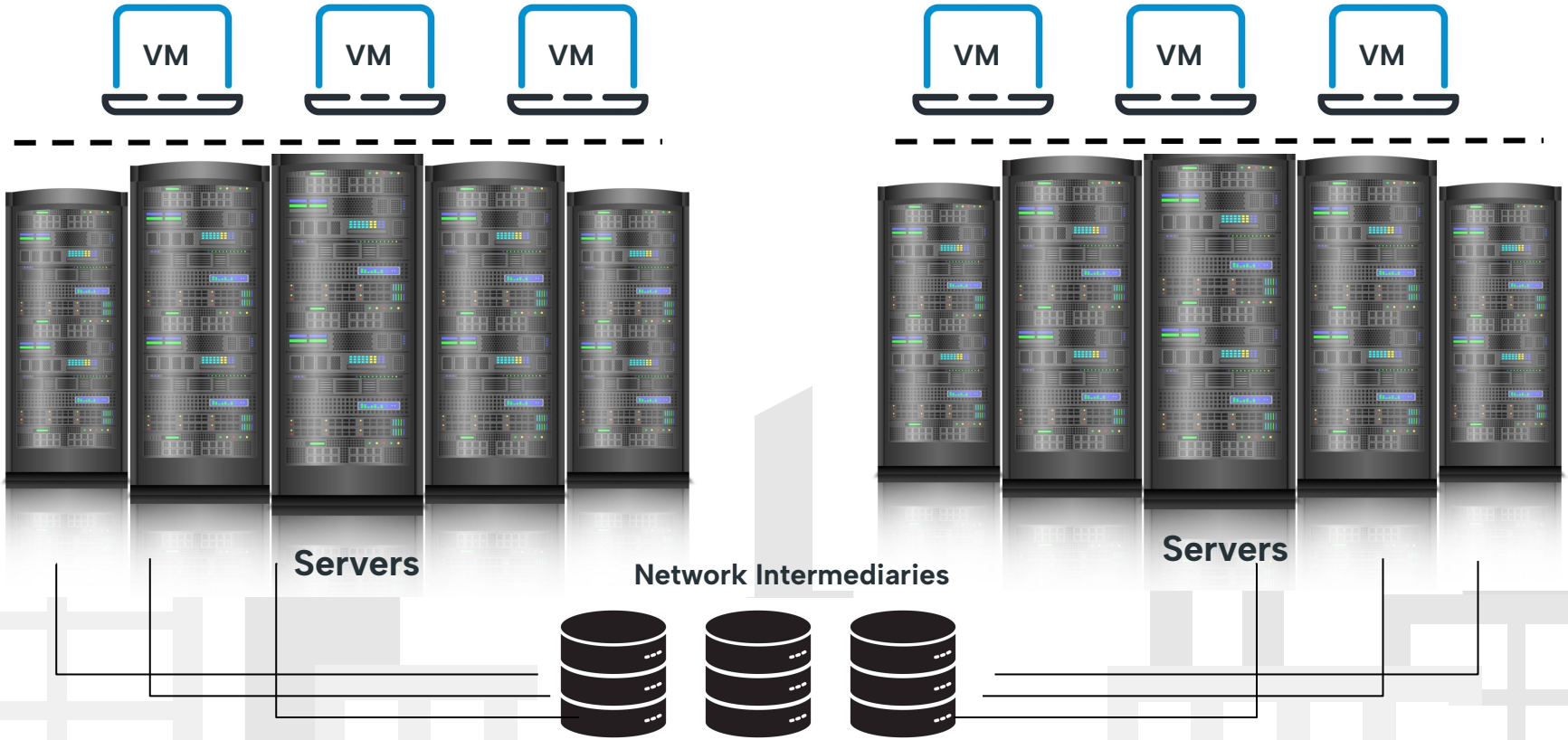
Enhancing System Checkpoints for Seamless Fault-Tolerant Live VM Migration

Name : **Pasindu Fernando**
Index : **20000512**
Supervisor : **Dr. Dinuni K. Fernando**

