KVM Live Migration: Weather forecast

Red Hat
Juan Quintela

May 29, 2013

Abstract

In this talk we would describe the Live Migration improvements since last year, from how to move it to its own thread and the testing and tuning done to run guest with huge amounts of memory.

Agenda

Section 1

Introduction

redhat.

What is migration

- The problem:
  Moving a guest to a different host
- To make things interesting:
  Do it without stopping the guest
- Even more interesting:
  Do it fast

redhat.

What is migration

- The problem:
  Moving a guest to a different host
- To make things interesting:
  Do it without stopping the guest
- Even more interesting:
  Do it fast

redhat.

What is migration

- The problem:
  Moving a guest to a different host
- To make things interesting:
  Do it without stopping the guest
- Even more interesting:
  Do it fast

redhat.

Copy

- Performance, networking and copies don't mix well
- We were doing 2-3 copies for each data sent
- We moved to use writev and use only one copy
- Thank Orit Wasserman and Paolo Bonzini

redhat.

Copy

- Performance, networking and copies don't mix well
- We were doing 2-3 copies for each data sent
- We moved to use writev and use only one copy
- Thank Orit Wasserman and Paolo Bonzini

redhat.

Copy

- Performance, networking and copies don't mix well
- We were doing 2-3 copies for each data sent
- We moved to use writev and use only one copy
- Thank Orit Wasserman and Paolo Bonzini

redhat.

Copy

- Performance, networking and copies don't mix well
- We were doing 2-3 copies for each data sent
- We moved to use writev and use only one copy
- Thank Orit Wasserman and Paolo Bonzini

redhat.

xbzrle

- Xor Based Zero Run Length Encoding
- or the most impossible to run acronym
- Maintain a cache of the already sent pages
- Thank Orit Wasserman and initial code from Benoit

redhat.

xbzrle

- Xor Based Zero Run Length Encoding
- or the most impossible to run acronym
- Maintain a cache of the already sent pages
- Thank Orit Wasserman and initial code from Benoit

redhat.

xbzrle

- Xor Based Zero Run Length Encoding
- or the most impossible to run acronym
- Maintain a cache of the already sent pages
- Thank Orit Wasserman and initial code from Benoit

redhat.

xbzrle

- Xor Based Zero Run Length Encoding
- or the most impossible to run acronym
- Maintain a cache of the already sent pages
- Thank Orit Wasserman and initial code from Benoit

redhat.

statistics

- Measure how long it is taking
- Measure how long we expect to take
- And maintain them on real time

redhat.

statistics

- Measure how long it is taking
- Measure how long we expect to take
- And maintain them on real time

redhat.

statistics

- Measure how long it is taking
- Measure how long we expect to take
- And maintain them on real time

Section 2

Migration thread

redhat.

We use callbacks and timers

- callbacks are not so great
- especially if your callback is slow
- and runs continuously
- So, we move to our own thread

redhat.

We use callbacks and timers

- callbacks are not so great
- especially if your callback is slow
- and runs continuously
- So, we move to our own thread

redhat.

We use callbacks and timers

- callbacks are not so great
- especially if your callback is slow
- and runs continuously
- So, we move to our own thread

redhat.

We use callbacks and timers

- callbacks are not so great
- especially if your callback is slow
- and runs continuously
- So, we move to our own thread

redhat.

synchronization, what is that?

- now some data is used in both the migration thread and the iothread
- find what/where and put locks are needed
- slow tedious process
- thanks to Paolo Bonzini, Umesh Deshpande, Juan Quintela

redhat.

synchronization, what is that?

- now some data is used in both the migration thread and the iothread
- find what/where and put locks are needed
- slow tedious process
- thanks to Paolo Bonzini, Umesh Deshpande, Juan Quintela

redhat.

synchronization, what is that?

- now some data is used in both the migration thread and the iothread
- find what/where and put locks are needed
- slow tedious process
- thanks to Paolo Bonzini, Umesh Deshpande, Juan Quintela

redhat.

synchronization, what is that?

- now some data is used in both the migration thread and the iothread
- find what/where and put locks are needed
- slow tedious process
- thanks to Paolo Bonzini, Umesh Deshpande, Juan Quintela

redhat

Section 3

Disk migration

redhat.

The good news

- We got a new disk migration code
- it works well
- it is more flexible
- Thanks to Paolo Bonzini

redhat.

The good news

- We got a new disk migration code
- it works well
- it is more flexible
- Thanks to Paolo Bonzini

redhat.

The good news

- We got a new disk migration code
- it works well
- it is more flexible
- Thanks to Paolo Bonzini

redhat.

