Jettison: Efficient Idle Desktop Consolidation with Partial VM Migration

Nils Asmussen

Paper Reading Group

05/30/2012

Outline

1. Introduction
2. Partial VM Migration
3. Implementation of Jettison
4. Evaluation
5. Scalability
6. Discussion

Outline

1. Introduction
2. Partial VM Migration
3. Implementation of Jettison
4. Evaluation
5. Scalability
6. Discussion

Motivation

- Office computers are left running, even when idle because of applications that require always-on semantics
- Power consumption in idle mode is typically 60% of a fully utilized system
- Straightforward solution requires large network transfers, much memory on the server and long reintegrate times
- Only a small fraction of the memory is needed for running on the server (10% of memory and $< 1\%$ of disk state)

Related work

- Focused on coarse grained migration, i.e. migrating the VM completely
- Live migration: costs significant amount of network traffic, memory and time
- Ballooning: takes considerable amount of time and I/O, although its cheap to finally migrate the shrunken VM
- Remote desktop access: limited, because it doesn't allow seamless access to local devices and has bad performance

Outline

1. Introduction
2. Partial VM Migration
3. Implementation of Jettison
4. Evaluation
5. Scalability
6. Discussion

How does it work?

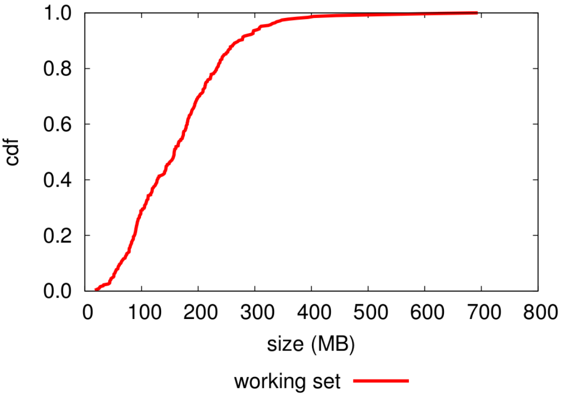
- During consolidation, partial VM migration transfers only a VM descriptor
- As the VM tries to access it's pages, it causes *remote faults* and pages are loaded from the desktop
- The disk state is also fetched on-demand from the desktop (if necessary)
- Between bursts of remote faults, the desktop can *microsleep*
- Reintegration transfers only the dirty state back to the desktop

Find answers to ...

1. When to microsleep?
2. How can prefetching improve microsleeps?

State access traces

Deployment with 3 users over 7 weeks, using Linux VMs with 4 GiB of memory:



When to microsleep?

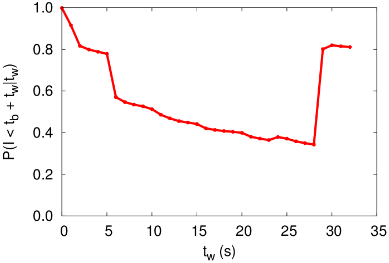
1. Formulate the idle energy usage and the energy usage when going to sleep and waking up again

When to microsleep?

1. Formulate the idle energy usage and the energy usage when going to sleep and waking up again
2. Calculate min. sleep-time to actually save energy (32s)

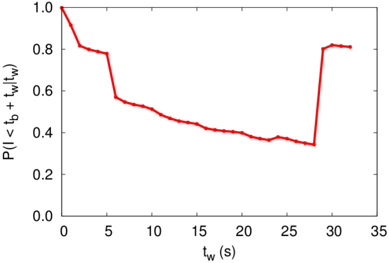
When to microsleep?

1. Formulate the idle energy usage and the energy usage when going to sleep and waking up again
2. Calculate min. sleep-time to actually save energy (32s)
3. Determine propability that the next remote fault will occur before sleeping saves energy, depending on the wait time



When to microsleep?