01 Introduction & Background

Live VM Migration



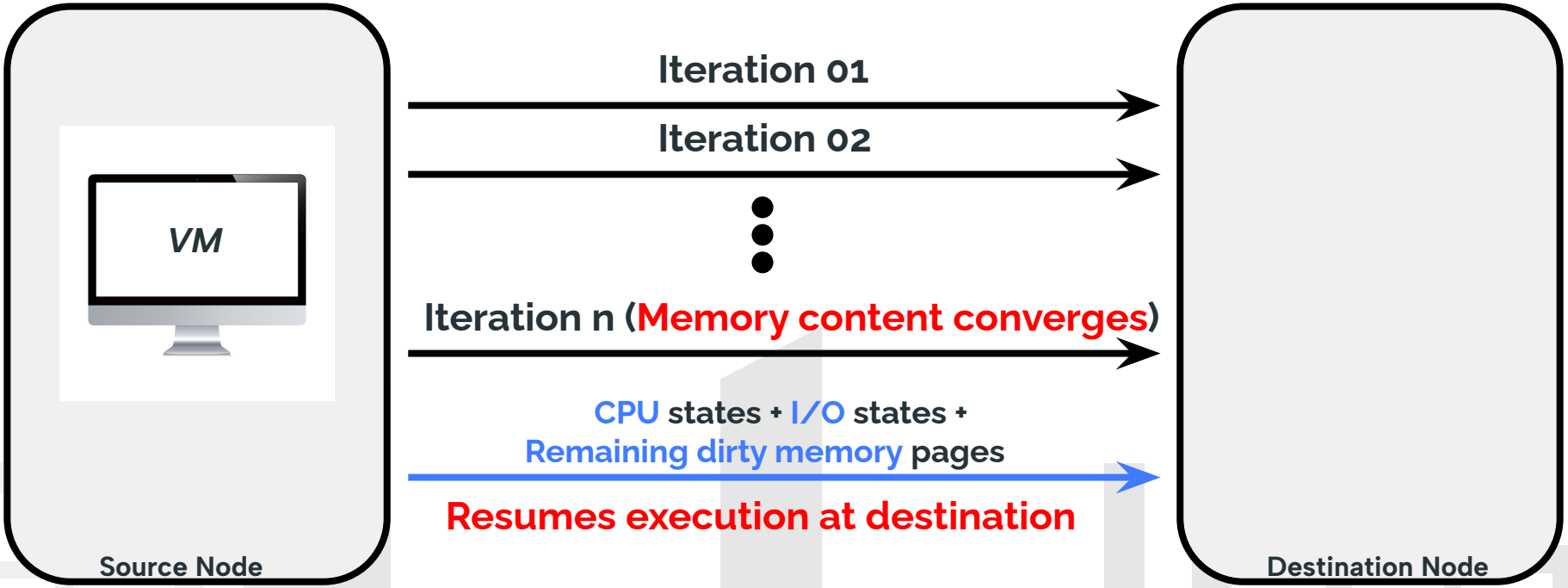
Live VM Migration Techniques

Major Live Migration techniques

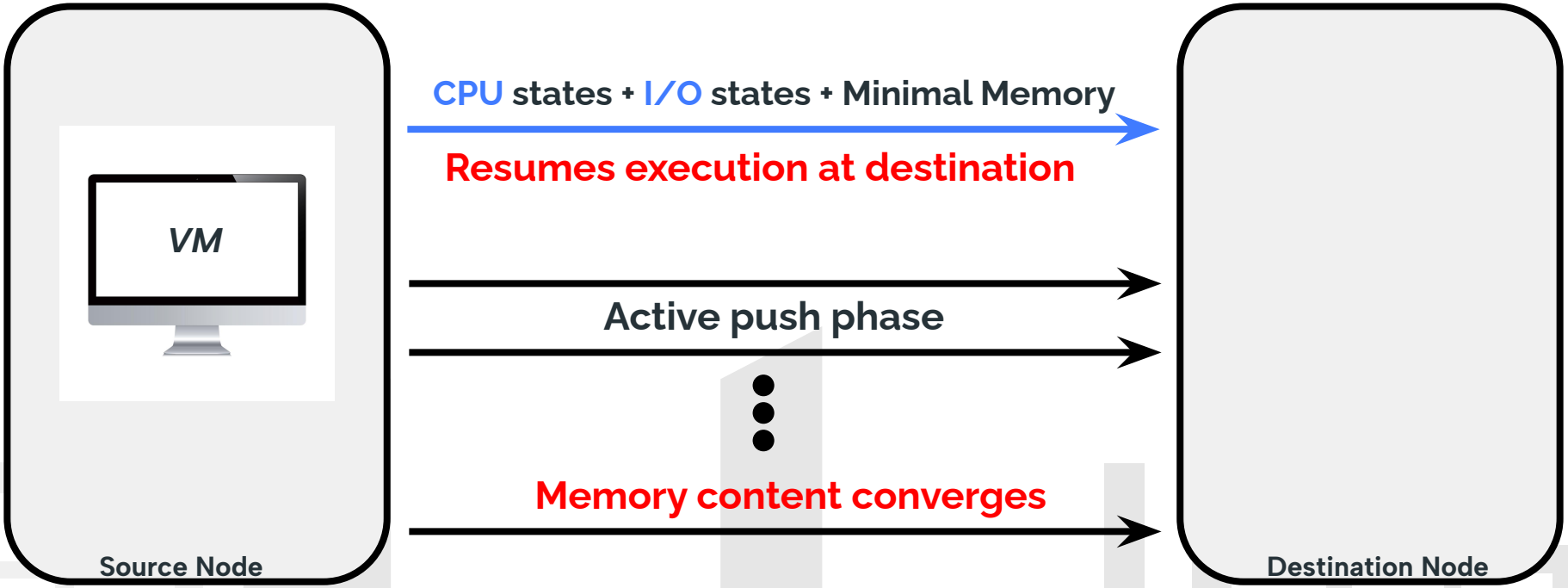
1. Pre-Copy migration
2. Post-Copy migration
3. Hybrid migration

The techniques differ in the order of the Memory, CPU and I/O states are transferred between the hosts.

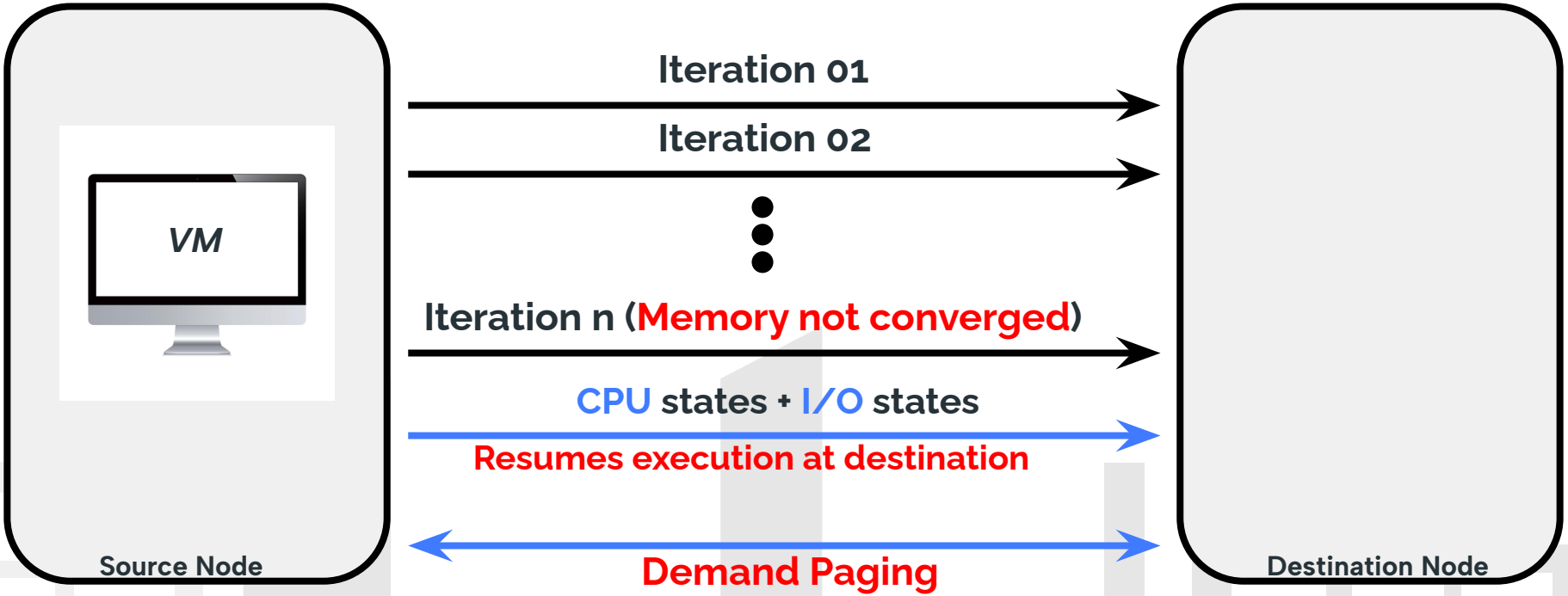
Pre-Copy Migration



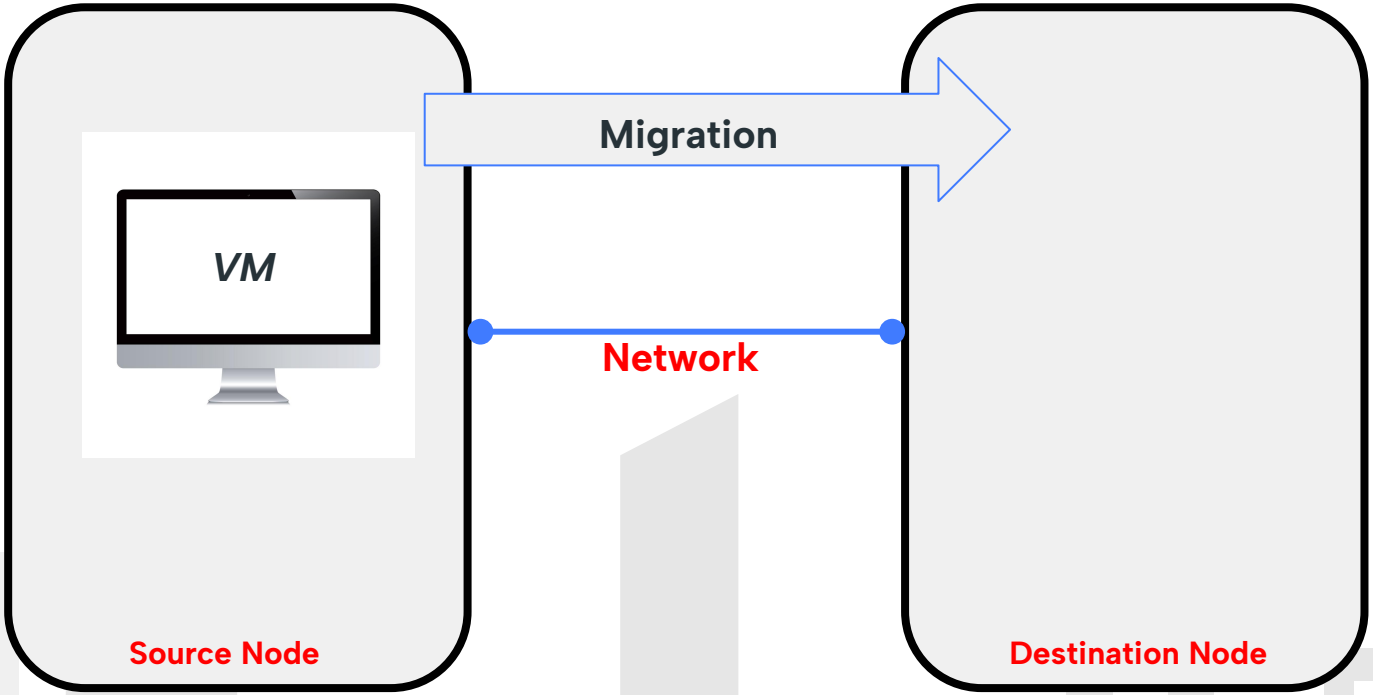
Post-Copy Migration



Hybrid Migration



What can go wrong during migration ?