The good news

- We got a new disk migration code
- it works well
- it is more flexible
- Thanks to Paolo Bonzini

redhat.

The bad news

- We were going to remove the old block-migration code
- Then people fixed it
- Good: it works now
- Bad: We have to maintain both
- It uses the same port than migration
- You need to migrate all/none of block devices

redhat.

The bad news

- We were going to remove the old block-migration code
- Then people fixed it
- Good: it works now
- Bad: We have to maintain both
- It uses the same port than migration
- You need to migrate all/none of block devices

redhat.

The bad news

- We were going to remove the old block-migration code
- Then people fixed it
- Good: it works now
- Bad: We have to maintain both
- It uses the same port than migration
- You need to migrate all/none of block devices

redhat.

The bad news

- We were going to remove the old block-migration code
- Then people fixed it
- Good: it works now
- Bad: We have to maintain both
- It uses the same port than migration
- You need to migrate all/none of block devices

redhat.

The bad news

- We were going to remove the old block-migration code
- Then people fixed it
- Good: it works now
- Bad: We have to maintain both
- It uses the same port than migration
- You need to migrate all/none of block devices

redhat.

The bad news

- We were going to remove the old block-migration code
- Then people fixed it
- Good: it works now
- Bad: We have to maintain both
- It uses the same port than migration
- You need to migrate all/none of block devices

redhat

Section 4

Live migration of large guests

redhat.

News at 11

- Each year servers have more CPU's
- Each year servers have more memory
- Ergo, each year precopy has life more difficult
- And we want them to be able to migrate with minimum downtime
- Thanks to Vinod Chegu

redhat.

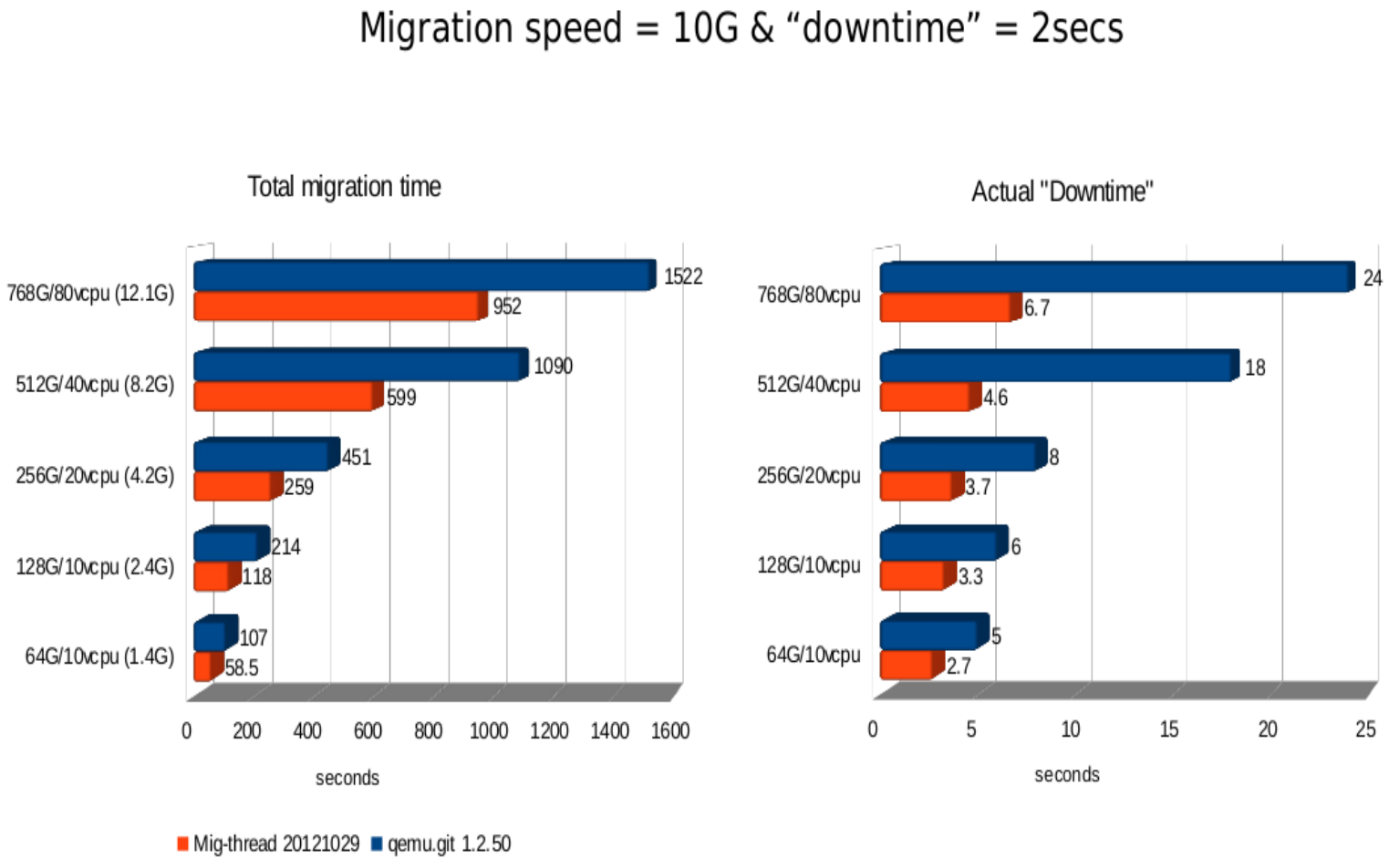
Testing, tuning, testing, ...

