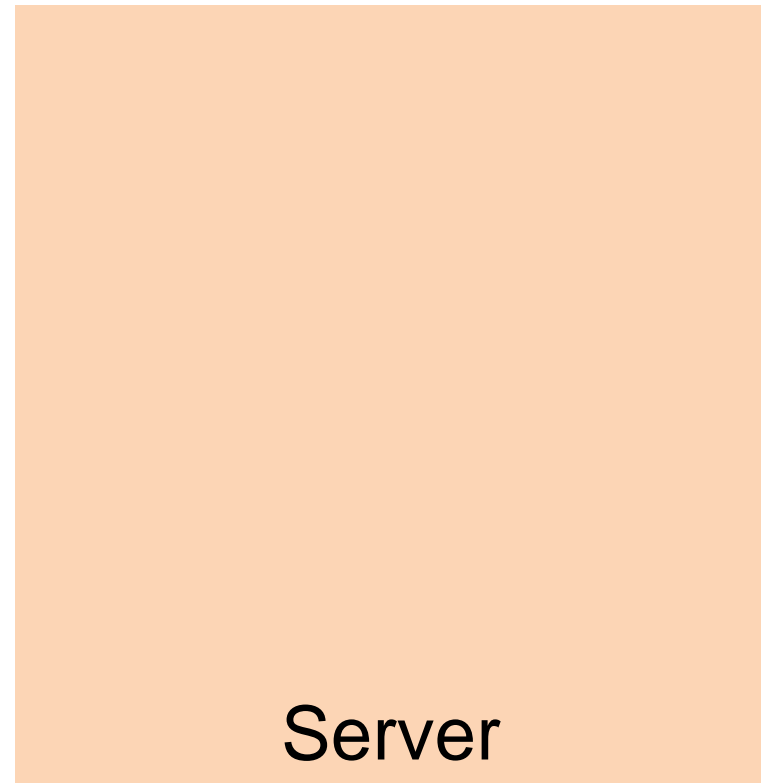
Virtual Machine

Virtualization

Hardware

Server

LiteGreen: virtualization and consolidation

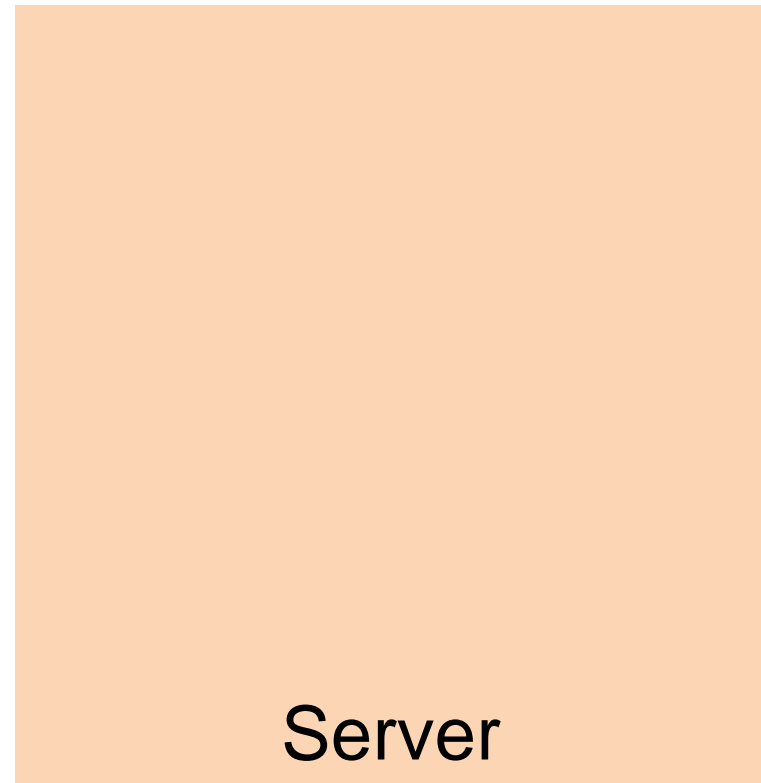


 Active VM

 Machine off

 Idle VM

LiteGreen: virtualization and consolidation

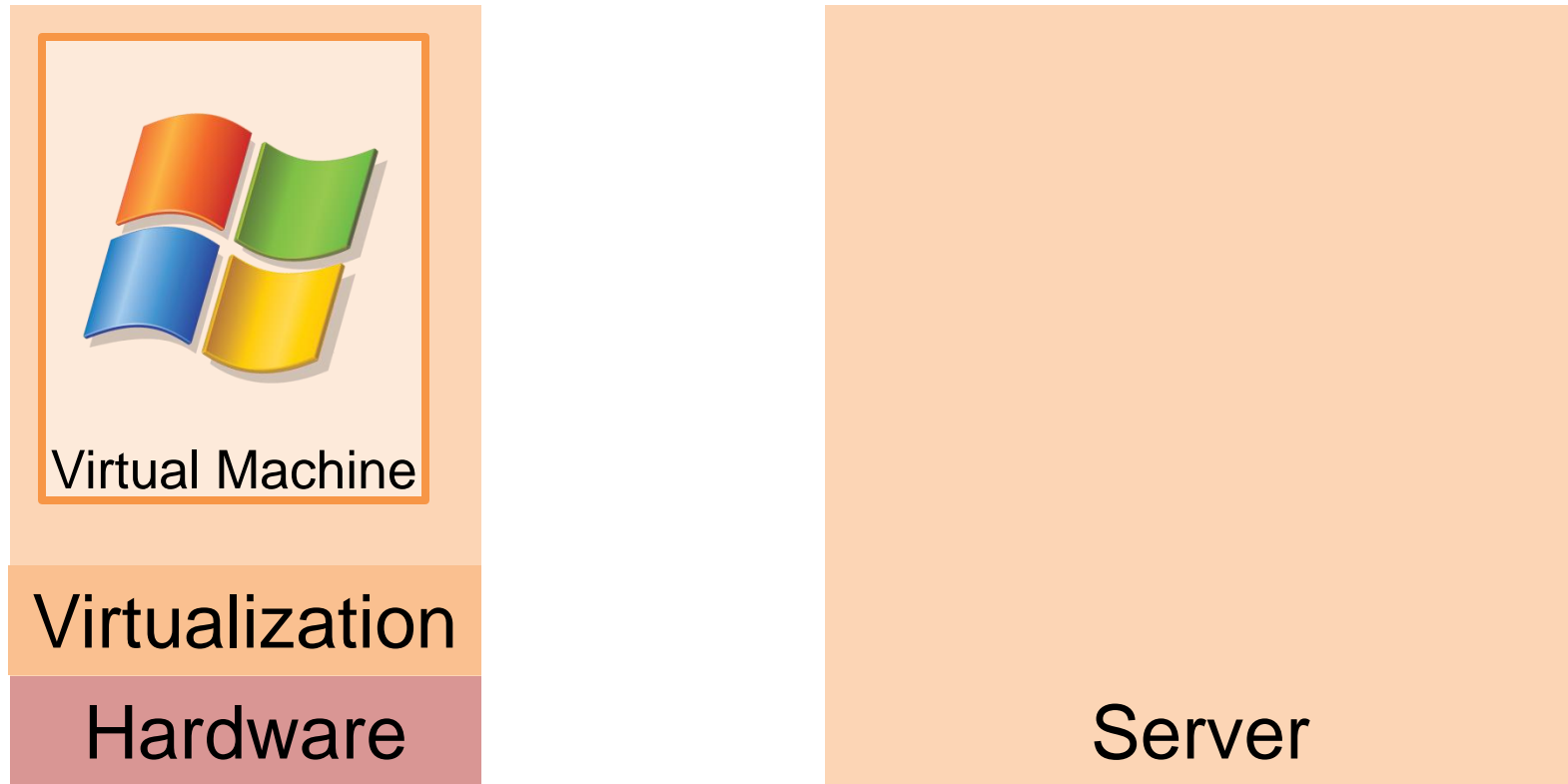


 Active VM

 Machine off

 Idle VM

LiteGreen: virtualization and consolidation



 Active VM

 Machine off

 Idle VM

LiteGreen Contributions

LiteGreen Contributions

1. Provide normal, uninterrupted 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, uninterrupted 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, uninterrupted 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, uninterrupted 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

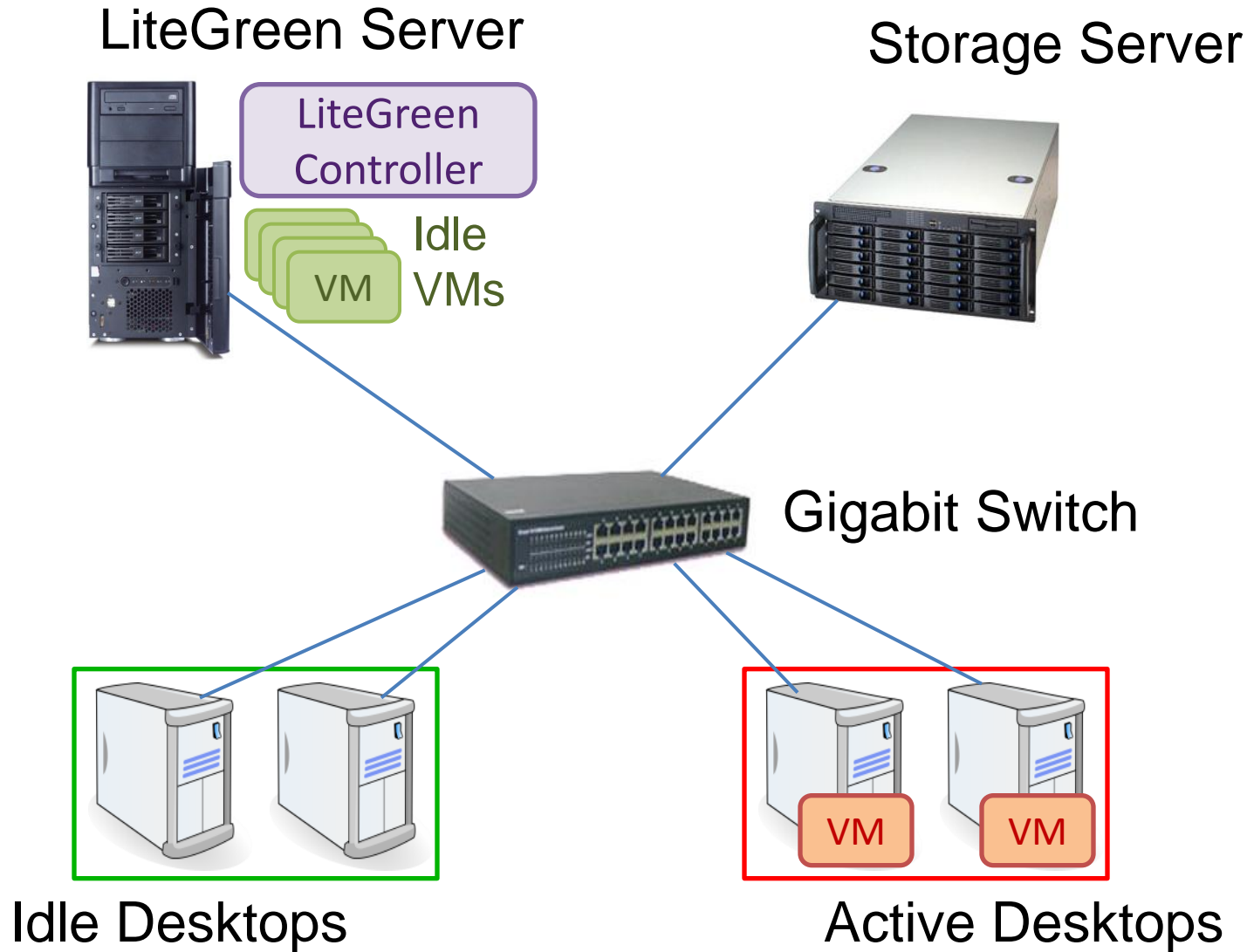
LiteGreen Contributions

1. Provide normal, uninterrupted 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

System Architecture



LiteGreen Operation

LiteGreen Operation

Problem: How to mask the effect of migration?

LiteGreen Operation

Problem: How to mask the effect of migration?