What can go wrong during migration ?

Need of Checkpoints

In the instance of a failure which can result in losing the VM, checkpoints come into play.

- Versioned entities
- Contains all the necessities to recover a VM to a previous working state
- Used during VM failure

Problems of Checkpointing

- Checkpointing related operations like
  - *Capturing VM memory, CPU, I/O content*
  - *Versioning*
  - *Buffering outgoing packets*
  - *Transmitting them* via networks etc.

- These operations will be using the same computational resources dedicated for the usage by VMs running in the nodes.

- Results in producing several overheads on VM migration.

Checkpointing related overheads

- Migration-time overhead
- Total downtime overhead
- Application performance degradation of running VM
- Checkpoint replication time overhead
- Network bandwidth overhead
- Operational cost incurred

02 Motivation

Cloud Expansion - Computing shifting towards cloud solutions: SaaS, IaaS, PaaS

Resource Optimization - Need to optimize VM operations and resource usage in Cloud Data Centers (CDCs)

Checkpoint Challenges - Larger VMs lead to bigger checkpoints, making replication difficult. Reducing checkpoint sizes is crucial

Performance overheads - Improving checkpointing can mitigate the overheads incurred on VM migration

Impact and Value

- Enhancements can boost CDC productivity and cost-efficiency
- Significant contribution to the growing cloud computing industry

03 Related work

Fine-grained high-frequency checkpoints

(Cully et al, 5th USENIX symposium on networked systems design and implementation', San Francisco (2008))

- OS image level checkpoints
- Asynchronously replicating them in the background to a failover node
- Needed high computation power and efficient architectures

Log based checkpointing

(Liu et al, 18th ACM international symposium on High performance distributed computing (2009))

- Logs recorded at the source are replayed at the destination with synchronization algorithms
- Reduces the content to be transmitted
- But strict assumptions such as,
  - Log transfer and log replay rates > log accumulation rate
  - Limited only to single processor environments

Multi-level checkpointing

(Gelenbe et al, 2nd international conference on Software engineering (1976))

- Several checkpoints taken at once with varying costs and resiliencies
- Stored at different locations like RAM, disks and parallel file systems

Scalable Checkpoint Restart Library

(Moody et al, 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (2010))

- Made multi level checkpointing implementable
- Supported checkpointing caching and updating the cache
- Checkpoints are transferred to external mediums from cache only when some period elapses

AUFS infrastructure

(Guitart & Torres et. al, IEEE Network Operations and Management Symposium (2010))

- Separating read only portion of checkpoint from read write portion
- Only the read write portions need to be sent again

Node local NVM for checkpoint storage

(Kannan et al, IEEE 27th International Symposium on Parallel and Distributed Processing (2013))

- Using non-volatile memory as virtualized storage units for checkpoints

Recycling checkpoints

(Knauth & Fetzer et. al, 16th Annual Middleware Conference (2015))

- Identified the VMs are migrated between nodes according to a pattern
- Can recycle the older checkpoints of the VM in a new migration
- Only a small amount of data needs to be sent to construct the new checkpoint from the recycled one

Checkpoint corruption

(Wang et al, IEEE Transactions on Dependable and Secure Computing (2014) and Li et al, IEEE 26th International Symposium on Software Reliability Engineering (2015))

- Dual checkpointing schema
- Checkpoint period > Detection latency of corruption

Highly related work

Forward Incremental checkpointing (FIC)

(Fernando et al, 9th International Conference on Utility and Cloud Computing (2016))

- Reducing eviction time
- Incremental checkpoints taken in a way that
  - All memory pages (entire pages) transmitted in first iteration
  - In the successive iterations, only the modified pages(entire pages) since the last iteration are transmitted

- During eviction time, only the remaining memory since the last checkpoint is transmitted

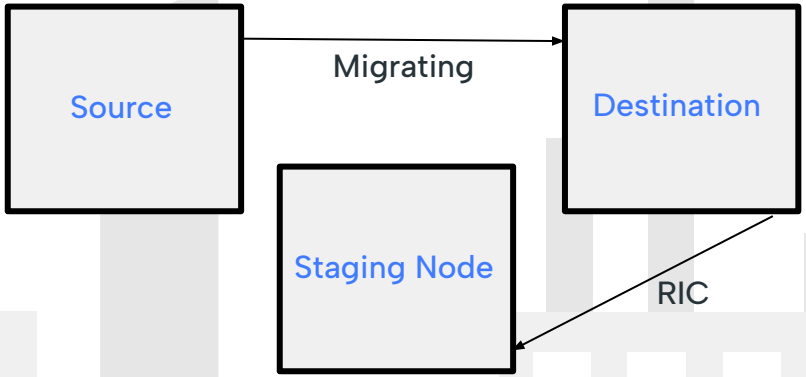
- Here checkpoints are not complete VM images, but portions of VM memory that will contribute to have a consistent memory checkpoint at the destination

Highly related work

Reverse Incremental checkpointing (RIC)

(Fernando et al,IEEE INFOCOM 2019-IEEE Conference on Computer Communications (2019))

- **In post-copy migration, execution states are resumed at the destination in the beginning.**
- **In a network or destination failure, the VM becomes unrecoverable.**
- **Uses the same incremental checkpointing concept, but in reverse.**
- **As soon as the execution is resumed at the destination, incremental checkpoints of the destination are taken and stored.**

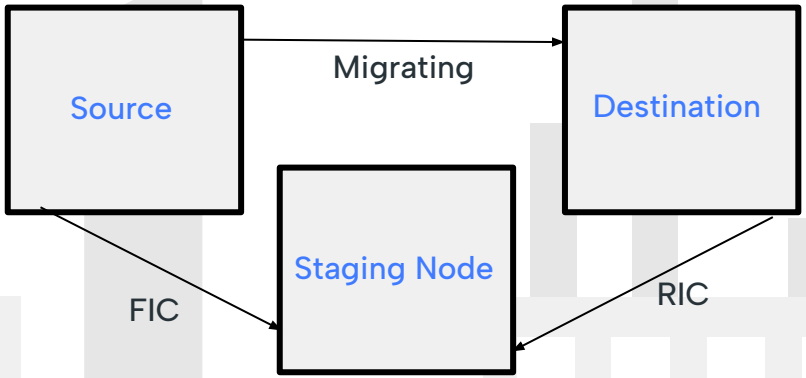


Highly related work

Combination of Forward and Reverse Incremental Checkpointing

(Fernando et al, IEEE Transactions on Cloud Computing(2023))

- **Forward incremental checkpointing: Handles source node failures**
- **Reverse incremental checkpointing : Handles destination node failures**
- **All points of failures are covered**
- ***Have not considered further reducing checkpointing data before transmitting via networks***



Highly related work

Checkpoint compression

Gzip Compression (Sardashti & Wood et al, ACM TACO 2017)

- Based on Lempel_Ziv (LZ77) coding and Huffman coding.
- Focuses on achieving high compression ratios.
- Ideal with data with repetitive patterns.

LZ4 Compression (Bartik et al, IEEE ICECS 2015)

- Focuses on low latency/high compression speed..
- Suitable for real time data processing.

Zstd Compression (Collet et al, IETF 2021)

- Developed by Facebook.
- Uses a combined effort of dictionary based compression and Huffman coding.

