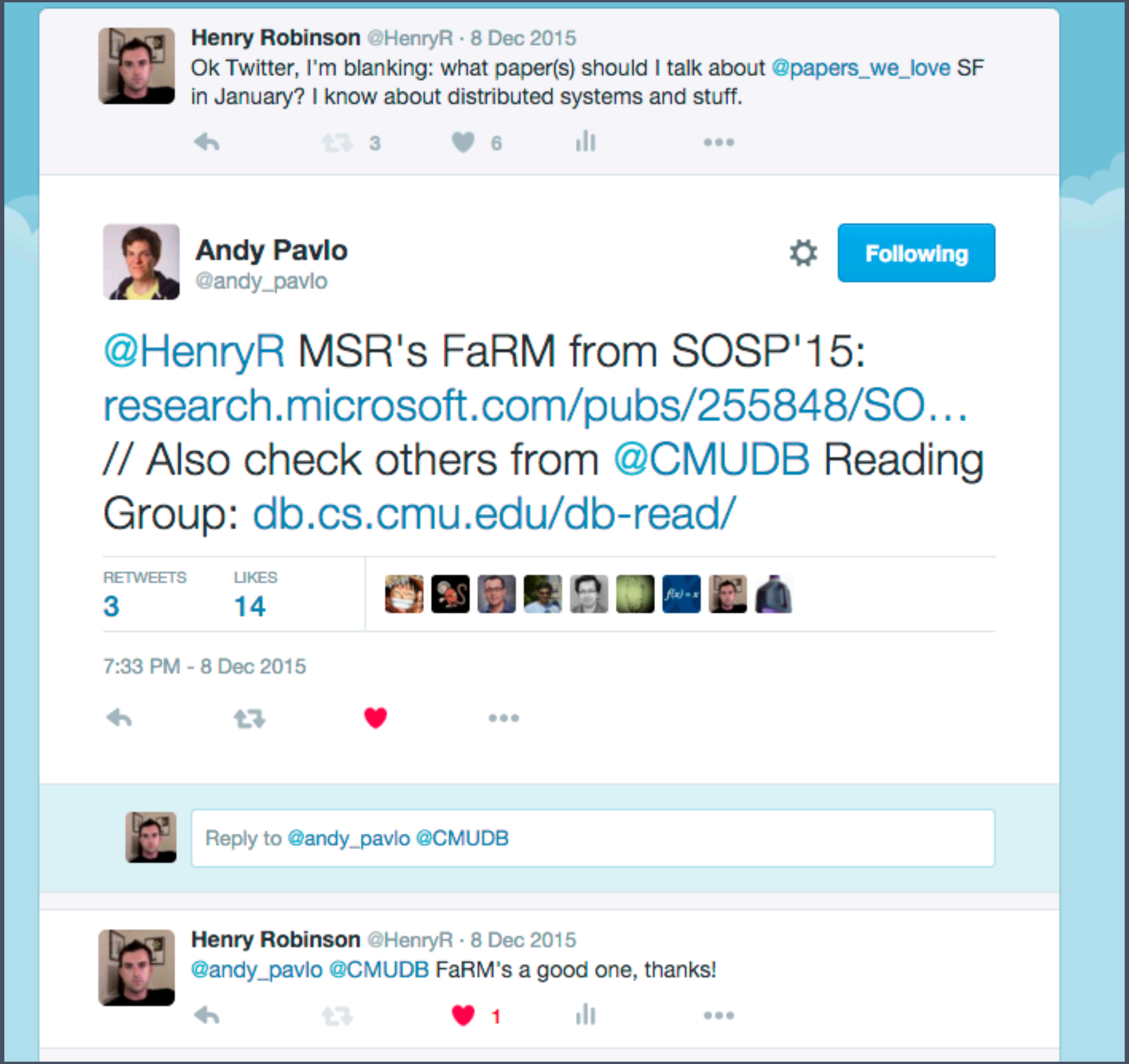
NO COMPROMISES: DISTRIBUTED TRANSACTIONS WITH CONSISTENCY, AVAILABILITY AND PERFORMANCE

DRAGOJEVIC ET. AL., SOSP '15

TODAY

- OVERVIEW OF FARM, PLUS TECHNOLOGICAL CONTEXT
- NO PROOFS THIS TIME! (YAY)
- ONLY CURSORY OVERVIEW OF RECOVERY PROTOCOL



Henry Robinson @HenryR · 8 Dec 2015

Ok Twitter, I'm blanking: what paper(s) should I talk about @papers_we_love SF in January? I know about distributed systems and stuff.