Solution: RDP client + Live migration

LiteGreen Operation

Solution: RDP client + Live migration





Virtual Machine

Virtualization

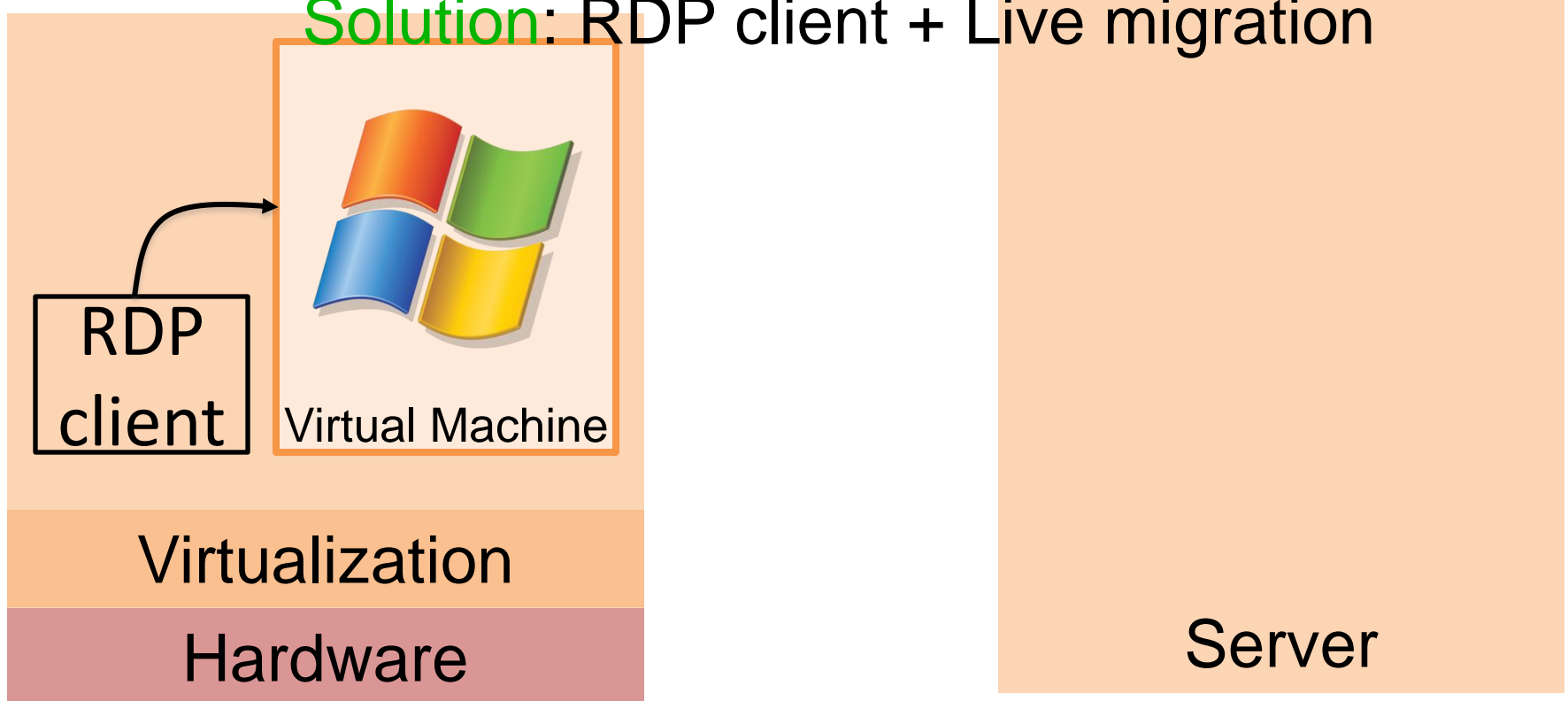
Hardware



Server

-  Active VM
-  Idle VM

LiteGreen Operation

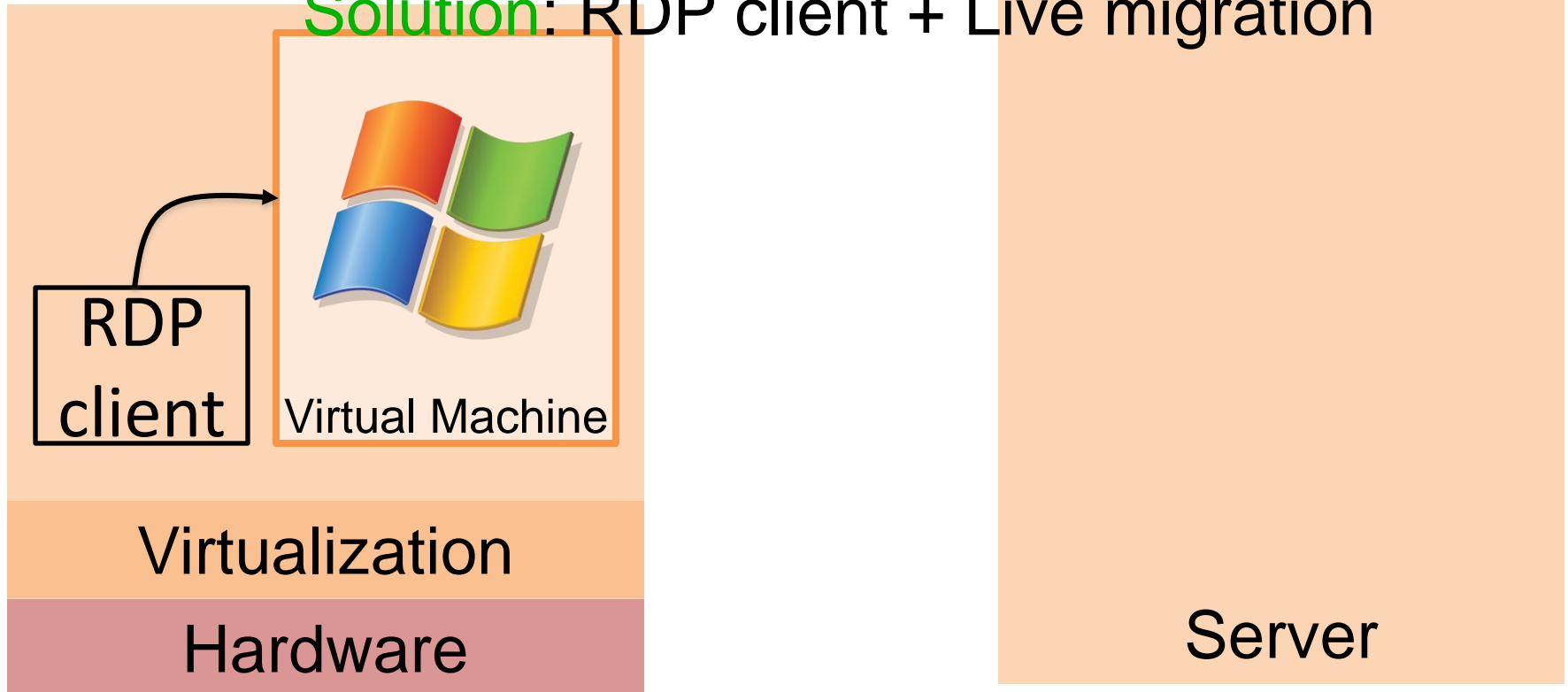
Solution: RDP client + Live migration





-  Active VM
-  Idle VM

LiteGreen Operation

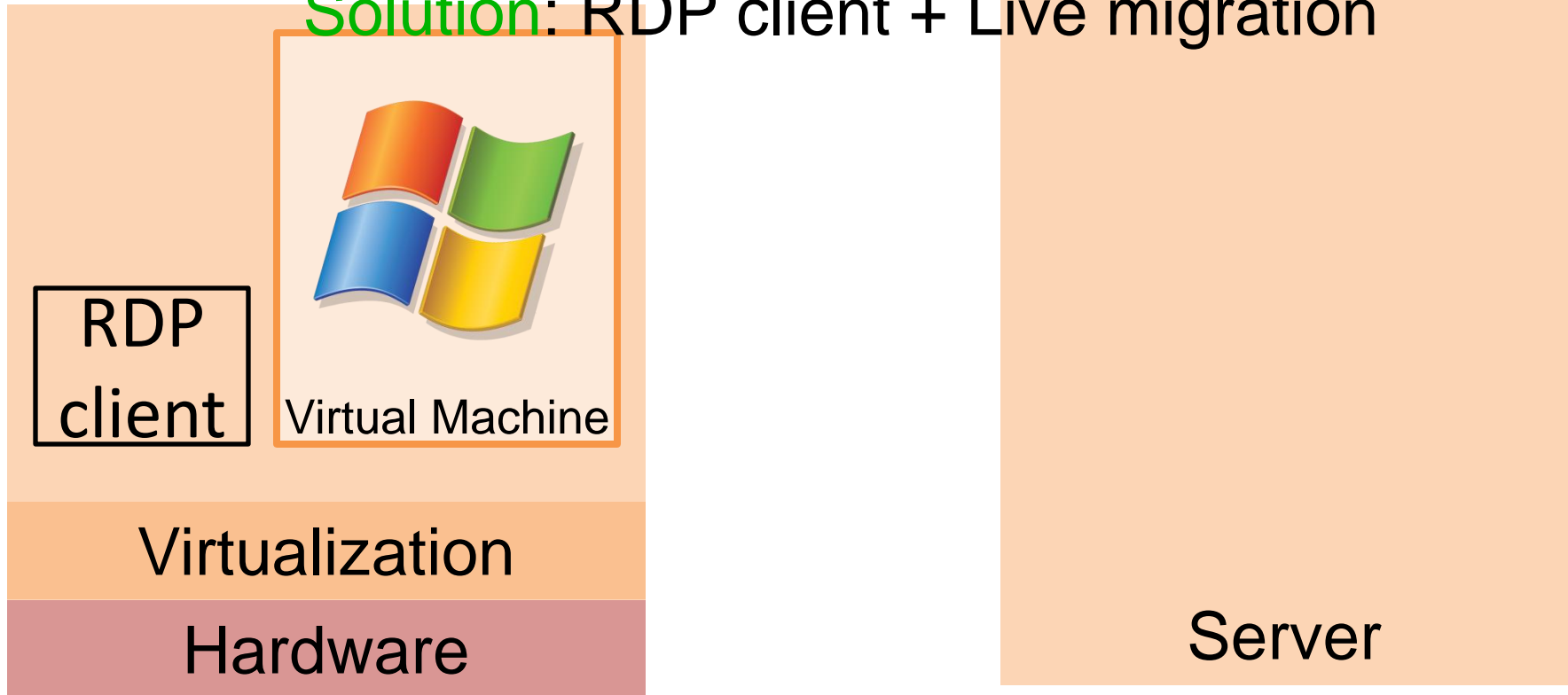
Solution: RDP client + Live migration





-  Active VM
-  Idle VM

LiteGreen Operation

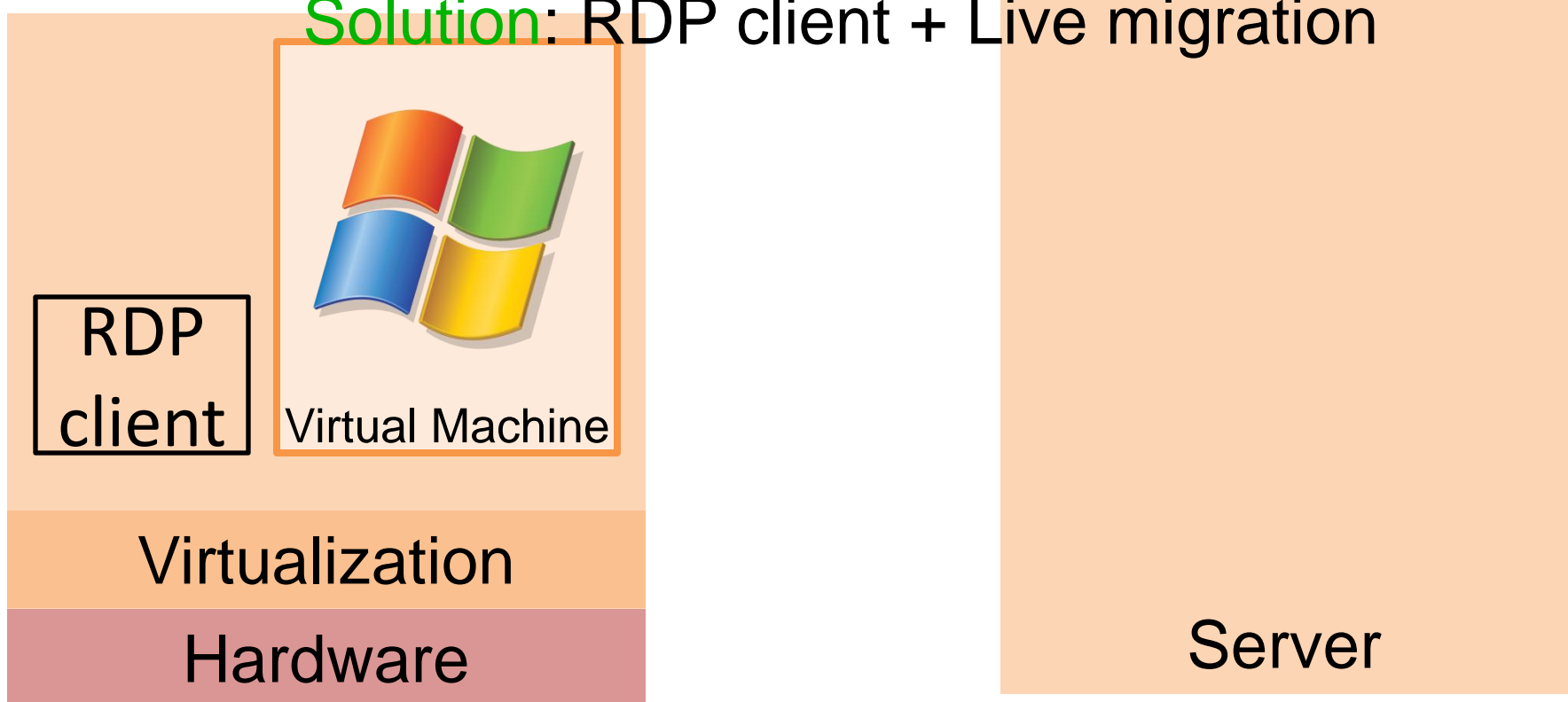
Solution: RDP client + Live migration





-  Active VM
-  Idle VM

LiteGreen Operation