Page delta tracking

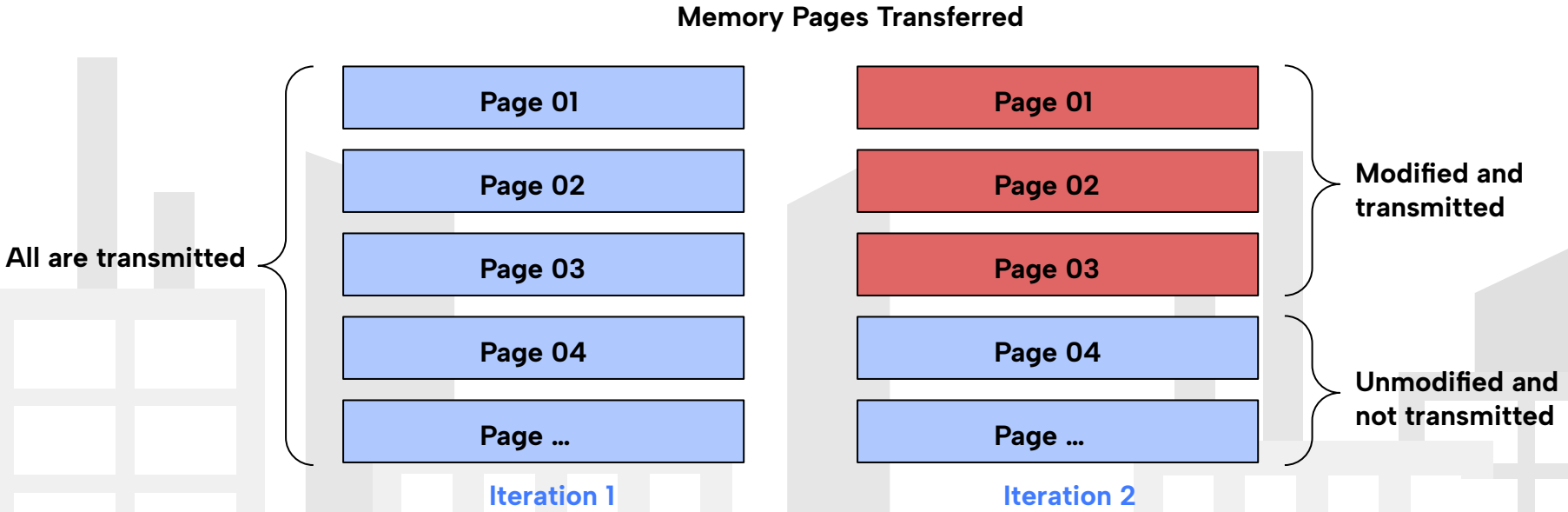
(Bhardwaj and Rama Krishna et al, Proceedings of 2nd ICCCN 2018)

- Explored the possibility of tracking the modified content within a page.
- Combined effort of XOR and run length encoding.

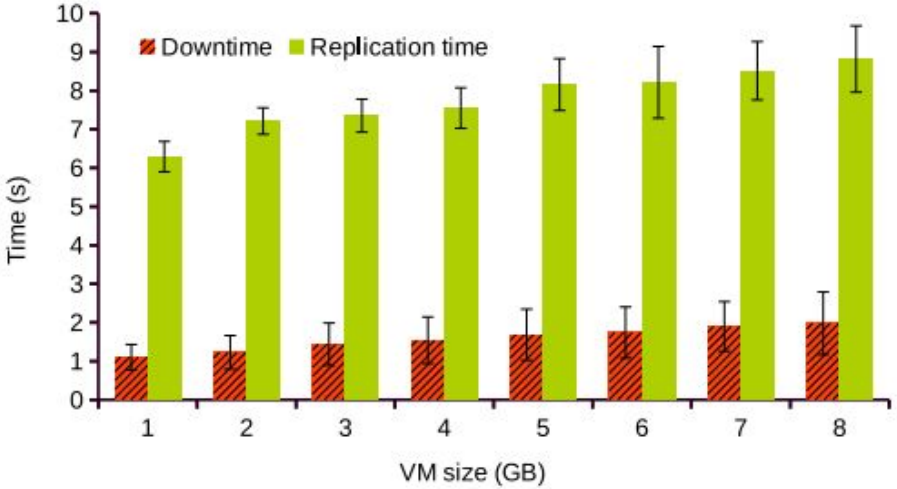
04 Research Gap

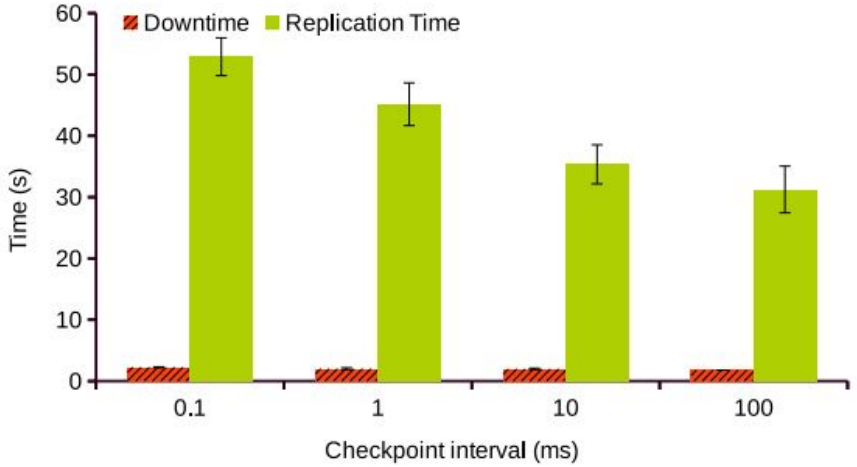
In the context of incremental checkpointing, so far the memory pages are transmitted in a way,

1. First, all the memory pages available are transmitted.
2. In successive iterative increments, only the modified pages are transmitted.



Why do we actually need incremental checkpointing ?



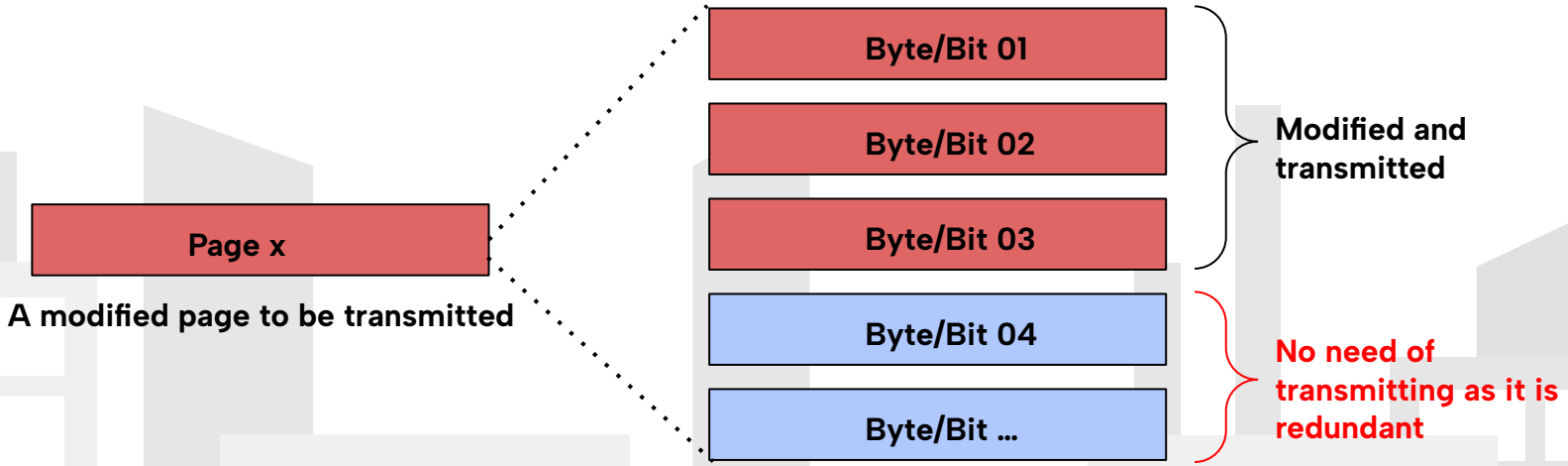


Fernando et al,IEEE INFOCOM 2019-IEEE Conference on Computer Communications (2019)