Andy Pavlo
@andy_pavlo

@HenryR MSR's FaRM from SOSP'15: research.microsoft.com/pubs/255848/SO… // Also check others from @CMUDB Reading Group: db.cs.cmu.edu/db-read/

RETWEETS 3 LIKES 14

7:33 PM - 8 Dec 2015

Reply to @andy_pavlo @CMUDB

Henry Robinson @HenryR · 8 Dec 2015

@andy_pavlo @CMUDB FaRM's a good one, thanks!

WHAT'S TO LOVE?

1. CHALLENGE TO ORTHODOXY

2. FORWARD LOOKING

.. (WITHOUT BEING OVERLY SPECULATIVE)

3. ENGINEERING IS GREAT

DO WE NEED TO COMPROMISE?

1980S: DISKS ARE SLOW AND MEMORY IS SMALL

1980S: DISKS ARE SLOW AND MEMORY IS SMALL ... SO LET'S INVENT GRACE JOIN AND FRIENDS.[1]

[1] 'IMPLEMENTATION TECHNIQUES FOR MAIN MEMORY DATABASE SYSTEMS', DEWITT ET. AL., SIGMOD'84

1990S: WANS ARE SLOW!

1990S: WANS ARE SLOW!

... SO LET'S BUILD A CROSS-SITE OPTIMIZER[2]

[2] 'MARIPOSA: A WIDE-AREA DISTRIBUTED DATABASE SYSTEM', STONEBRAKER ET. AL.

2000S: MEMORY IS SLOW!

2000S: MEMORY IS SLOW!

... SO LET'S BUILD A CACHE-EFFICIENT JOIN ALGORITHM (X-100)[3]

[3] 'DATABASE ARCHITECTURE OPTIMIZED FOR THE NEW BOTTLENECK, MEMORY ACCESS', BONCZ ET. AL., VLDB'99

2010: DISKS ARE SLOW AGAIN!

2010: DISKS ARE SLOW AGAIN!

... SO LET'S PUT LOTS OF THEM IN A SINGLE MACHINE!

DATABASE SYSTEM DESIGN CAN BE VIEWED AS AN EXERCISE IN CHASING A MOVING TARGET.

2015: CPUS ARE GOING TO BECOME SLOW

2015: CPUS ARE GOING TO BECOME SLOW

... WHAT CAN WE DO ABOUT IT?

WHY ARE CPUS GOING TO BECOME SLOW?

- NON-VOLATILE STORAGE IS GOING TO GET MUCH, MUCH QUICKER
  - MESSAGE LATENCY IS GOING TO DECREASE

WHY ARE CPUS GOING TO BECOME SLOW?

- NON-VOLATILE STORAGE IS GOING TO GET MUCH, MUCH QUICKER
  - MESSAGE LATENCY IS GOING TO DECREASE

AND BOTH WILL BECOME AFFORDABLE IN DATACENTERS

FASTER NON-VOLATILE STORAGE

- ADD A UPS TO MAIN MEMORY
  - WHEN POWER IS LOST, WRITE TO SSD!
- NV-DRAM IS NOT NEW, BUT THIS IS A CHEAP (EFFECTIVE) HACK.

LOW-LATENCY IN-DATACENTER MESSAGING

- REMOTE DIRECT MEMORY ACCESS (RDMA) IS A LOW-LATENCY LINK (V1) OR IP (V2)-LEVEL PROTOCOL
- ALLOWS MACHINES TO DIRECTLY ACCESS MEMORY OF REMOTE PEERS
  - WITH NO CPU INVOLVEMENT AT ALL!
- INFINIBAND WAS EXPENSIVE, BUT RDMA-OVER-ETHERNET (ROCE) IS CHEAPER AND BECOMING POPULAR.

DISTRIBUTED DATABASE CONTEXT

DURABILITY REQUIRES WRITES TO NON-VOLATILE STORAGE

MESSAGING IS EXTREMELY CPU EXPENSIVE

THE CPU COST OF AN RPC:

- INTERRUPT FOR KERNEL SERVICE
- MEMORY COPY INTO KERNEL
- COPY INTO USERSPACE
- WAKE-UP HANDLER THREAD
- DE-SERIALIZE MESSAGE
- DO SOMETHING

THE CPU COST OF AN RPC:

- INTERRUPT FOR KERNEL SERVICE
- MEMORY COPY INTO KERNEL
- COPY INTO USERSPACE
- WAKE-UP HANDLER THREAD
- DE-SERIALIZE MESSAGE
- DO SOMETHING