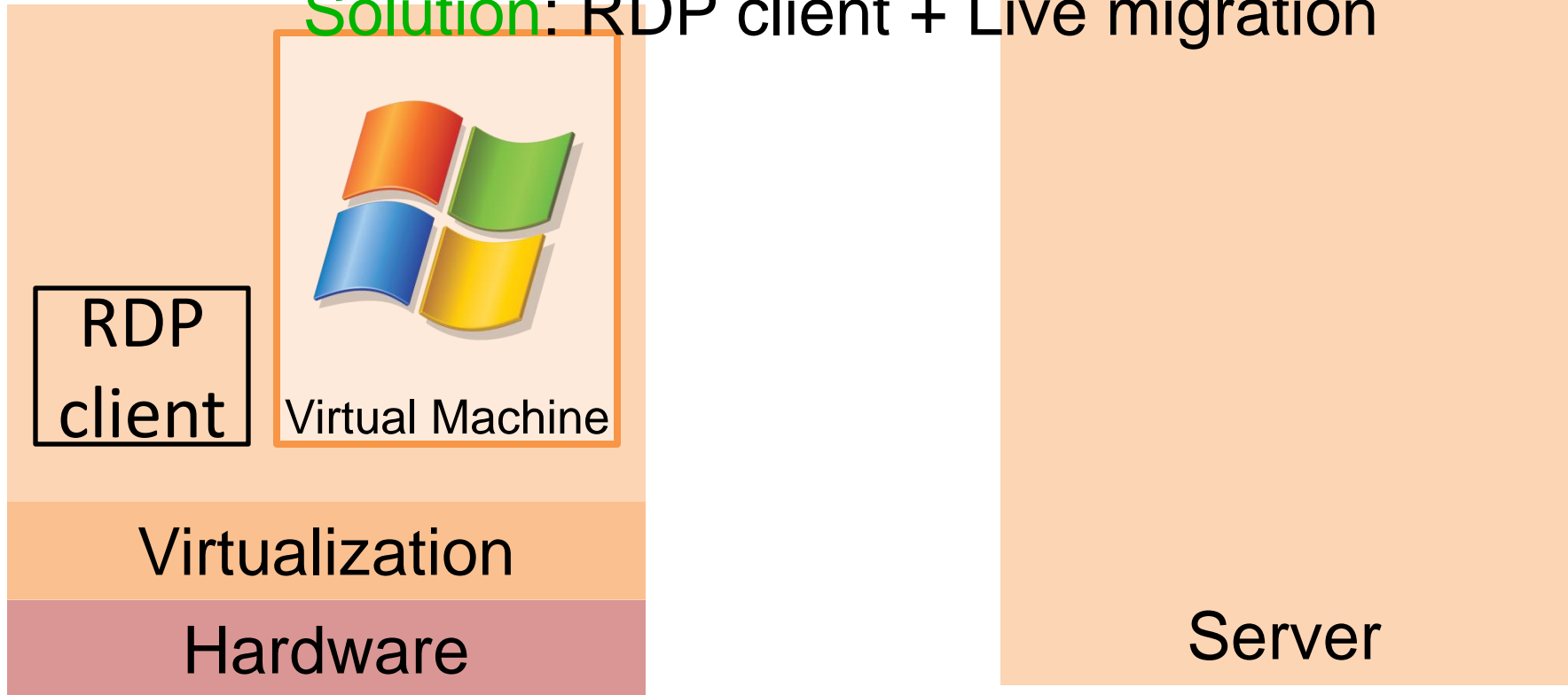
Solution: RDP client + Live migration





-  Active VM
-  Idle VM

LiteGreen Operation

Solution: RDP client + Live migration



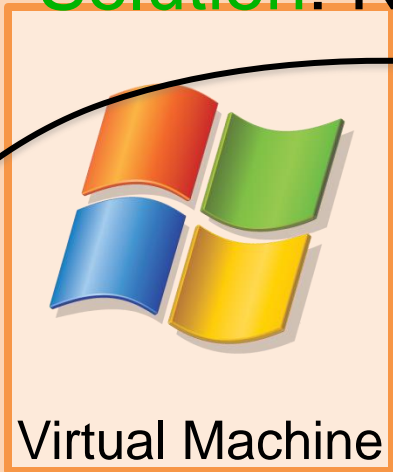
-  Active VM
-  Idle VM

LiteGreen Operation

Solution: RDP client + Live migration

Few
seconds



RDP
client



Virtualization

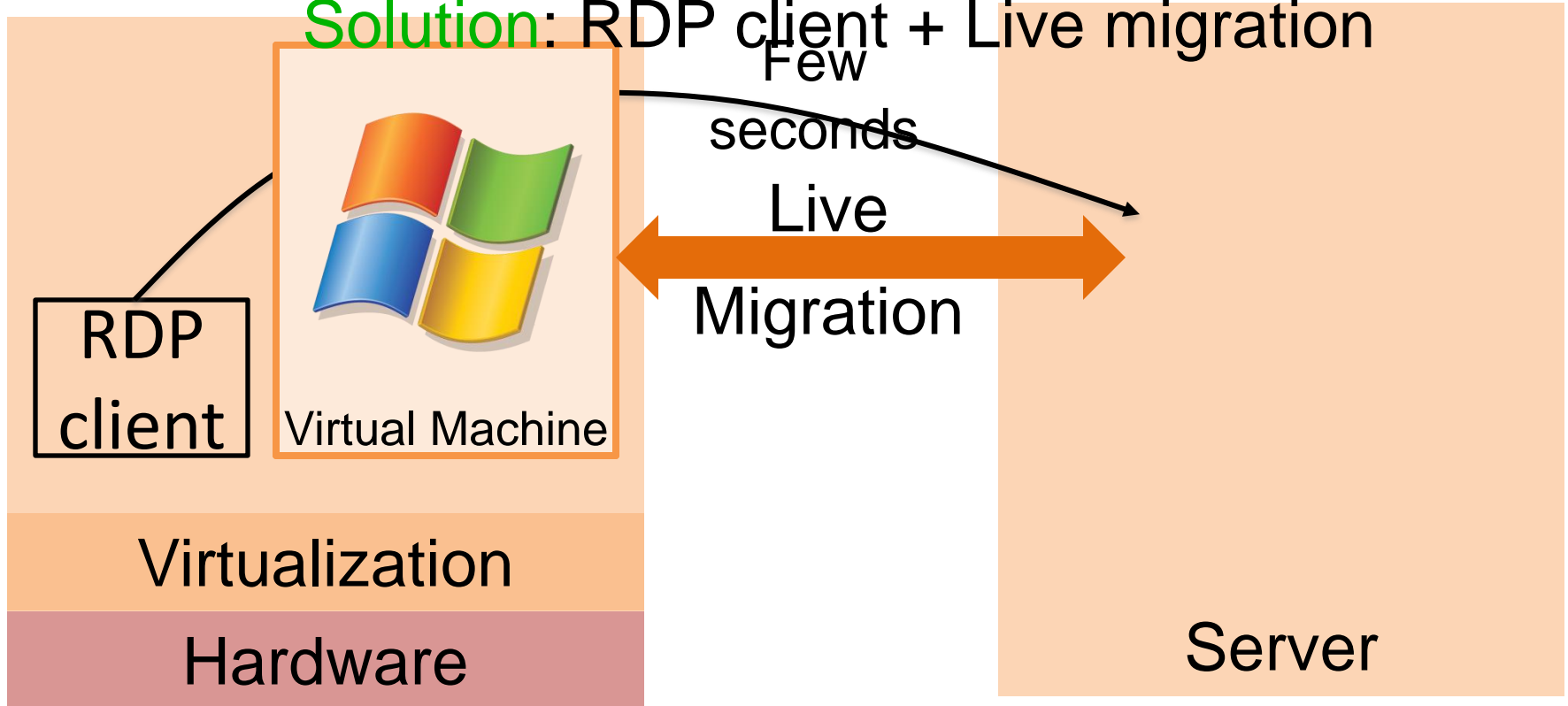
Hardware



Server

-  Active VM
-  Idle VM

LiteGreen Operation

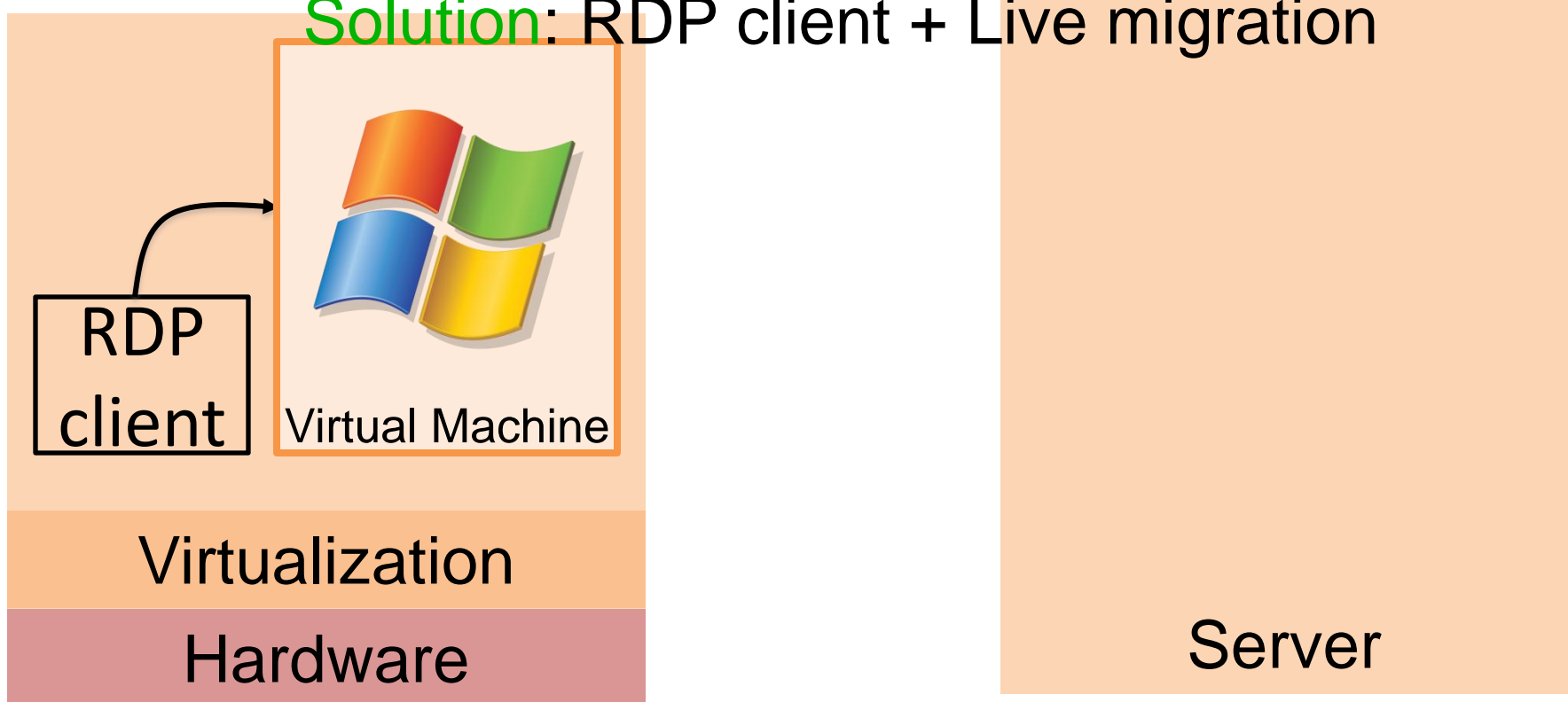
Solution: RDP client + Live migration





-  Active VM
-  Idle VM

LiteGreen Operation

Solution: RDP client + Live migration



-  Active VM
-  Idle VM

LiteGreen in action (demo)