- Run test
- Profile
- Fix bottleneck
- Repeat

redhat.

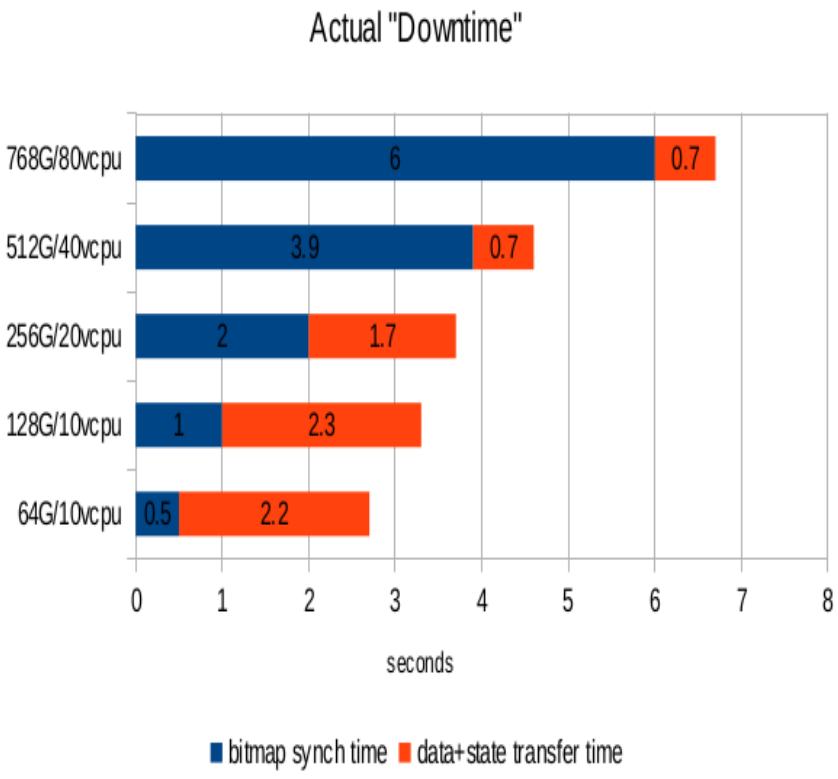
Recent optimization

Idle guest



redhat.