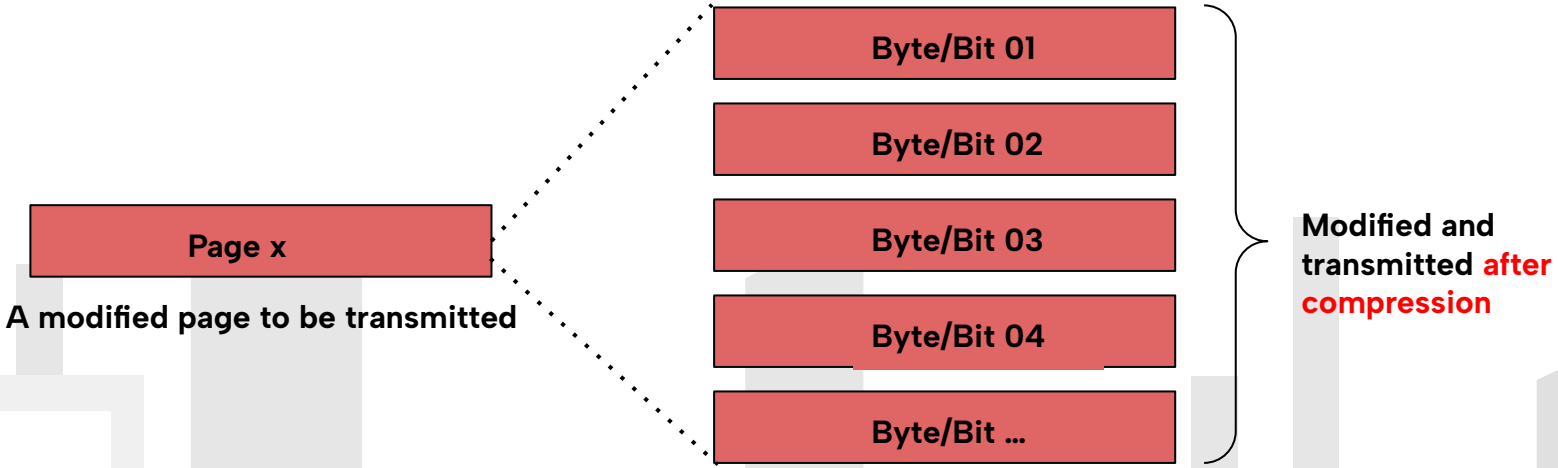
- In VM migration, the time to replicate the checkpoint to the desired location is really costly.
- The checkpointing data should be made smaller as possible to mitigate this issue

This drastically reduces the checkpointing data to be transmitted reducing the overheads incurred on VM migration

However, although the whole modified page is transmitted, the modifications can be limited to several bits/bytes out of a page of approximately 4 kilobytes



Additionally in incremental checkpointing, the literature hasn't explored the possibility of compressing the checkpointing data before sending them via the network



How to improve incremental checkpointing?

- Rather than tracking the modified pages and sending them, the pages can be examined deeper to identify the modified bytes/bits and only consider them in the checkpoints.
- The checkpoint data can be further compressed with checkpoint compression techniques to further reduce checkpoint data overheads.

What can we expect ?

- Will reduce the checkpointing data to be sent by mitigating redundant data transfer.
- Will reduce checkpoint replication time and other overheads.
- Ultimately, it will result in a lesser faulty window.

05 Research Questions

Research Questions

01

How can the incremental checkpoints be further improved to reduce overheads incurred on VM migration ?

02

How can the state of the art compression techniques be incorporated in reducing checkpoint size during state transfers ?

06 Objectives

Research Objectives

01

Design and develop a strategy to make the checkpoint data transmitted more granular for the purpose of reducing the overheads associated with current checkpointing operations.

02

Analyze the improvements in the overheads incurred, introduced by the new granular methodology with respect to the existing methodologies.

03

Integrate checkpoint compression techniques and analyze the improvements made in terms of the overheads introduced

07 Scope

In Scope

- Baseline model reproduction- The baseline incremental checkpointing model in Fernando et al. (2019) will be reproduced in QEMU/KVM.
- Byte/Bit-level granular model implementation- A byte/bit level fine granular checkpointing method will be introduced that goes beyond page level modification tracking and transfer in checkpointing iterations.
- Implementation of a working prototype- A working prototype will be implemented with the granular byte/bit-level mechanism.
- Integration of checkpoint compression techniques- State of the art checkpoint compression techniques will be integrated to Fernando et al. (2019) reduce the checkpoint data transfer overheads.
- Evaluation- The results and overhead reductions will be evaluated with approved realistic industrial benchmarks.

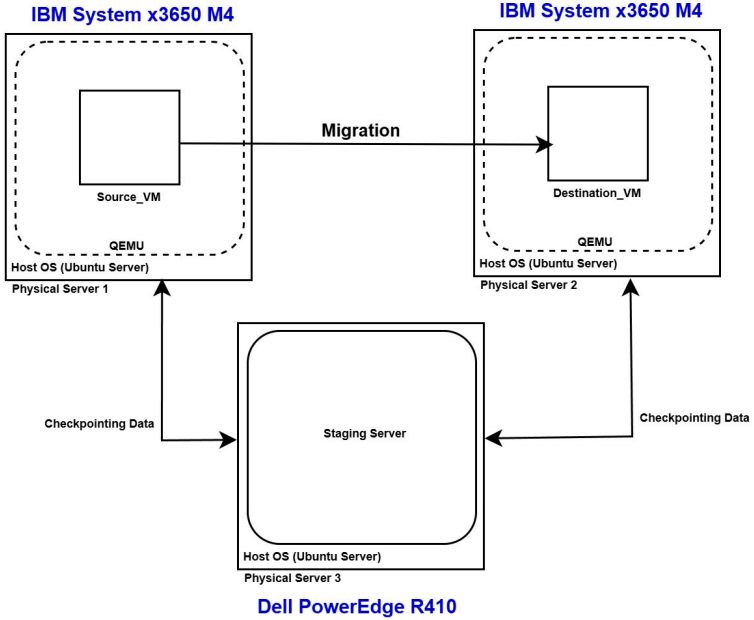
In Scope

- **Scope limited to KVM-QEMU hypervisor - All implementations will be restricted to the KVM-QEMU hypervisor environment**
- **Single VM failure scenarios- The focus will be on handling single VM failures, with the potential for future extension to multiple VM failure scenarios.**
- **Linux-based Ubuntu OS- The implementations will be developed and tested exclusively on Linux-based Ubuntu operating systems.**

08 Research Approach

Research approach

Testbed setup - Two servers + One staging server for the checkpoints connected via Gigabit ethernet with QEMU/KVM as the hypervisor



- **Host Servers**
  - **CPU -** Intel Xeon E5-2697 v2 (48) @ 3.500GHz
  - **RAM** - 314GB
- **Staging Server**
  - **CPU -** Intel Xeon E5503 (4) @ 1.596GHz
  - **RAM** - 8GB
- **Connections**
  - **1000Mbps** Gigabit ethernet

Research approach

Implement Baseline Model - Page level incremental checkpointing utilized in Fernando et al. (2019) will be reproduced in QEMU/KVM.

Introduce Byte/Bit-level Granularity - Byte/Bit-level granularity will be introduced to the implemented baseline model deviating deeper from page level granularity.

Evaluate overhead improvements- Evaluate the overhead reductions with the byte-level granularity introduced compared to page-level Fernando et al. (2019) findings.