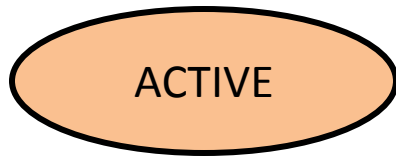
Savings algorithm – Intuition

Savings algorithm – Intuition

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

Savings algorithm – Intuition

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



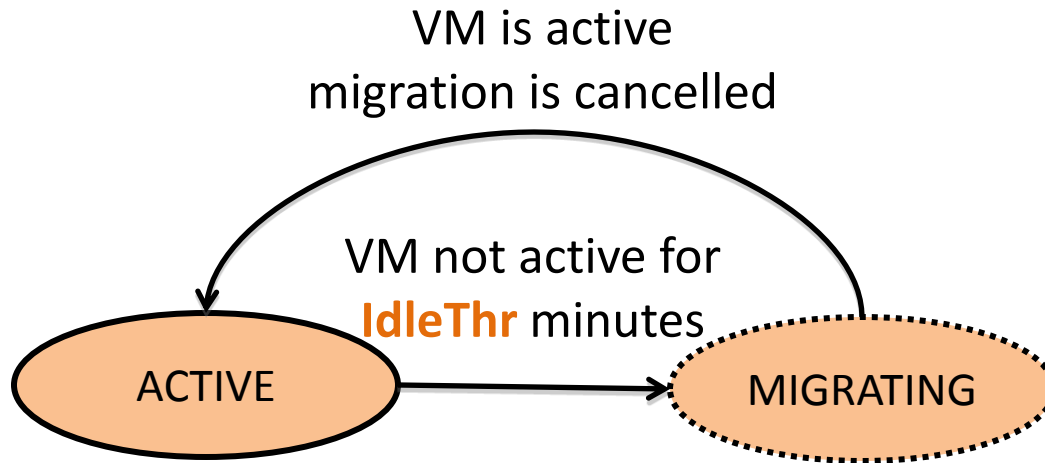
Savings algorithm – Intuition

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



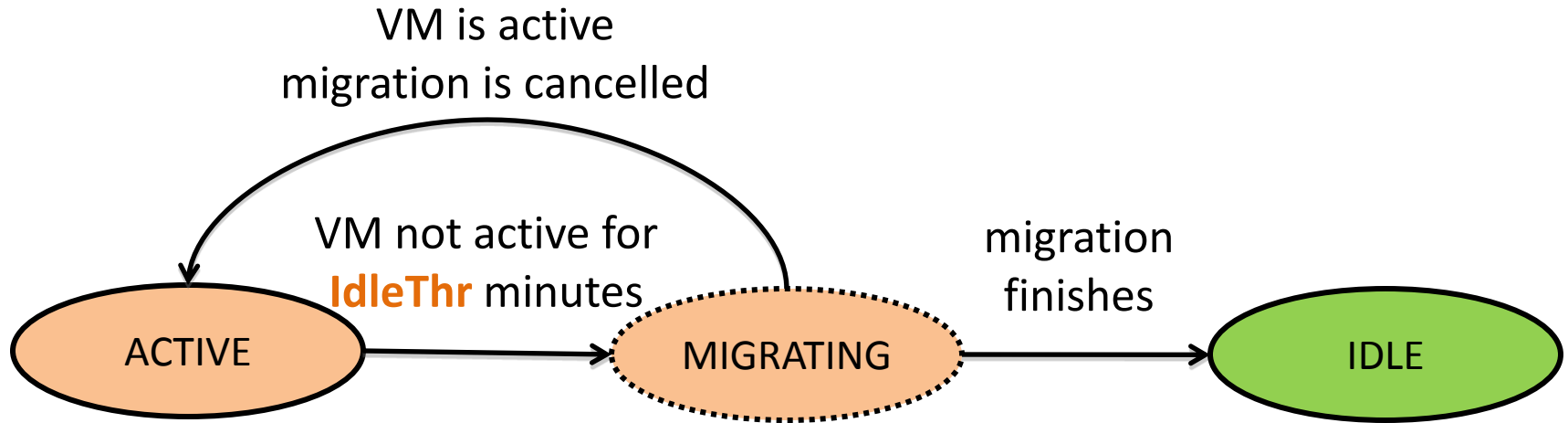
Savings algorithm – Intuition

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



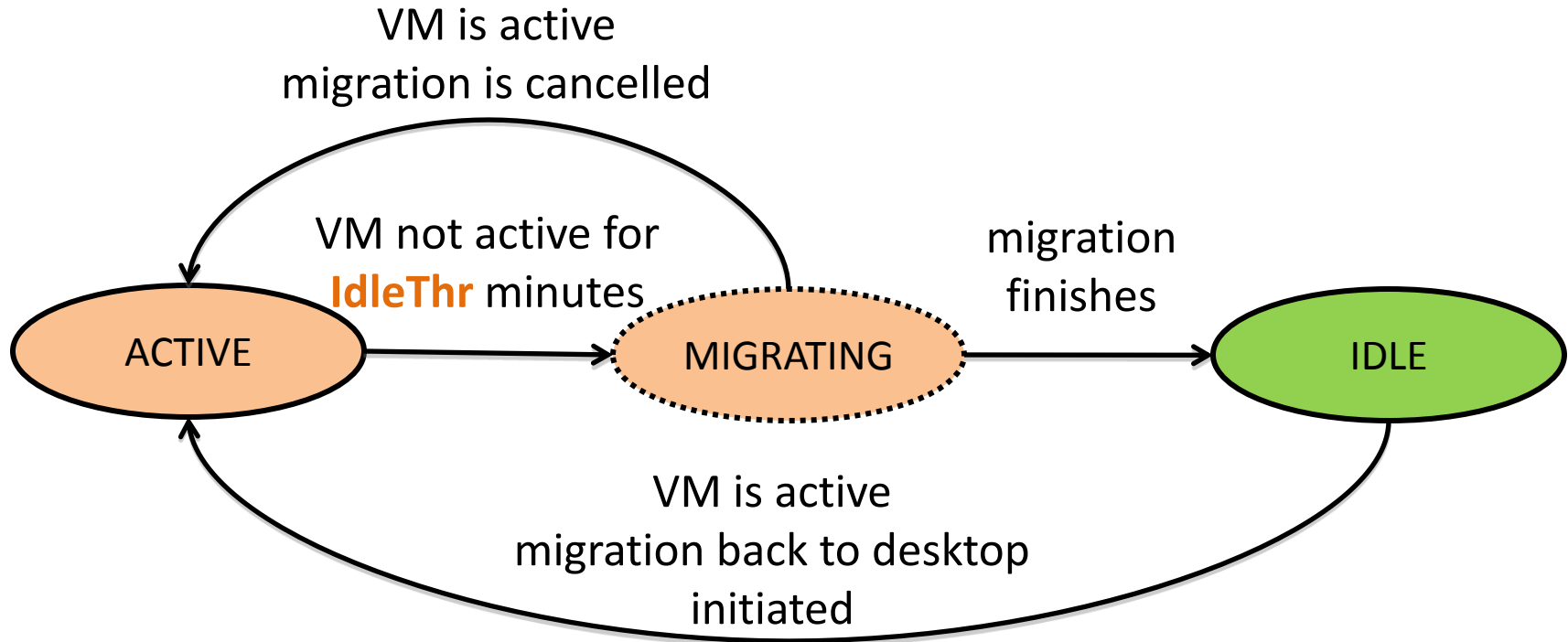
Savings algorithm – Intuition

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



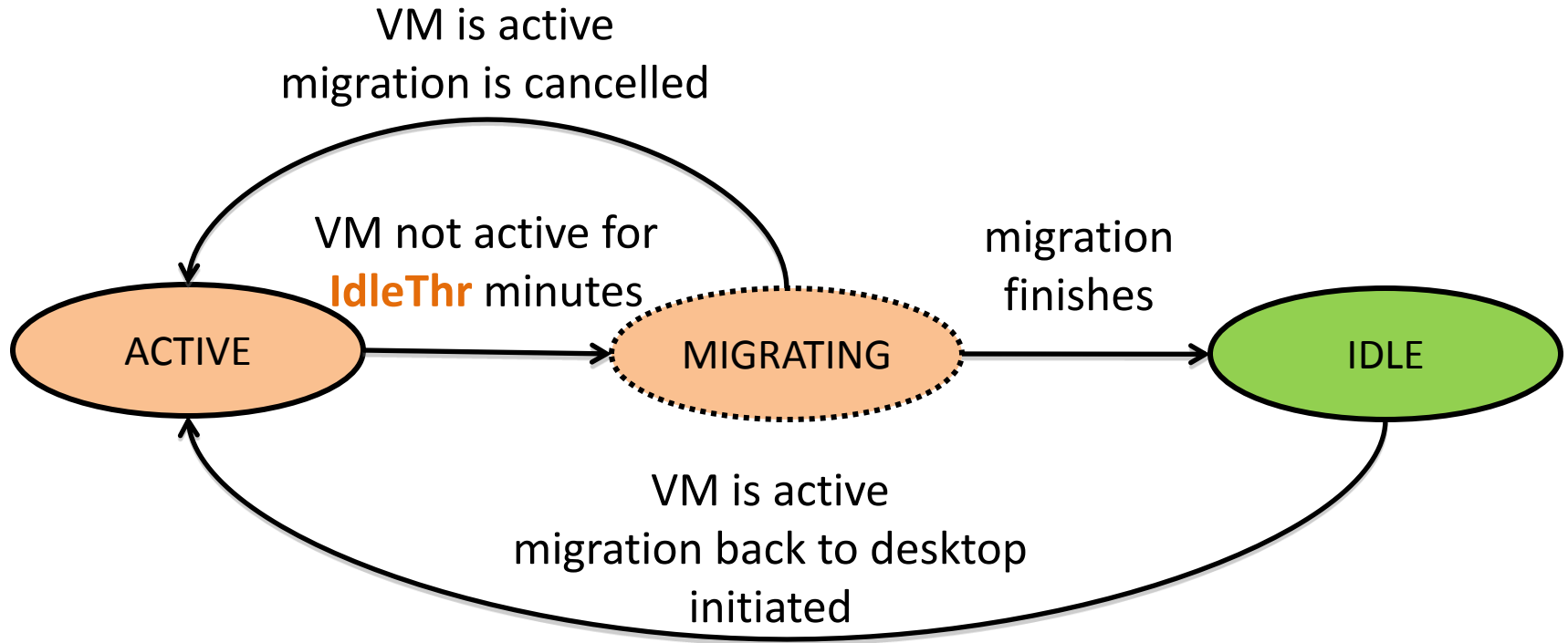
Savings algorithm – Intuition

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



Savings algorithm – Intuition

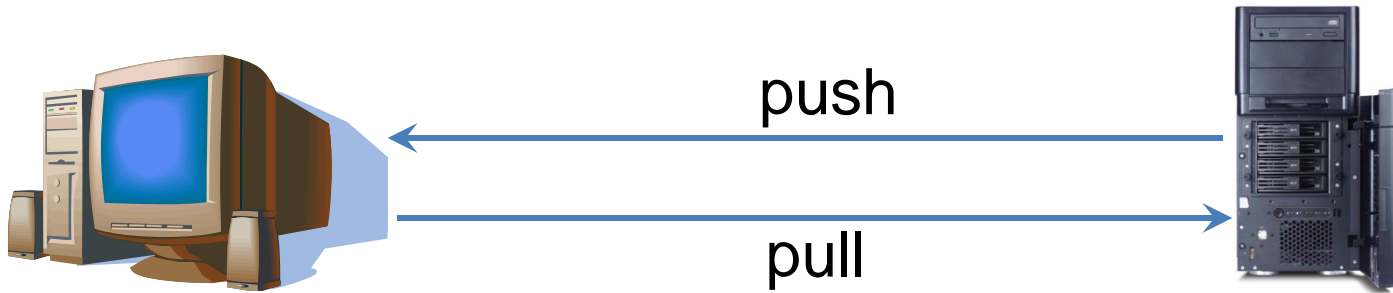
Active VM = (User is interactive) OR (Resource usage > ResThr)



How to choose **ResThr** and **IdleThr**? Statistical analysis

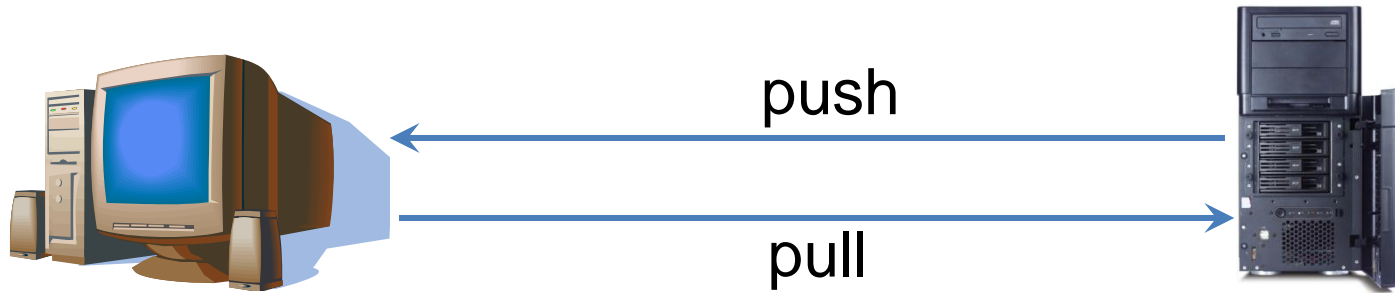
Savings Algorithm - Policies

- 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]



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)