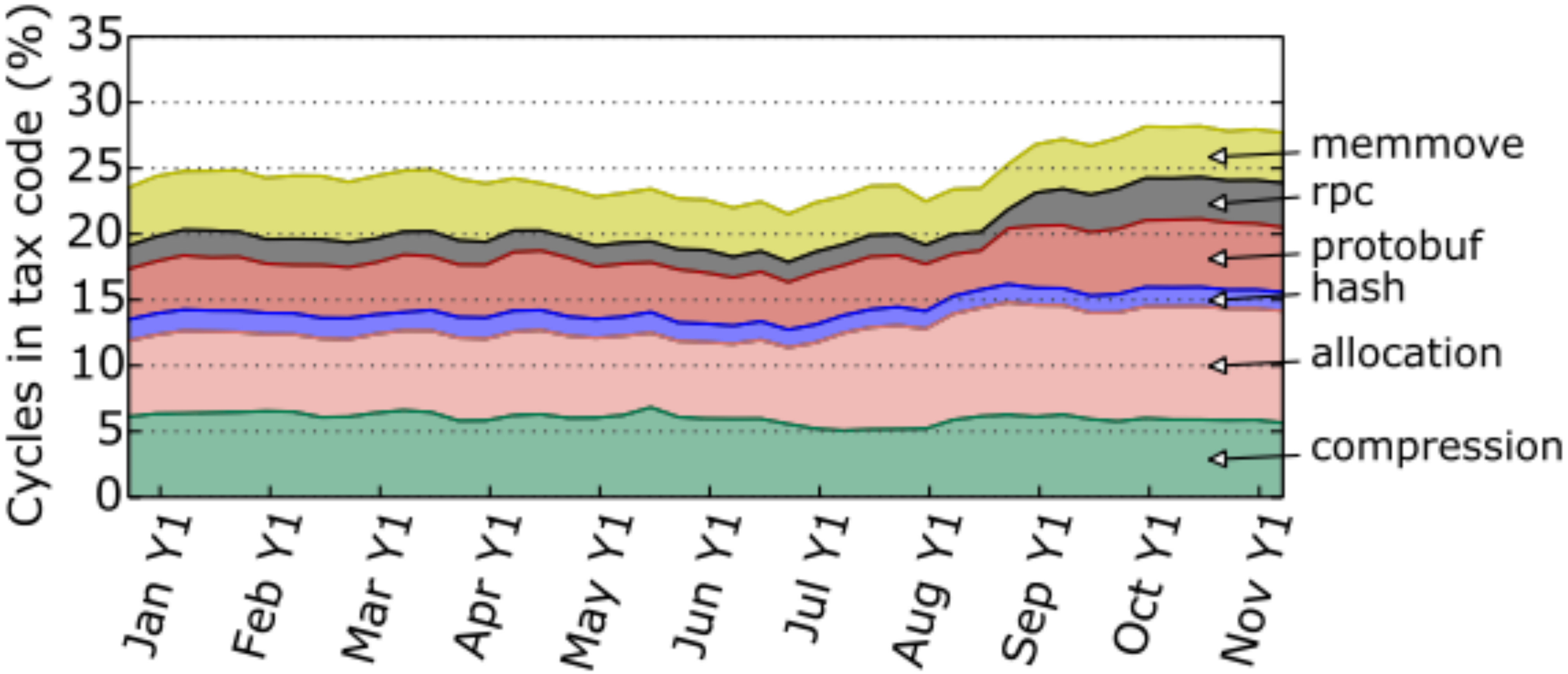


Figure 4: 22-27% of WSC cycles are spent in different components of "datacenter tax".

[4]