1. Formulate the idle energy usage and the energy usage when going to sleep and waking up again
2. Calculate min. sleep-time to actually save energy (32s)
3. Determine propability that the next remote fault will occur before sleeping saves energy, depending on the wait time



4. Determine optimal wait time by minimizing wasted energy (6s)

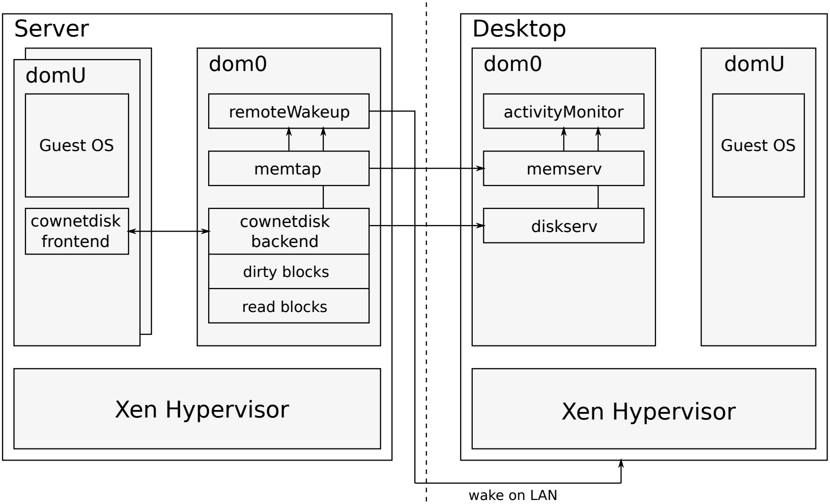
How can prefetching improve microsleeps?

- Prefetching is used to increase the frequency and length of microsleeps
- They tried 2 strategies:
  1. *hoarding*: at consolidation, transfer a few pages that have been requested in previous migrations of the same VM.
  2. *on-demand prefetch*: prefetch pages that are near the requested page
- Result: on-demand prefetch reaches longer microsleeps, which results in better energy savings

Outline

1. Introduction
2. Partial VM Migration
3. Implementation of Jettison
4. Evaluation
5. Scalability
6. Discussion

Architecture



Consolidation and reintegration

When to consolidate?

1. User idles
2. Server has enough capacity
3. VM can execute sufficiently autonomous on the server

When to reintegrate?

1. User becomes active
2. Server capacity is exceeded
3. VM becomes active (requires a large amount of state from desktop)