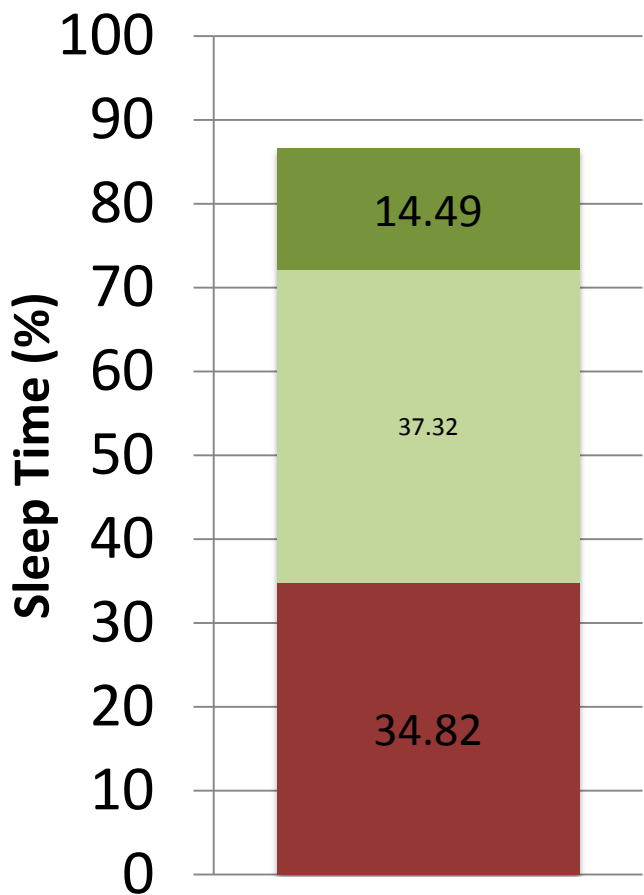
Talk outline

- 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

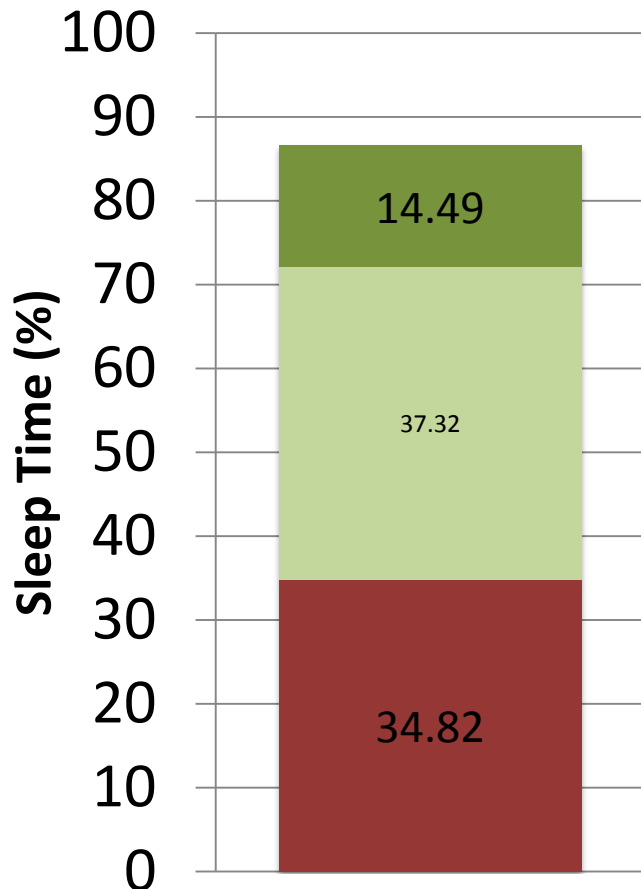
Trace driven analysis: Energy savings

- LG MaxSavings
- LG Conservative
- Existing



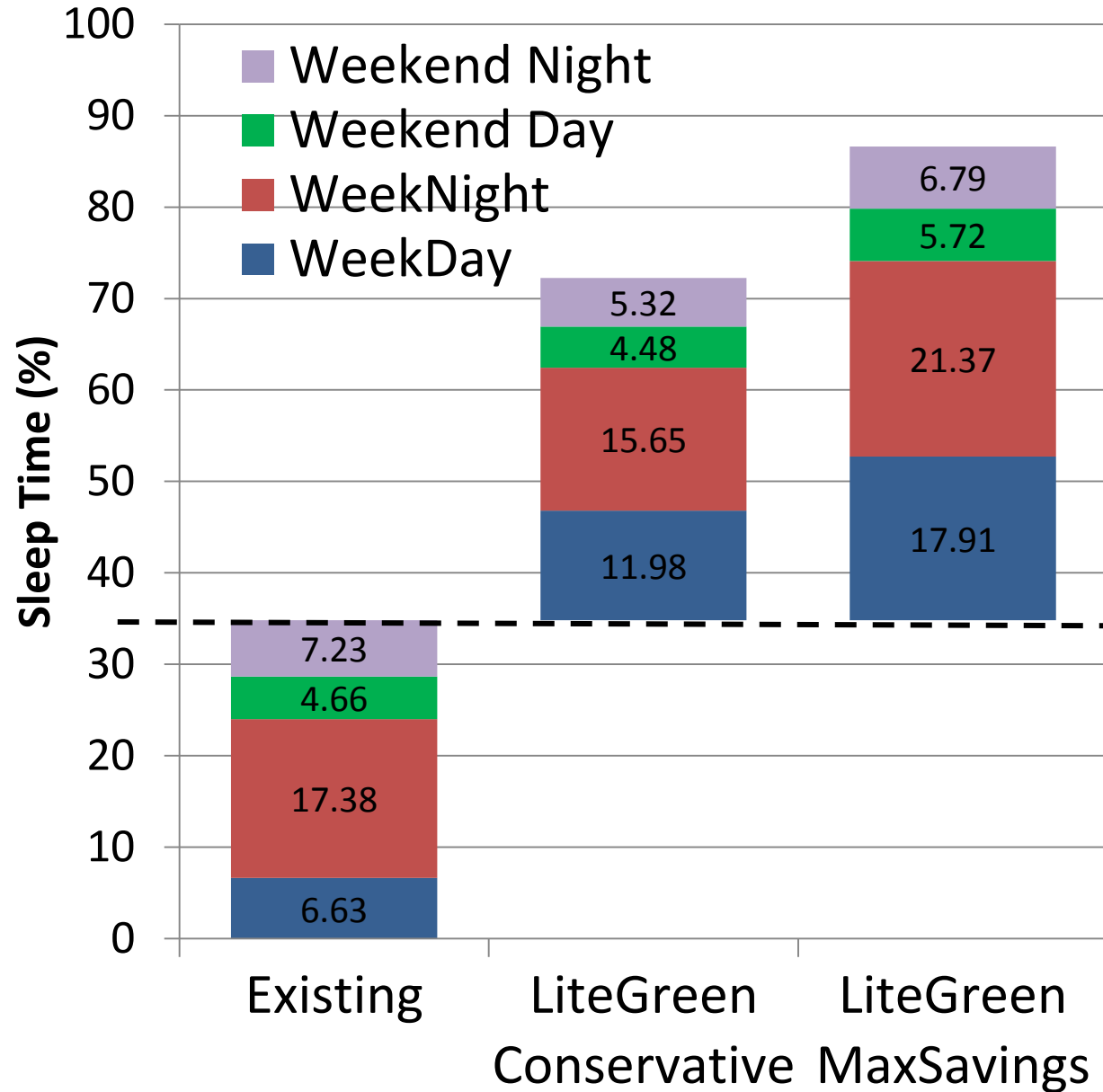
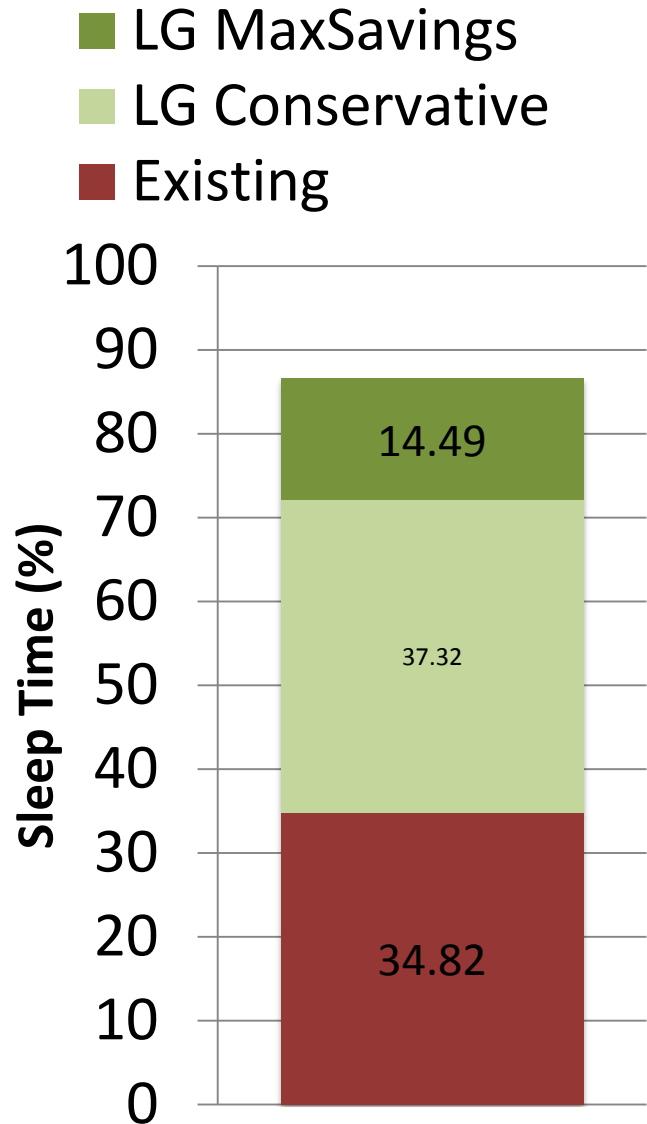
Trace driven analysis: Energy savings

- LG MaxSavings
- LG Conservative
- Existing

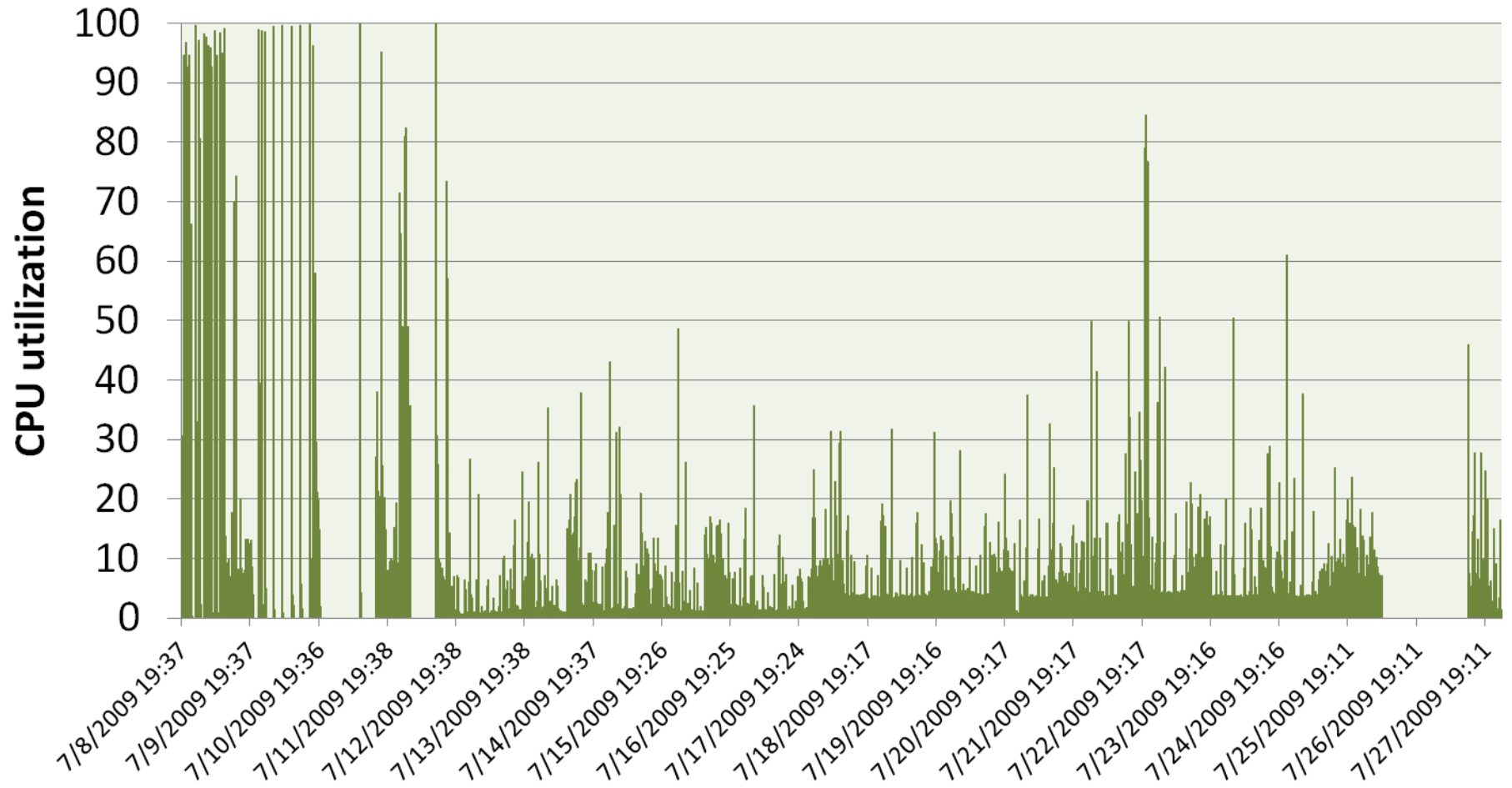


- Data gathered at MSR India, 120 machines
- 65,000 user-hours of data
- Existing power management
 - Manual switching off
 - Windows PM plans
 - Balanced (most users)
 - 30 min IdleThr (unclear what idle means)
 - Few users have high performance plan
- LiteGreen thresholds
 - CPUThr = 10%, IdleThr =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