[4] 'Profiling a Warehouse-Scale Computer', Kanev et. al. ISCA'15

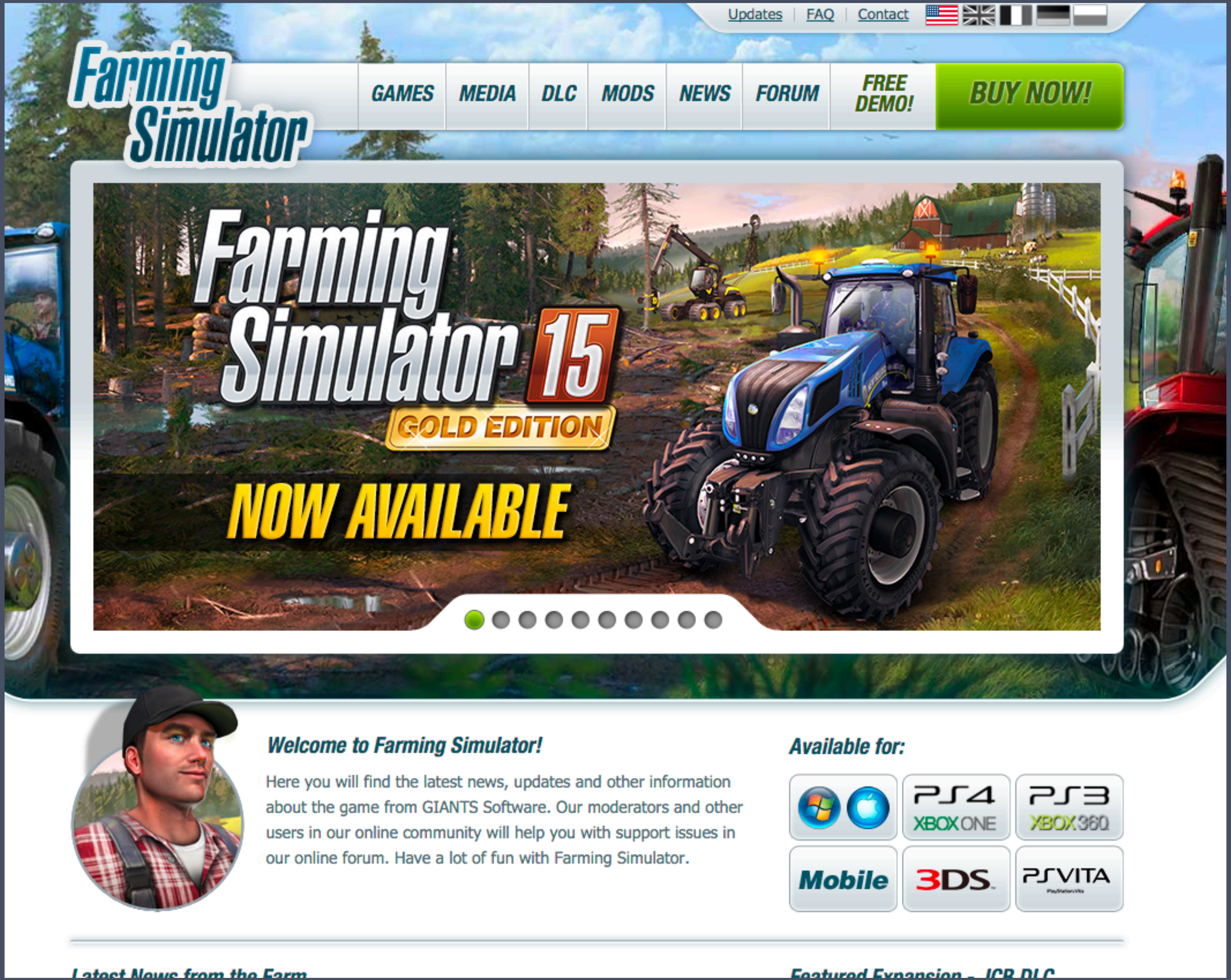
RDMA

- NO CPU ON THE USUAL WRITE OR READ PATH
- NIC HAS ITS OWN SET OF PAGE TABLES (WITHOUT PAGING)
- ADDRESS MEMORY REGIONS DIRECTLY
- FARM USES TWO DATA STRUCTURES:
    - TRANSACTIONAL LOG
    - MESSAGING RING-BUFFER

FARM

TWO PAPERS:

- 'NO COMPROMISES...', DRAGOJEVIC ET. AL., SOSP'15
- 'FARM: FAST REMOTE MEMORY', DRAGOJEVIC ET. AL., NSDI'14



Welcome to Farming Simulator!

Here you will find the latest news, updates and other information about the game from GIANTS Software. Our moderators and other users in our online community will help you with support issues in our online forum. Have a lot of fun with Farming Simulator.

Latest News from the Farm

Featured Expansion - JCB DLC

MAIN CONTRIBUTIONS:

- VERY LOW-LATENCY, HIGH-THROUGHPUT TRANSACTIONAL SYSTEM.
- VERY FAST FAILURE DETECTION / RECOVERY PROTOCOL
- UNUSUAL DISTRIBUTED SYSTEM ARCHITECTURE BASED ON VERTICAL PAXOS
- COMMIT PROTOCOL OPTIMISED FOR RDMA / LOW MESSAGE COUNT

WHAT YOU GET: ABSTRACTIONS

- GLOBAL ADDRESS SPACE OF ADDRESSABLE MEMORY
- TRANSACTIONAL API, INCLUDING LOCK-FREE READS

PROGRAMMING MODEL

- APPLICATION THREADS RUN IN FARM SERVERS
- CAN PERFORM ARBITRARY LOGIC DURING TRANSACTION (BUT NO SIDE-EFFECTS, PLEASE!)
- MAY HAVE TO DEAL WITH ANOMOLIES ON READ, THANKS TO OPTIMISTIC COMMIT