Research approach

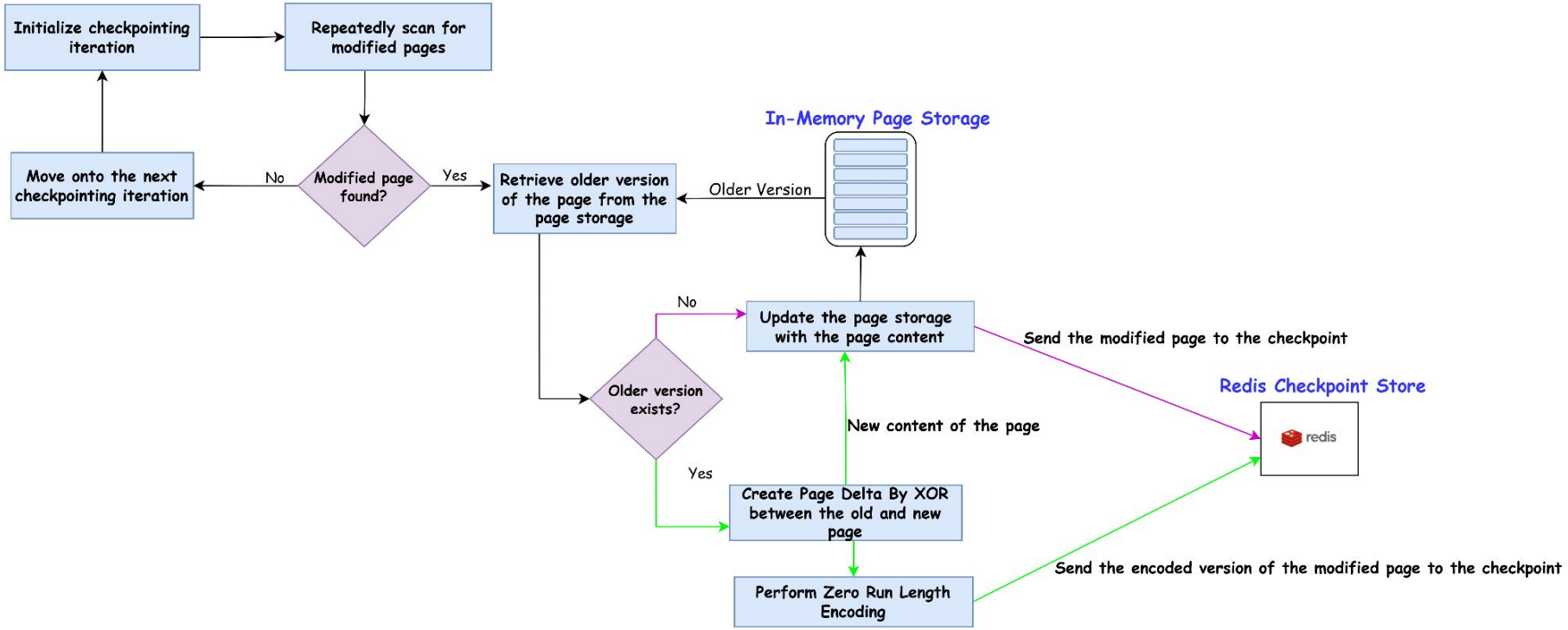
Integrate checkpoint compression techniques - Integrate state of the art checkpoint compression techniques to further reduce the checkpoint data transfer overheads

Evaluate findings with realistic benchmarks

- **Evaluate the findings, overhead reductions with different CPU, memory, network bandwidth utilization using realistic industrial and synthetic benchmarks like SysBench, working set , iPerf , Quicksort etc.**
- **Improvements by the byte/bit level granularity method and compression algorithms integration will be compared to existing page level mechanism.**

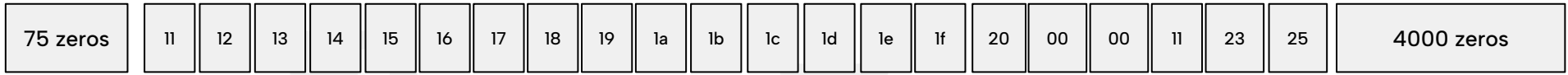
09 Research Design and Progress So Far

Byte-level granularity approach

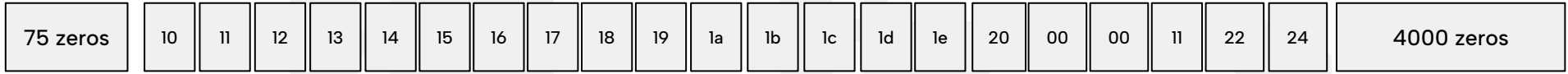


Byte-level granularity approach example

Old Buffer

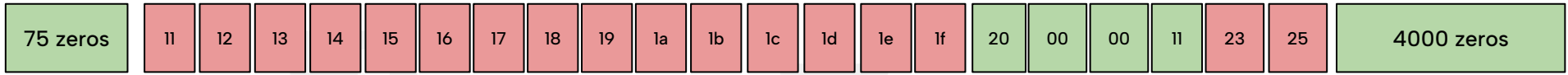


New Buffer

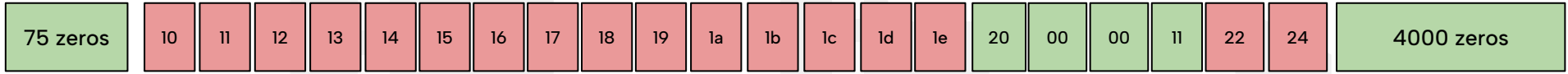


Byte-level granularity approach example

Old Buffer



New Buffer



Byte-level granularity approach example

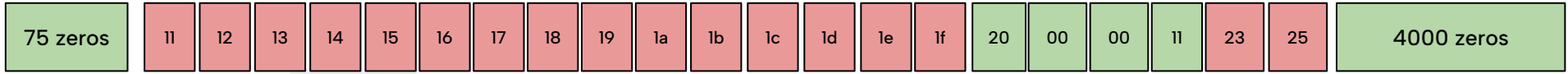
How is encoding taking place?

The page difference is encoded to a series of runs. Each run is either a

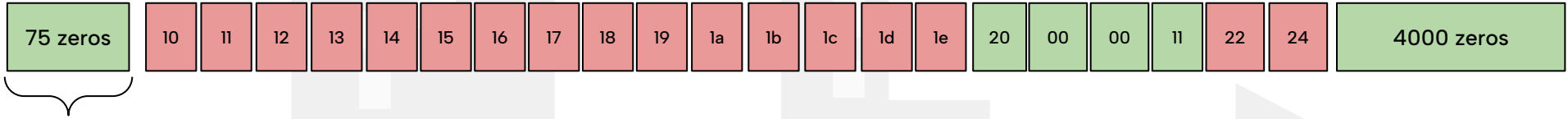
1. **Zero run**
   - Represents a portion of the page which was not modified/changed
   - Represented with the length of that portion (ULEB128 format)
2. **Non - Zero run**
   - Represents a portion of the page which was modified/changed
   - Represented by the length of that portion (ULEB128 format) followed by the content that was modified

Byte-level granularity approach example

Old Buffer



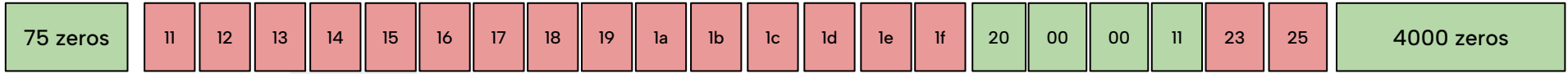
New Buffer



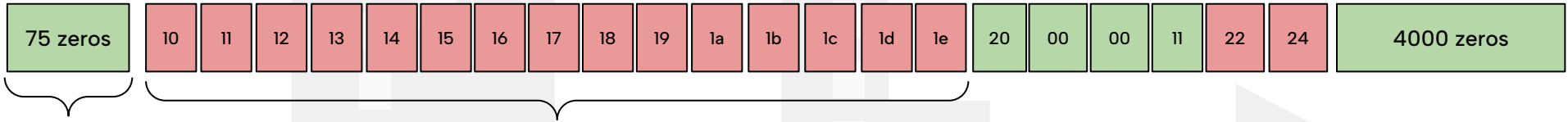
First 75 zeros are unchanged, represented as "4b" which means 75 in ULEB128 format

Byte-level granularity approach example

Old Buffer



New Buffer



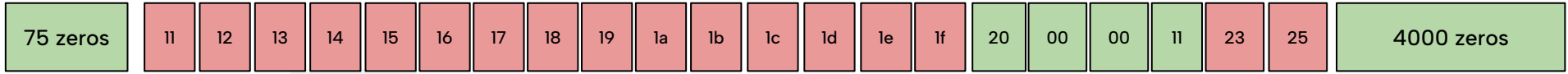
4b

Represents a non-zero run of 15 bytes. Therefore represented as "0f" which means 15 in ULEB128 format, followed by the 15 bytes which were modified