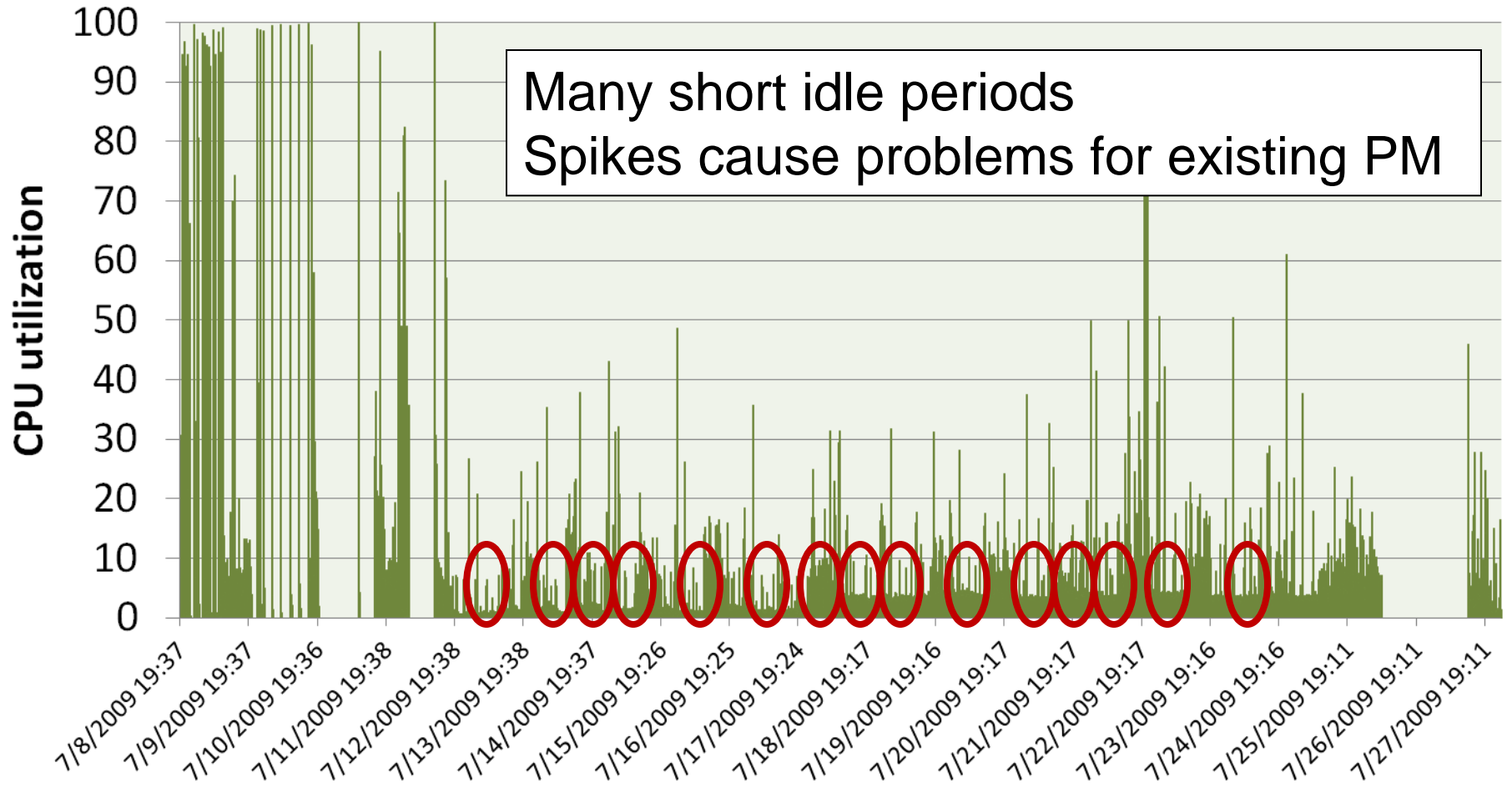
Trace driven analysis: Energy savings



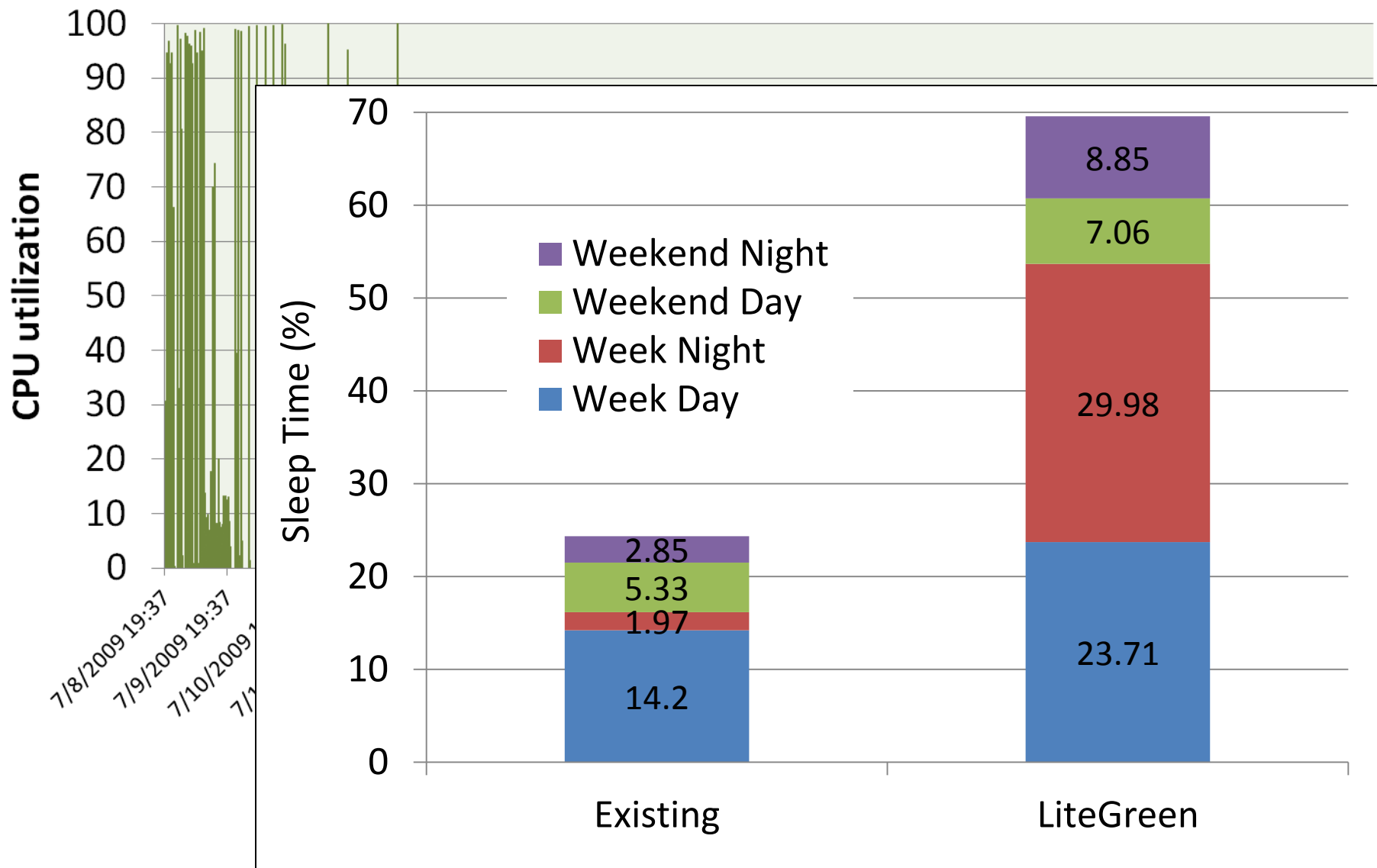
User who is always on ...



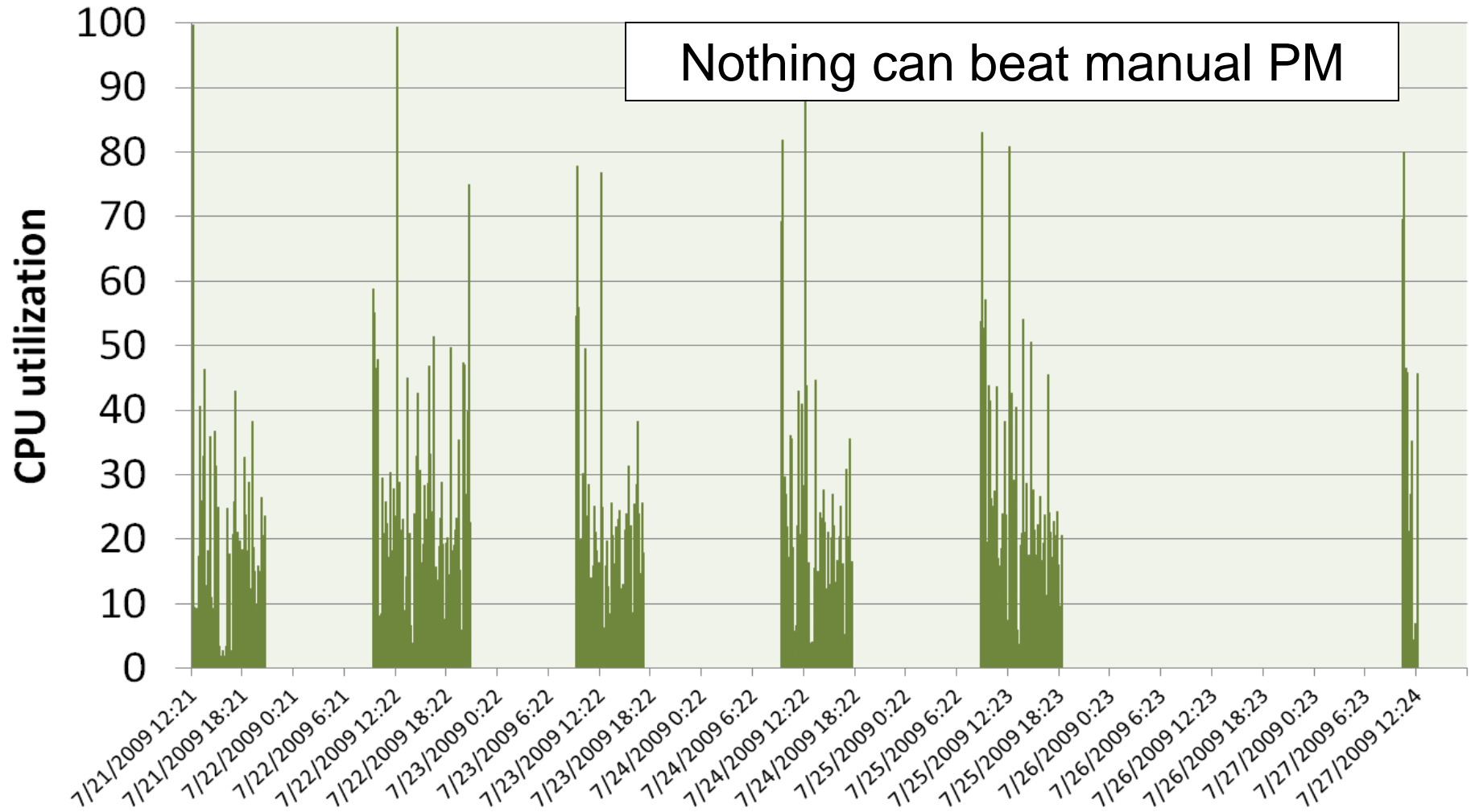
User who is always on ...



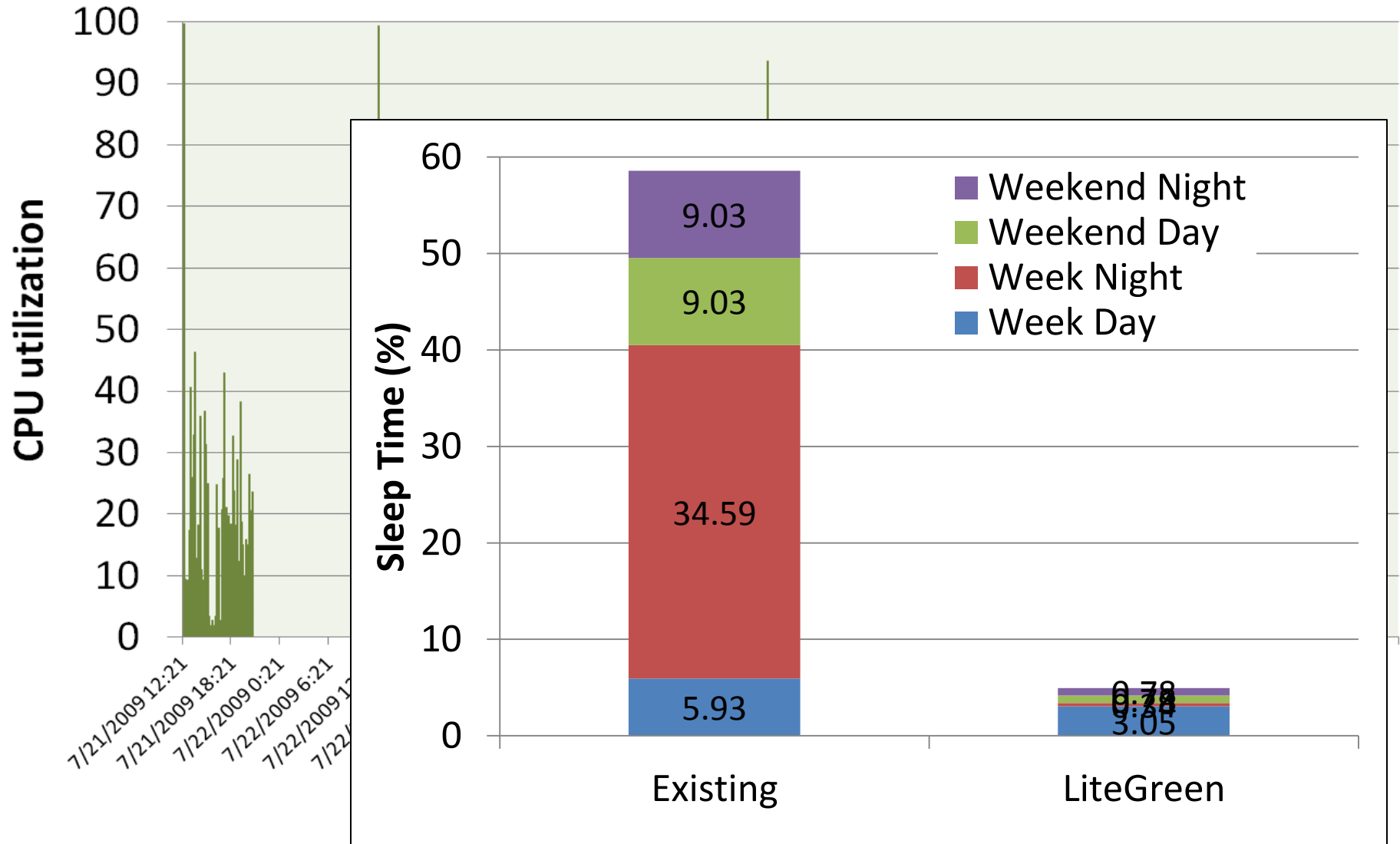
User who is always on ...