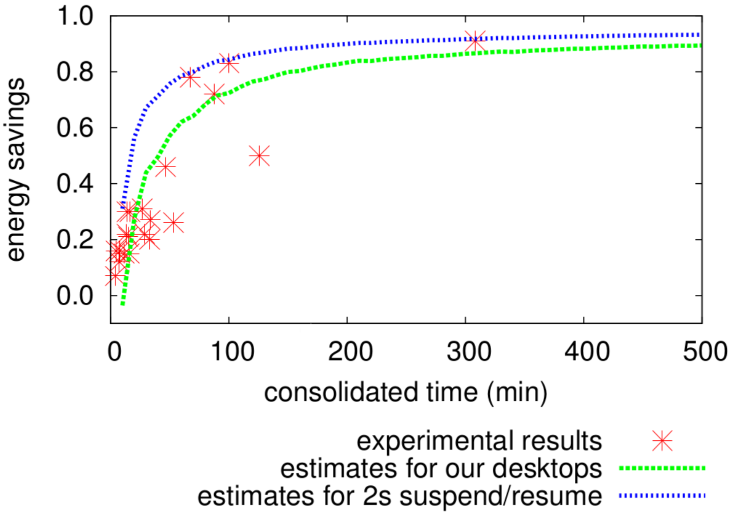
Outline

1. Introduction
2. Partial VM Migration
3. Implementation of Jettison
4. Evaluation
5. Scalability
6. Discussion

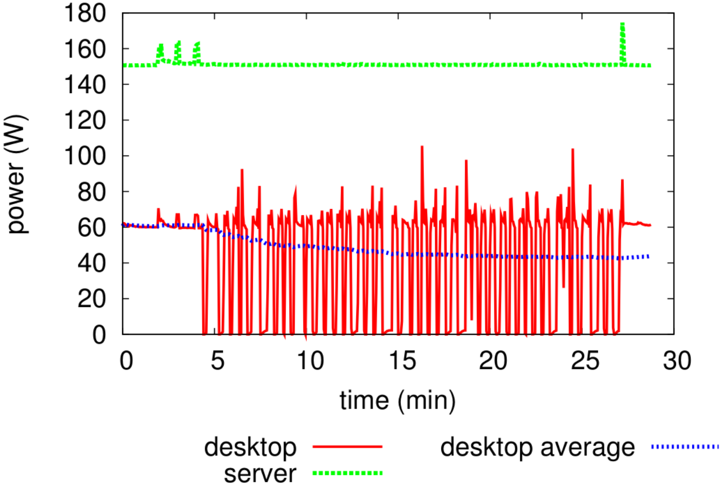
Setup

- Deployment of the prototype with 4 users over 6 days
- Desktops used standard Linux systems with 4 GiB of memory and 12 GiB of disk
- Server had 16 GiB of memory
- Connection was a 1 GiB/s ethernet switch

Energy savings



Power consumption

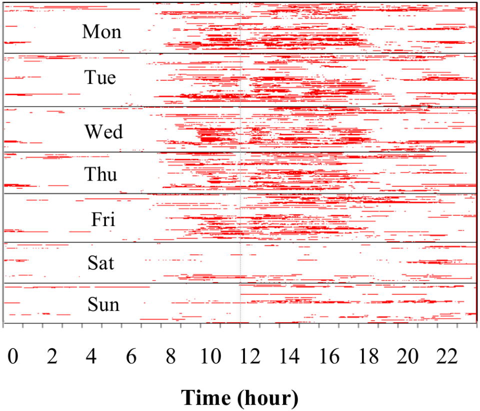


Outline

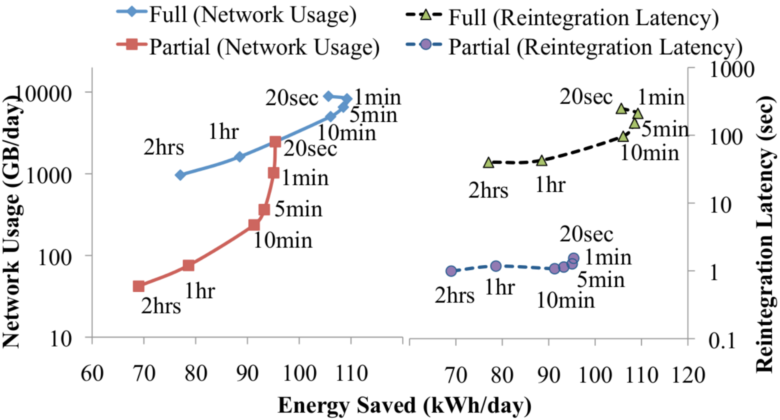
1. Introduction
2. Partial VM Migration
3. Implementation of Jettison
4. Evaluation
5. Scalability
6. Discussion

Approach

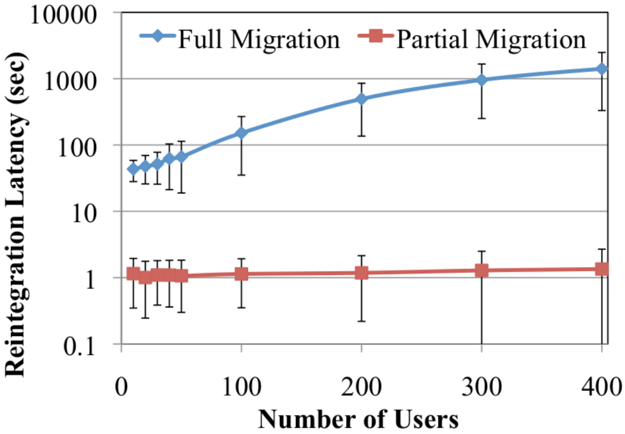
- Collect idleness traces from real users in an office environment
- Idleness was tracked for 4 months at an industrial research lab with 22 researchers



Network usage and reintegration latency



Reintegration latency



Outline

1 Introduction

2 Partial VM Migration

3 Implementation of Jettison

4 Evaluation

5 Scalability

6 Discussion

Ideas & Questions

- Do the desktop need to be virtualized?
- The whole concept is based on the assumption that systems in idle mode consume nearly as much energy as under full load. Will that be true for future systems?
- Maybe one could use the same concept to allow people to “take their system home”?