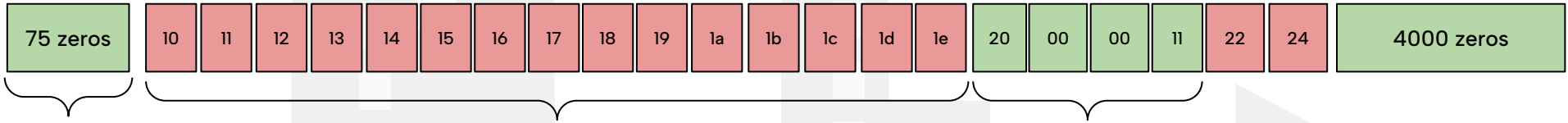
0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e

Byte-level granularity approach example

Old Buffer



New Buffer

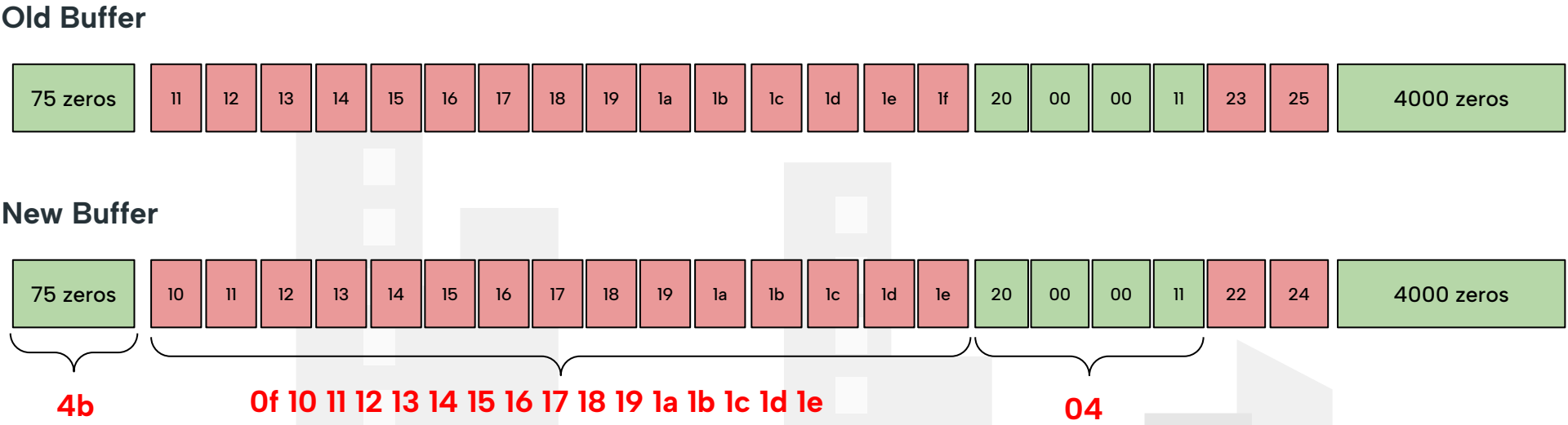


4b

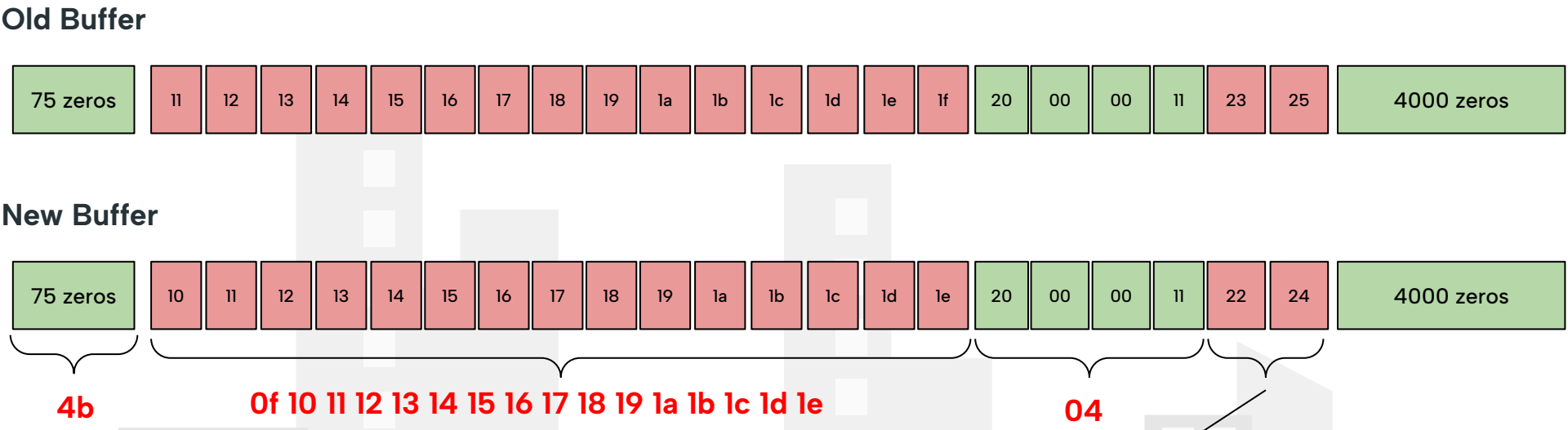
0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e

Represents a non modified portion/zero run.
Represented as “04” which means 4 in ULEB128 format