SYSTEM ARCHITECTURE

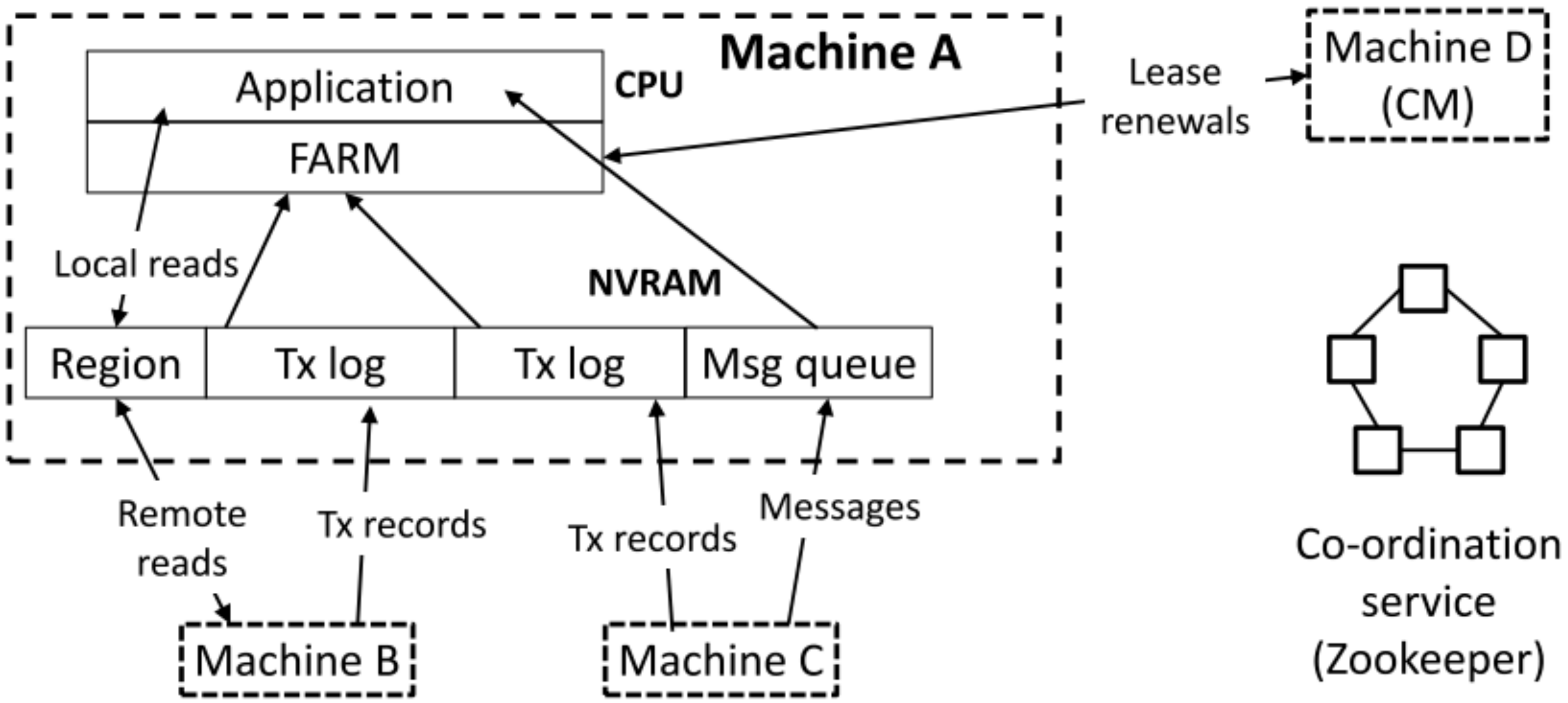


Figure 3. FaRM architecture

ADDRESSABLE MEMORY: REGIONS

- MEMORY IS PARTITIONED INTO 2GB REGIONS, PINNED INTO MEMORY ON EACH MACHINE
- REGIONS ARE SERVED BY A PRIMARY, BUT HAVE F BACKUPS
  - REGION->PRIMARY MAPPING IS MAINTAINED BY THE 'CONFIGURATION MANAGER'
- REGIONS MAY BE CO-LOCATED AT APPLICATION'S BEHEST

HOW A CHUNK OF MEMORY BECOMES A REGION

- TWO-PHASE COMMIT FROM CM (INITIATED BY MACHINE)
- ENSURES THAT ALL REPLICAS HAVE MAPPING BEFORE IT GETS USED

REGION MAPPING RECOVERY?