User who manually switches off



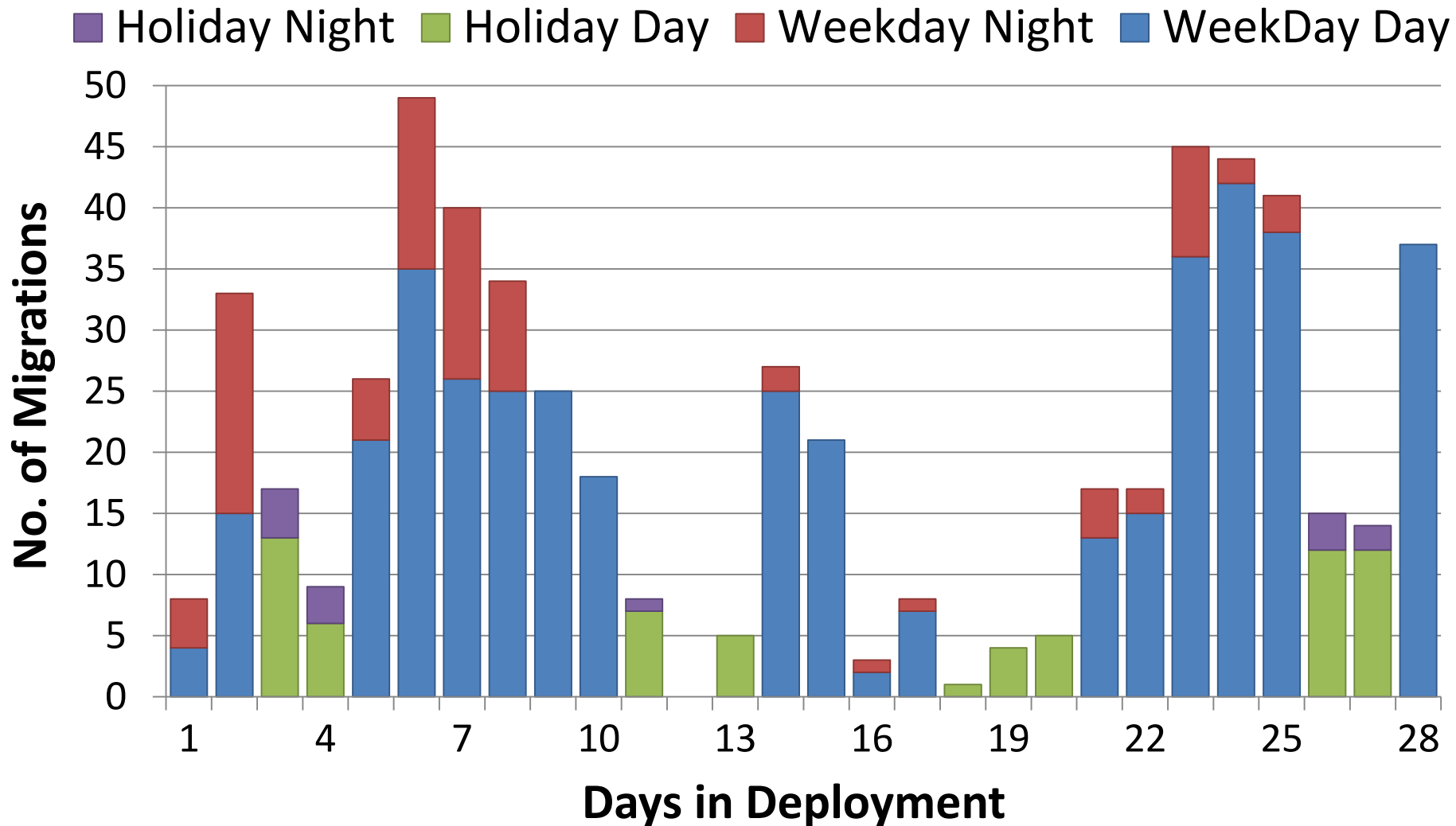
User who manually switches off



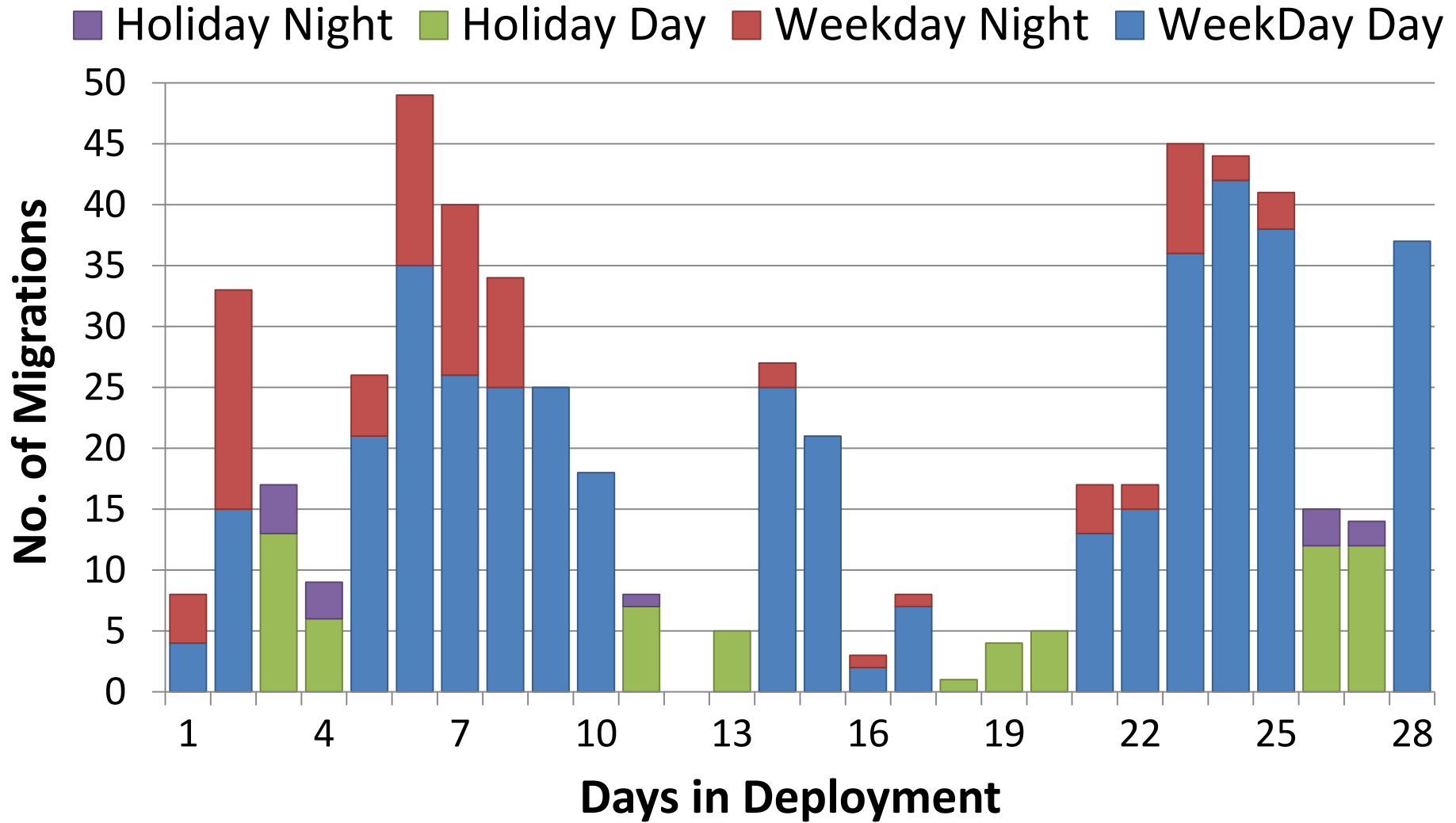
Prototype with Hyper-V and Xen

- 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

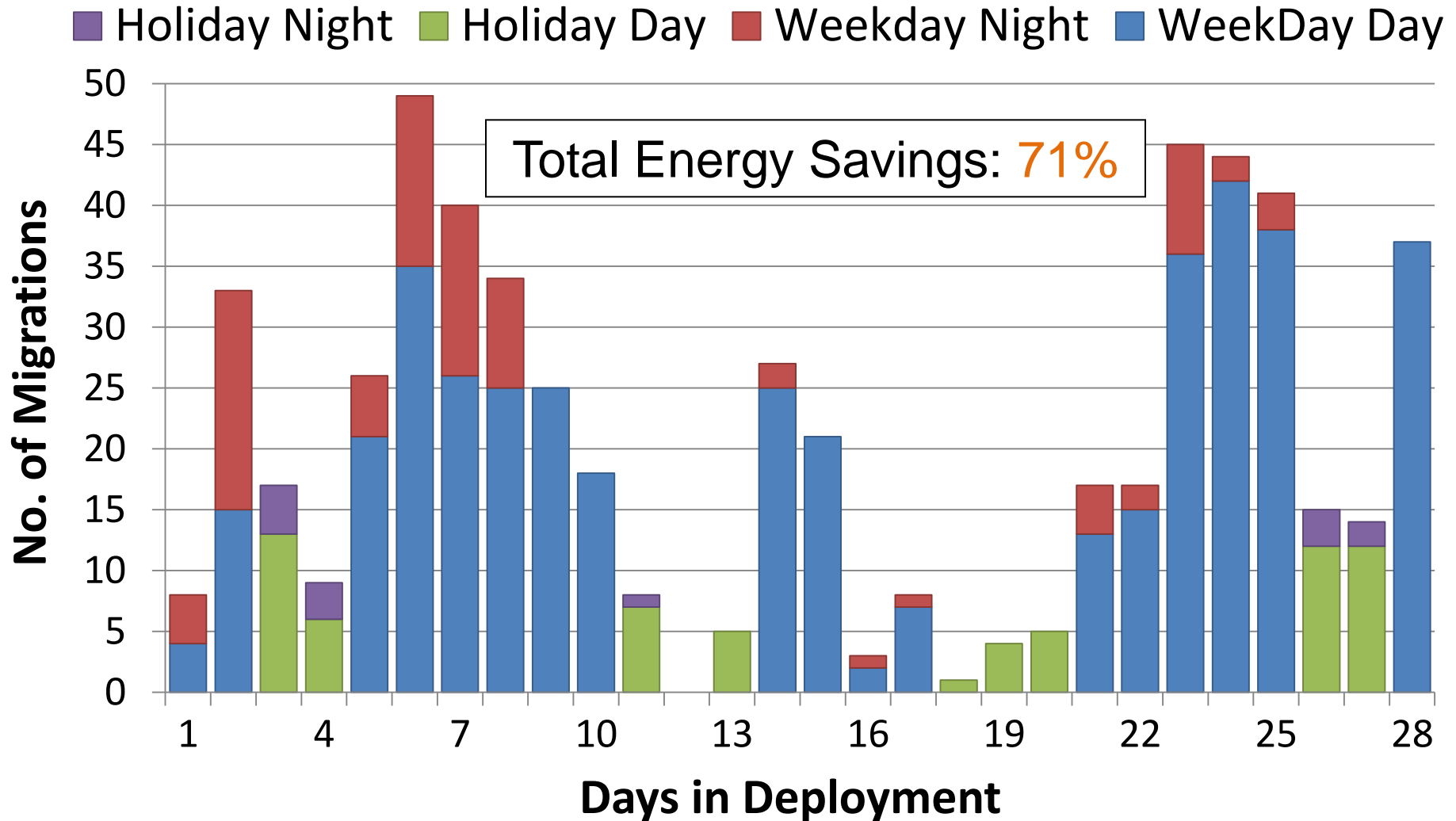


Deployment results - # of migrations



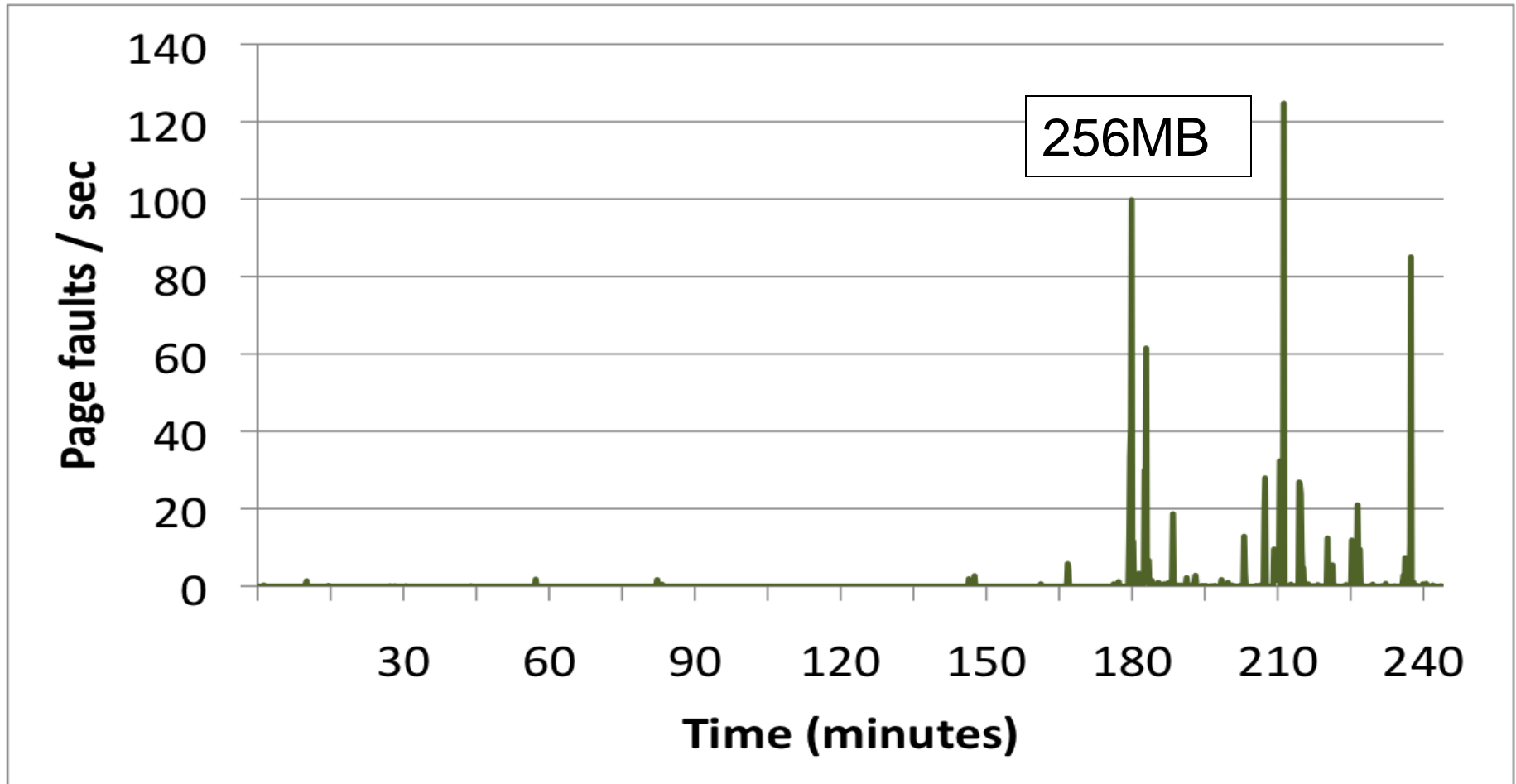
- LiteGreen can exploit idle periods during both **day and night**

Deployment results - # of migrations



- LiteGreen can exploit idle periods during both **day and night**

Xen Memory Ballooning Effects



- Every 30 minutes memory of desktop VM reduced by 128MB
- Adaptive mechanism based on page faults can be developed