Observations



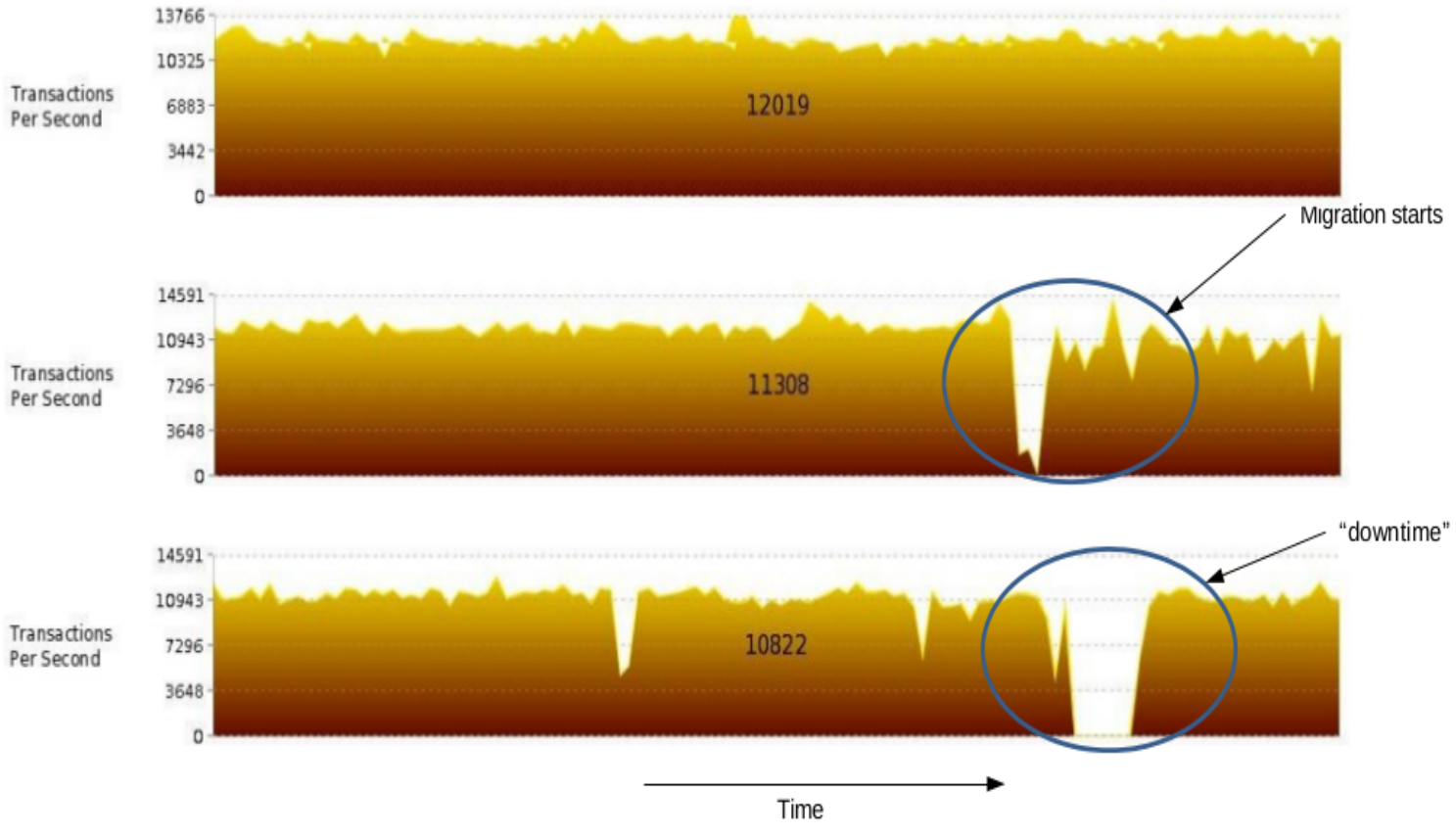
- Bitmap synch-ups for large guests
  - Major contributor to the actual “downtime”.
  - Guest freezes during the start of the migration !
- Utilization of allocated B/W
  - Peaks at ~3 Gbps.
  - Perhaps not enough data ready to be sent through the allocated pipe. i.e. Unable to saturate.

redhat.

OLTP workload

(128G/80VCPUs, 40% SGA, 75 users (CR, BP, OP, PO, BO))

Migration speed =10G , “downtime” = 4secs



Section 5

Future work

redhat.

dirty bitmap

- What pages have been dirtied
- Use 1 byte/page
- Move to 1 bit/page
- bitmap size vs main memory size

	1GB	16GB	64GB	256GB	512GB
1 bit	32KB	512KB	2MB	8MB	16MB
8 bits	256KB	4MB	16MB	64MB	128MB

redhat.

dirty bitmap

- What pages have been dirtied
- Use 1 byte/page
- Move to 1 bit/page
- bitmap size vs main memory size

	1GB	16GB	64GB	256GB	512GB
1 bit	32KB	512KB	2MB	8MB	16MB
8 bits	256KB	4MB	16MB	64MB	128MB

redhat.

dirty bitmap

- What pages have been dirtied
- Use 1 byte/page
- Move to 1 bit/page
- bitmap size vs main memory size

	1GB	16GB	64GB	256GB	512GB
1 bit	32KB	512KB	2MB	8MB	16MB
8 bits	256KB	4MB	16MB	64MB	128MB

redhat.

dirty bitmap

- What pages have been dirtied
- Use 1 byte/page
- Move to 1 bit/page
- bitmap size vs main memory size

	1GB	16GB	64GB	256GB	512GB
1 bit	32KB	512KB	2MB	8MB	16MB
8 bits	256KB	4MB	16MB	64MB	128MB

redhat.

CPU throttling

- If migration is not converging, throttle down cpus until it converge
- Vinod Chegu

redhat.