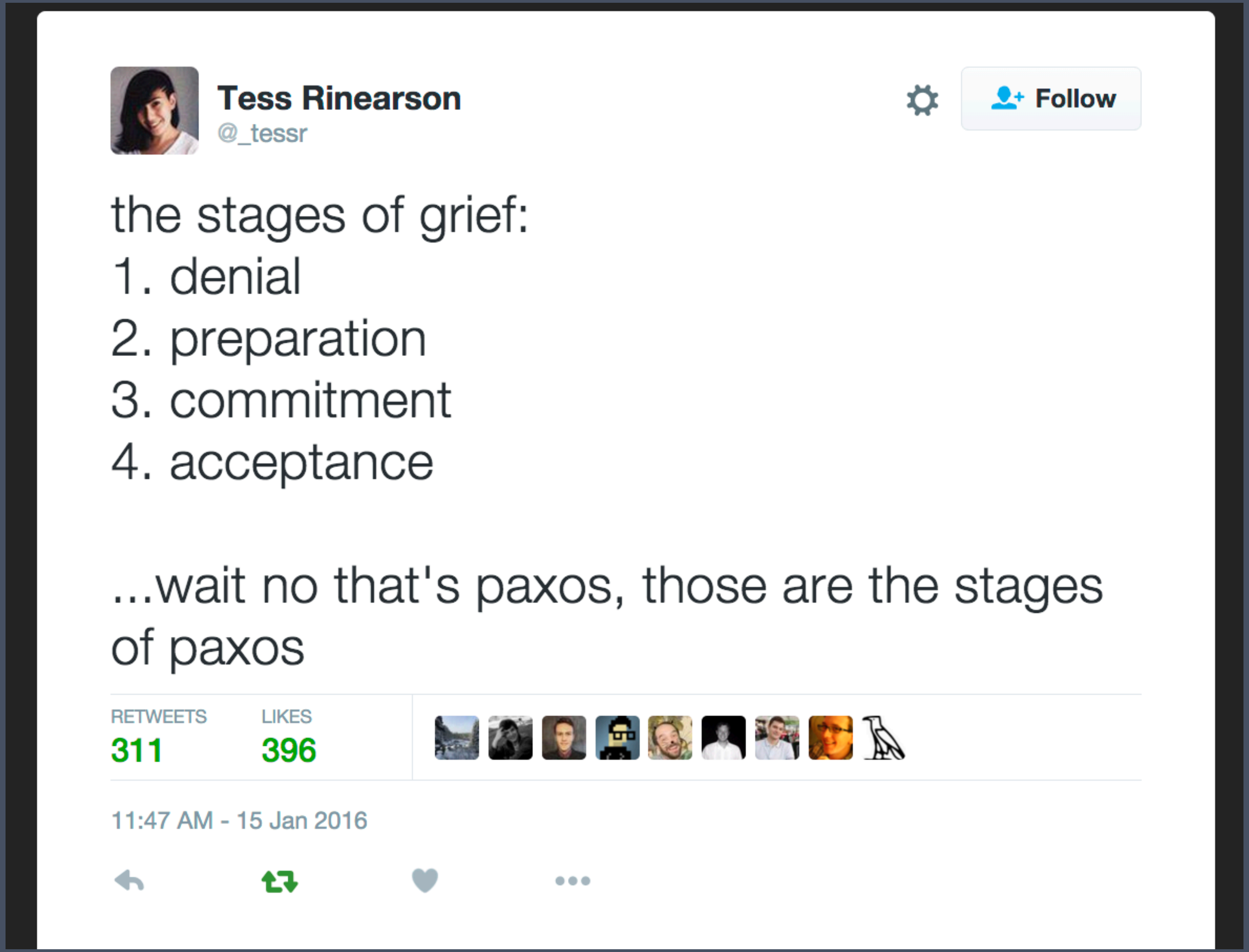
- STATE IS PRESENT IN THE CLUSTER, SO IF CM FAILS CAN RECOVER IT FROM ACTIVE REPLICAS.
- INDIVIDUAL MACHINES CACHE MAPPING AFTER FETCHING THROUGH RDMA

TRANSACTIONAL PROTOCOL

OPTIMISTIC CONCURRENCY:

TRANSACTIONS MAY FAIL AFTER LOCK ACQUISITION

COMMIT PROTOCOL



the stages of grief:
1. denial
2. preparation
3. commitment
4. acceptance

...wait no that's paxos, those are the stages of paxos

COMMIT PROTOCOL

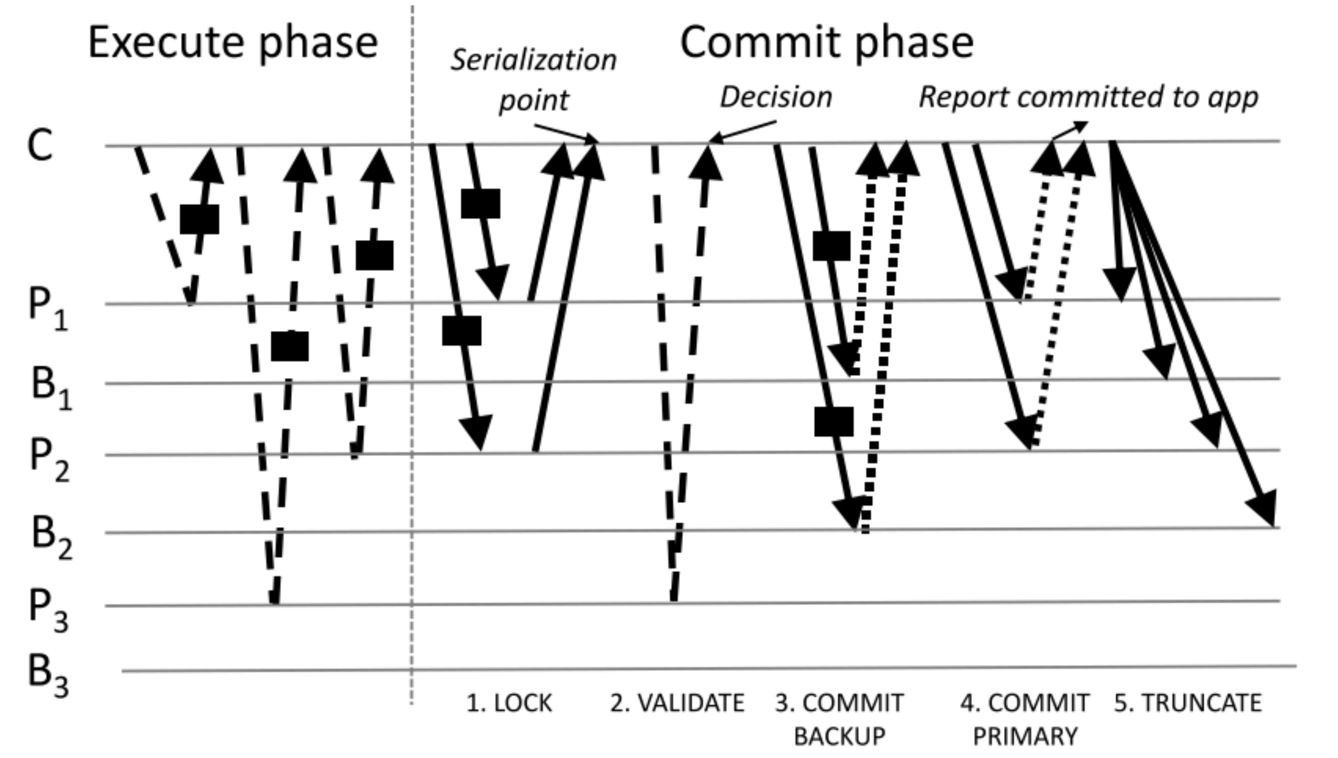


Figure 4. FaRM commit protocol with a coordinator C, primaries on P_1, P_2, P_3, and backups on B_1, B_2, B_3. P_1 and P_2 are read and written. P_3 is only read. We use dashed lines for RDMA reads, solid ones for RDMA writes, dotted ones for hardware acks, and rectangles for object data.

COMMIT PROTOCOL NOTES

- ALL COMMUNICATION IS OVER RDMA
- TOTAL MESSAGE DELAYS NOT FEWER THAN PAXOS
- BUT TOTAL NUMBER OF MESSAGES IS: 4P(2F + 1) VS PW(F+3) + PR
- AND SOME OF THOSE ARE EXTREMELY CHEAP

*

FAILURE DETECTION AND RECOVERY

LEASES

- I.E. REGISTRATION + KEEPALIVE, CREATED BY THREE-WAY-HANDSHAKE
- 5MS LEASES FOR 90-NODE CLUSTER, WITH 1MS-FREQUENCY RETRIES!!