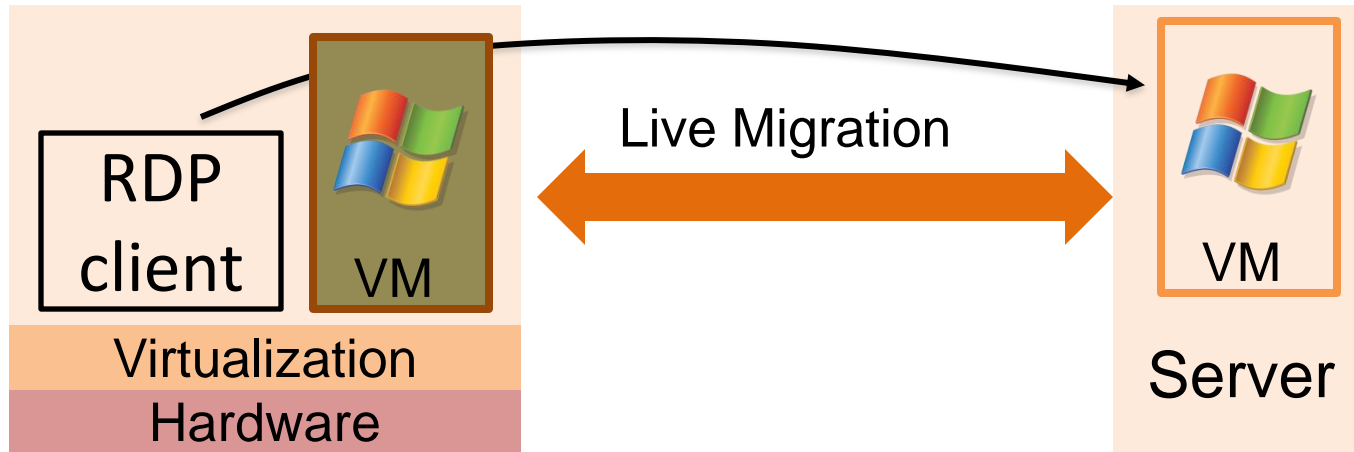
Talk outline

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

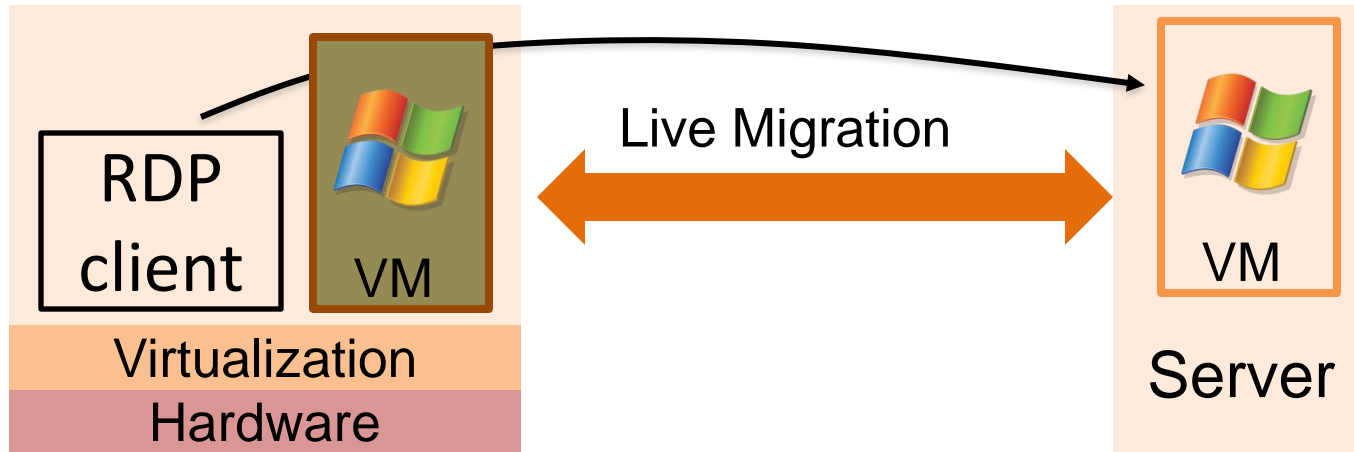
Limitations and Future work

- 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

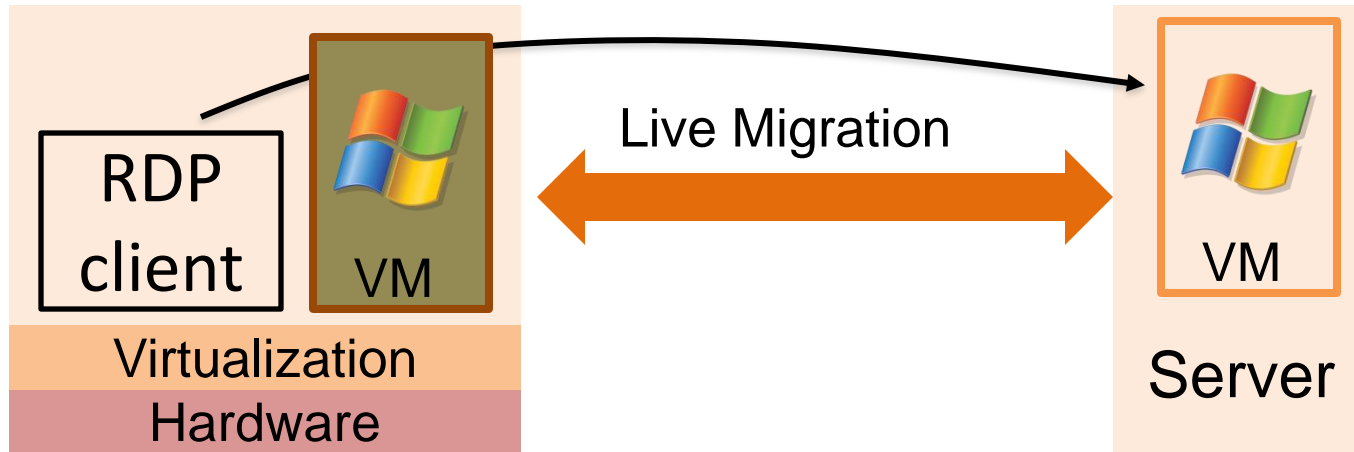


Summary: LiteGreen in one slide



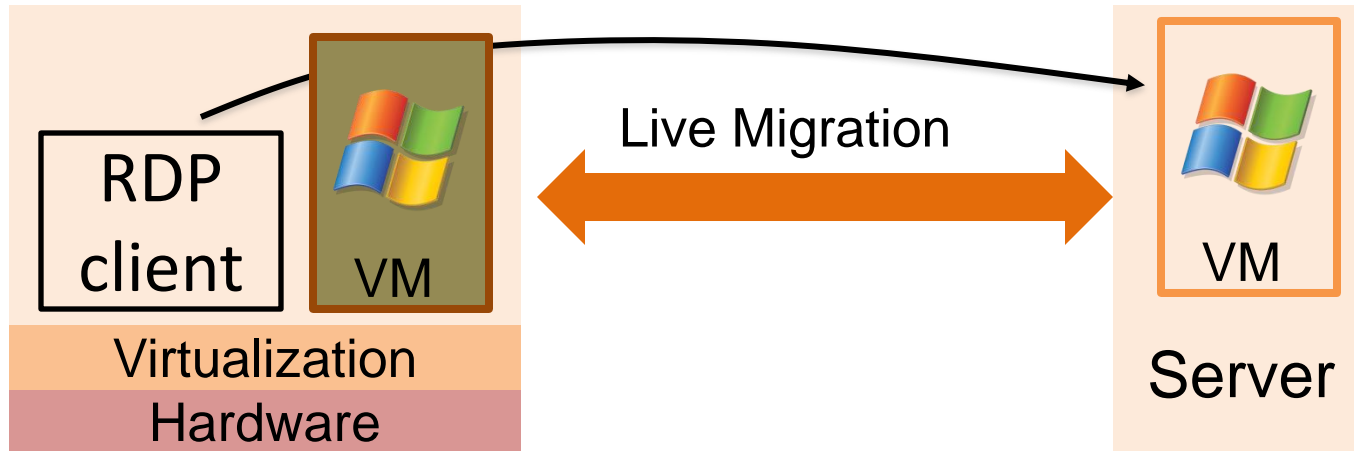
1. Significant energy savings exist in **short idle periods**

Summary: LiteGreen in one slide



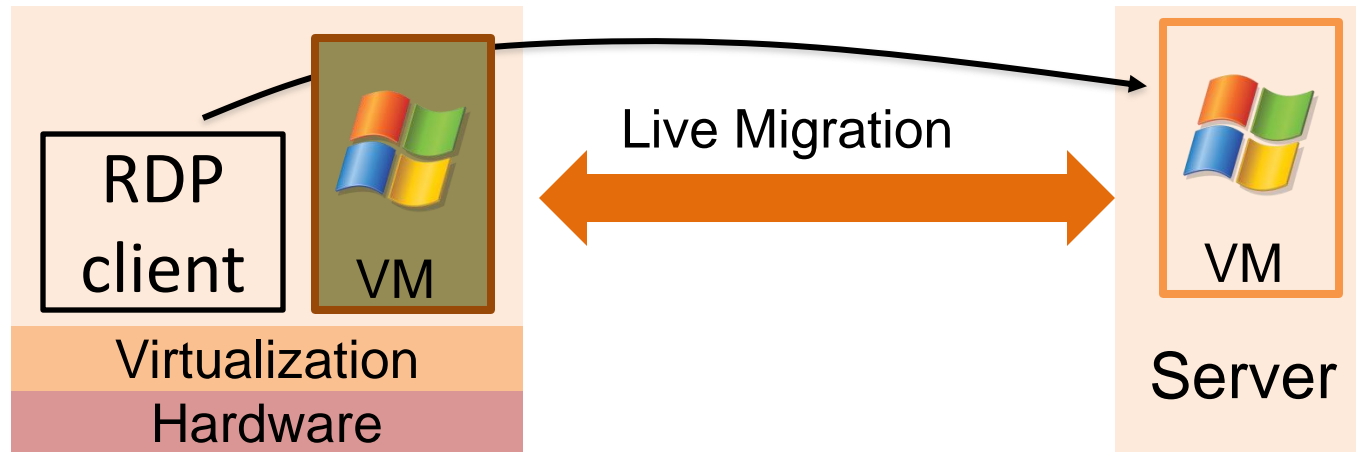
1. Significant energy savings exist in **short idle periods**
2. Holygrail = High manageability + High savings

Summary: LiteGreen in one slide



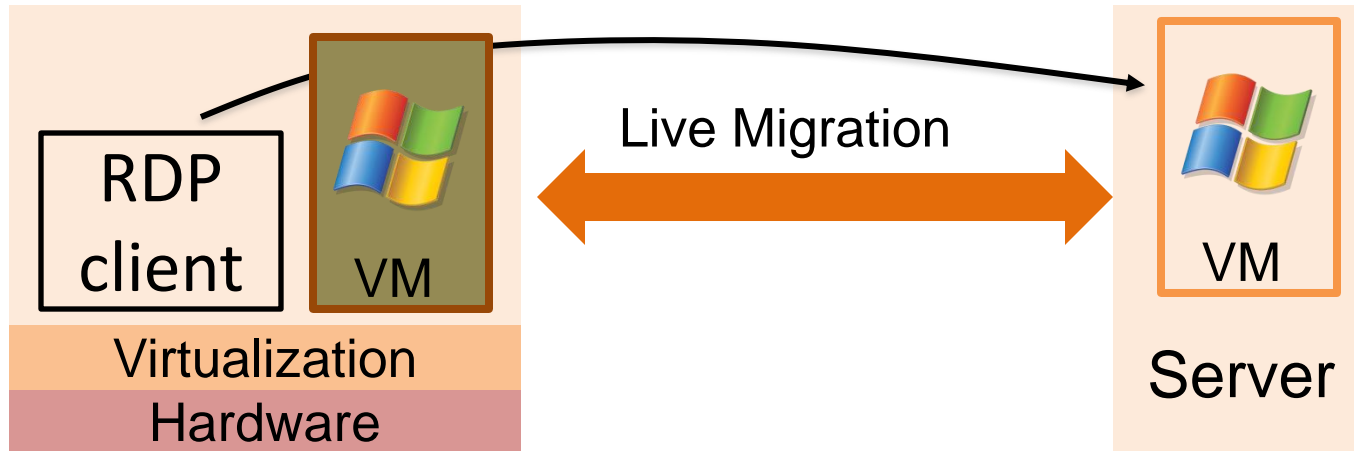
1. Significant energy savings exist in **short idle periods**
2. Holygrail = High manageability + High savings
3. LiteGreen exploits **live migration** and **RDP** to gain short idle period savings

Summary: LiteGreen in one slide



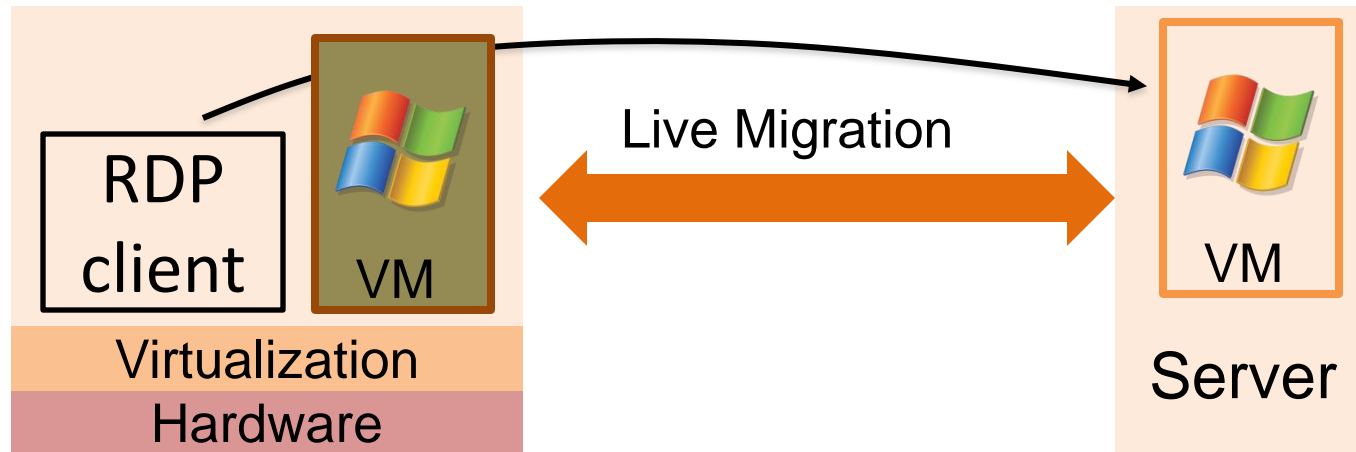
1. Significant energy savings exist in **short idle periods**
2. Holygrail = High manageability + High savings
3. LiteGreen exploits **live migration** and **RDP** to gain short idle period savings
4. LiteGreen can save **72-74%** more energy on top of existing savings (32%) for collected traces

Summary: LiteGreen in one slide



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

Summary: LiteGreen in one slide



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