Continuous VMState testing

- Ensure that we save all the needed state
- Do it during the whole time that we run the guest

redhat.

Postcopy?

- Currently we use precopy: send date while we run in source
- Postcopy: run on target and page fault over the network

redhat.

RDMA

- Can we saturate infiniband while doing migration?
- We can even try, problems from what is being done ¡put reference to patches¿

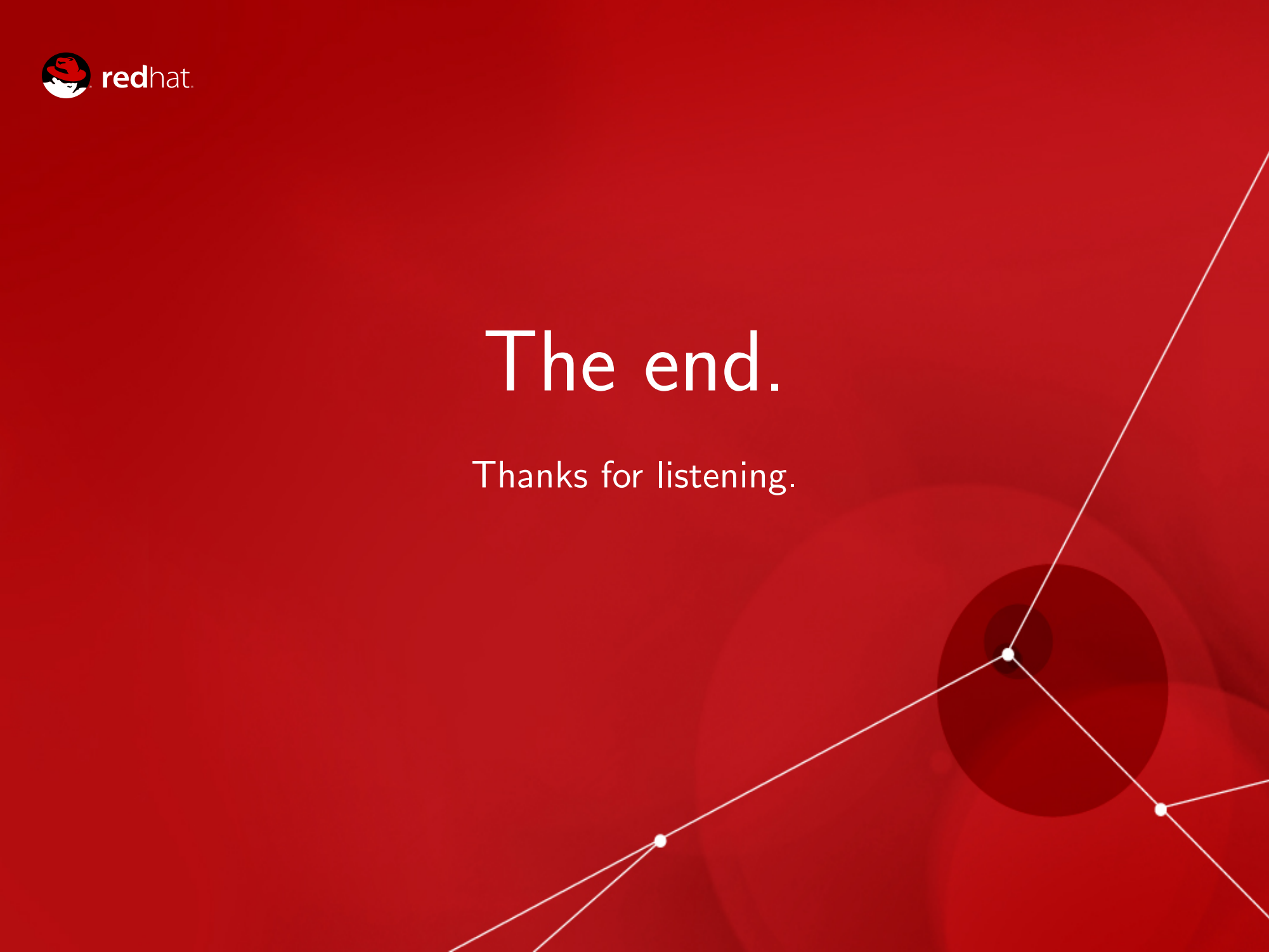
redhat.

Fault Tolerance

- If we can migrate
- Why can't we do it continuously so when one machine breaks we continue on the destination
- This needs to be fast and reliable
- One prototype with Kemari

Section 6

Questions


redhat.
The end.
Thanks for listening.