LEASES - HOW THEY DID IT

- PREALLOCATION OF LEASE MANAGER MEMORY
- PIN CODE IN RAM
- KEEP HARDWARE THREADS FREE
- USE UNRELIABLE TRANSPORT

SEVEN-STEP PROCESS TOWARDS RECOVERY

1. SUSPECT - BLOCK EXTERNAL REQUESTS
2. PROBE - CHECK FOR CORRELATED FAILURES
3. UPDATE CONFIGURATION - ATOMICALLY MOVE CONFIGURATION TO NEXT VERSION IN ZK
4. REMAP REGIONS - RECOVER REPLICATION GUARANTEE FROM EXISTING REPLICAS

SEVEN-STEP PROCESS: COMMIT PROTOCOL

1. SEND NEW CONFIGURATION - REPLICAS ARE INFORMED OF NEW CONFIGURATION

2. APPLY NEW CONFIGURATION - REPLICAS UPDATE THEIR CONFIGURATIONS IN PARALLEL, AND WAIT...

3. COMMIT NEW CONFIGURATION - REPLICAS ARE TOLD TO START SERVING REQUESTS AGAIN

COMMIT PROTOCOL ENSURES CONSISTENT MEMBERSHIP STATE.

TRANSACTION RECOVERY

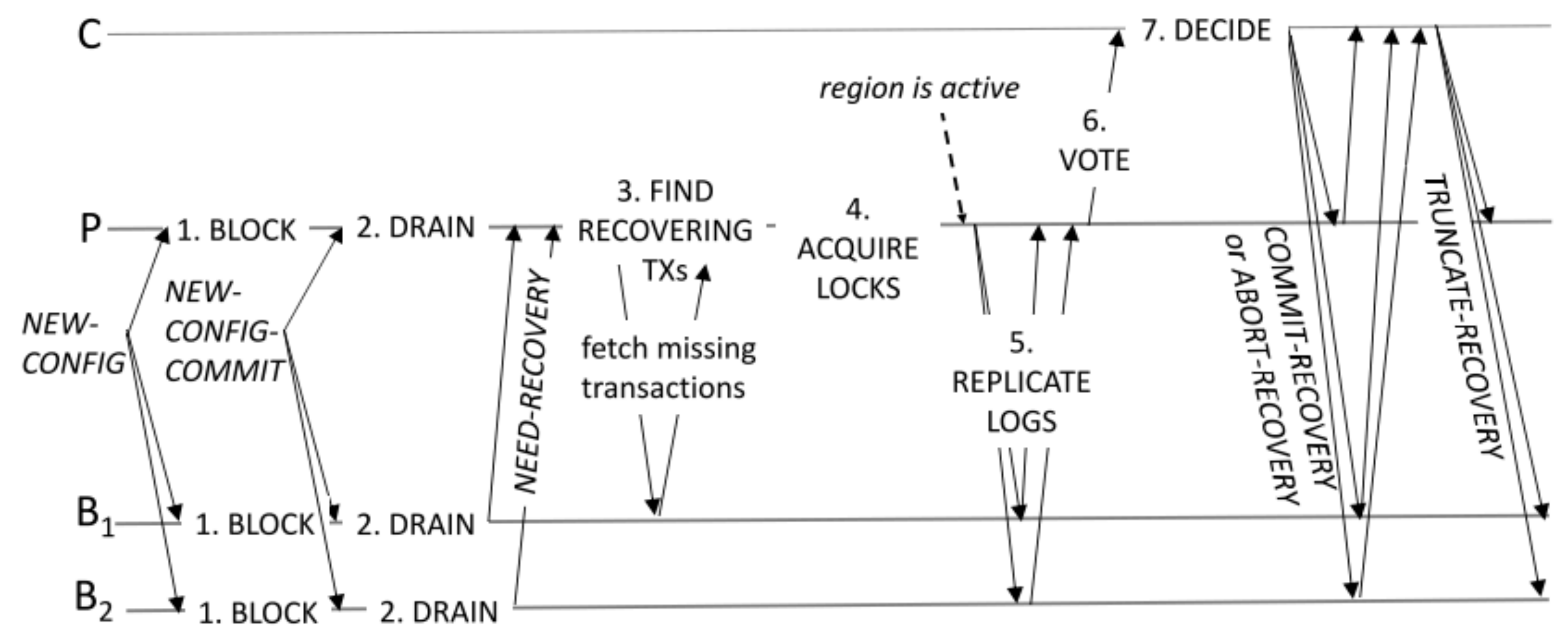


Figure 6. Transaction state recovery showing a coordinator C, primary P, and two backups B_1 and B_2

THANKS! QUESTIONS?

@HENRYR / HENRY.ROBINSON@GMAIL.COM