Byte-level granularity approach example

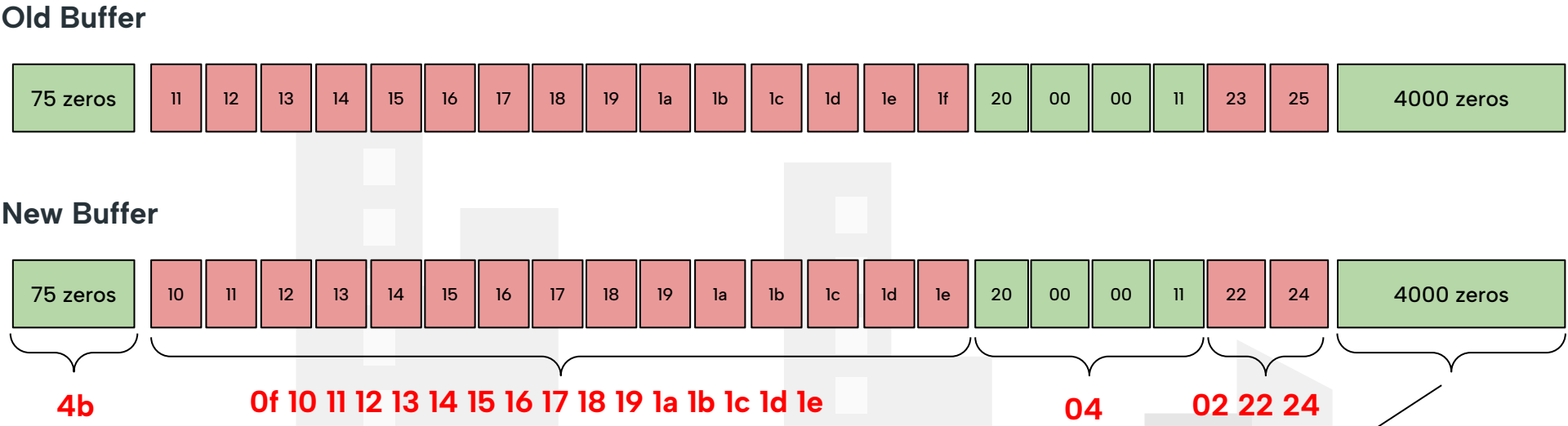


Byte-level granularity approach example



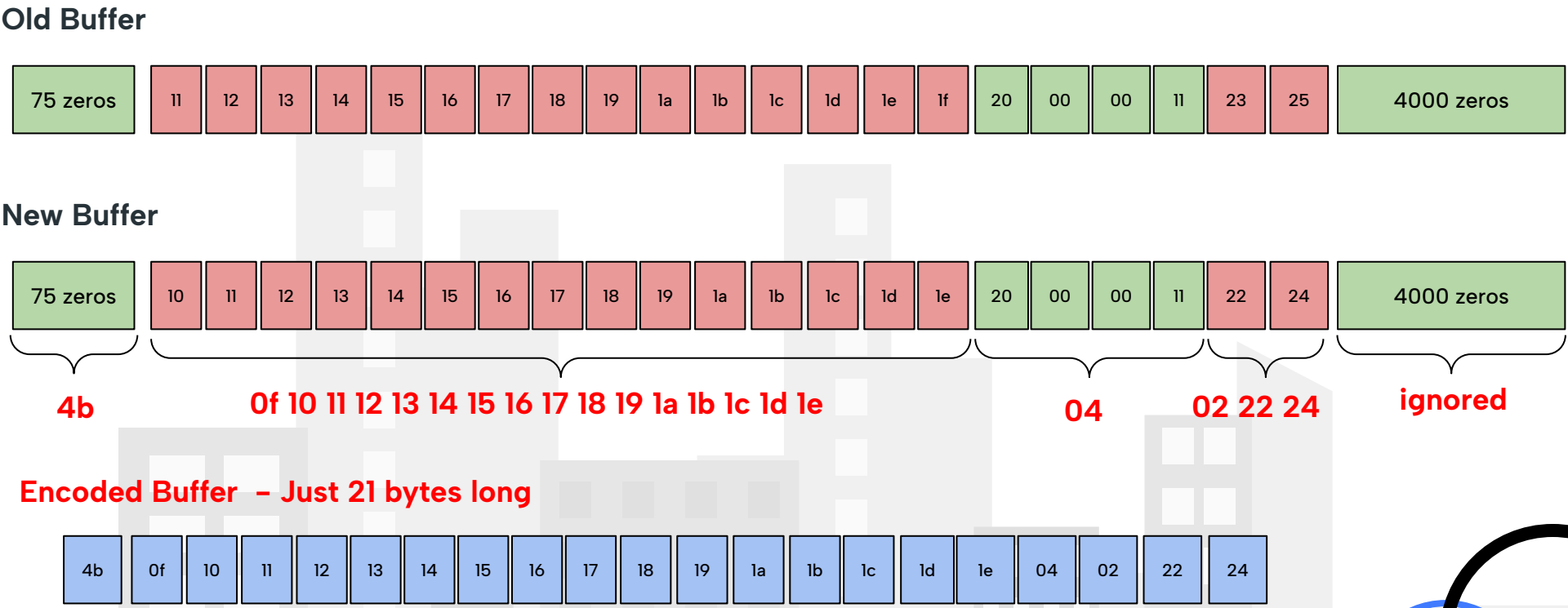
Represents a modified portion of length two.
Represented as “02” in ULEB128 format and followed by the two bytes modified

Byte-level granularity approach example



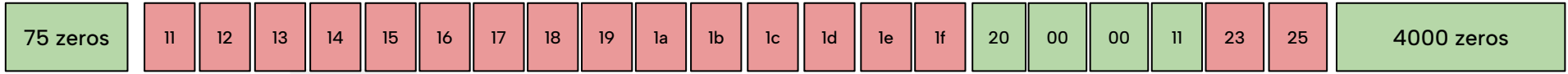
The last zero run is ignored meaning that the rest of the page is unchanged

Byte-level granularity approach example

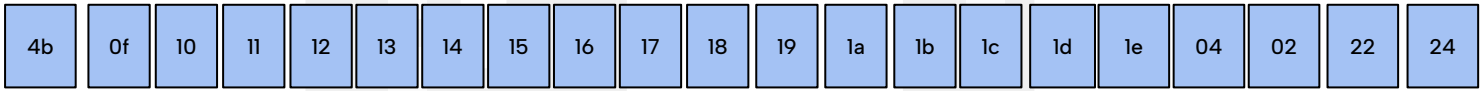


Byte-level granularity approach example - Decoding end

Old Buffer

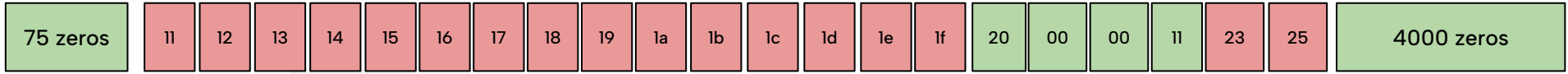


Encoded Buffer – Just 21 bytes long

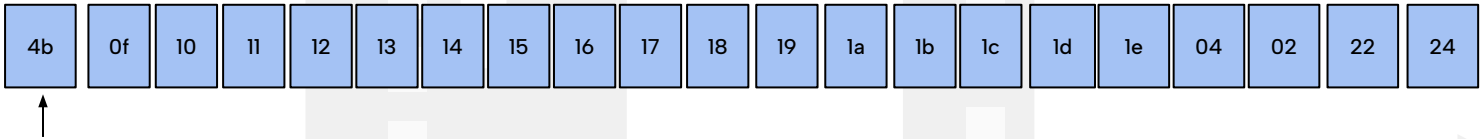


Byte-level granularity approach example - Decoding end

Old Buffer



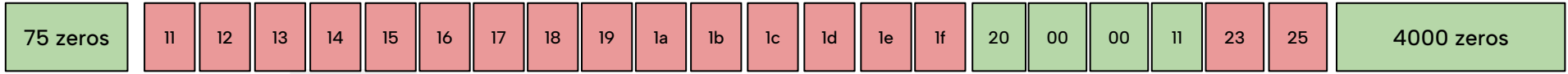
Encoded Buffer – Just 21 bytes long



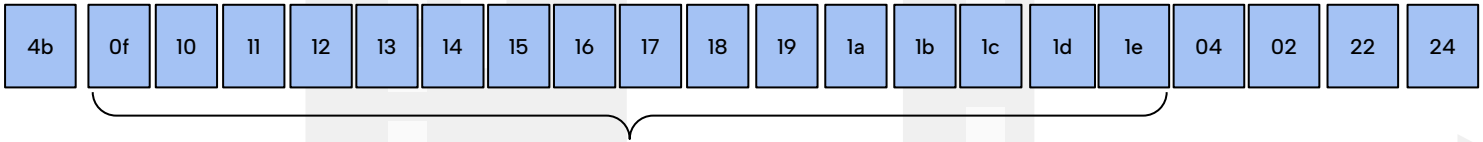
Represents a unmodified portion of length 75. Hence nothing needs to be done

Byte-level granularity approach example - Decoding end

Old Buffer



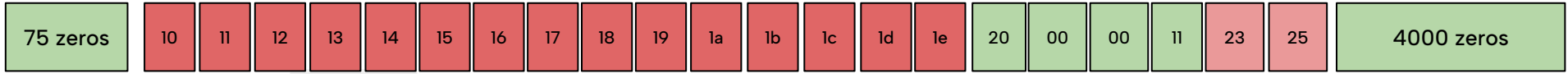
Encoded Buffer – Just 21 bytes long



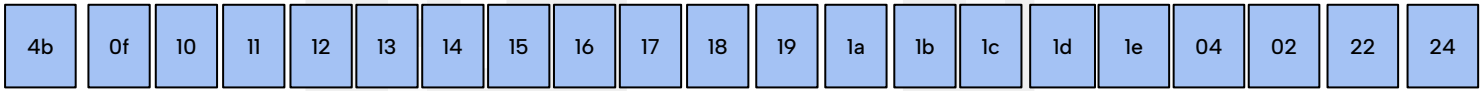
Represents a modified portion of length 15.
Therefore the older buffer is updated with the new values

Byte-level granularity approach example - Decoding end

Old Buffer



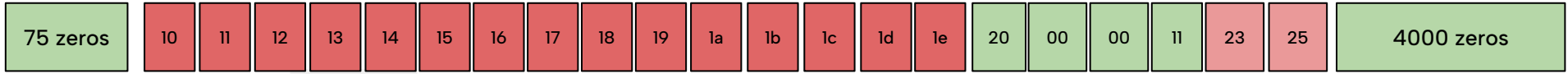
Encoded Buffer – Just 21 bytes long



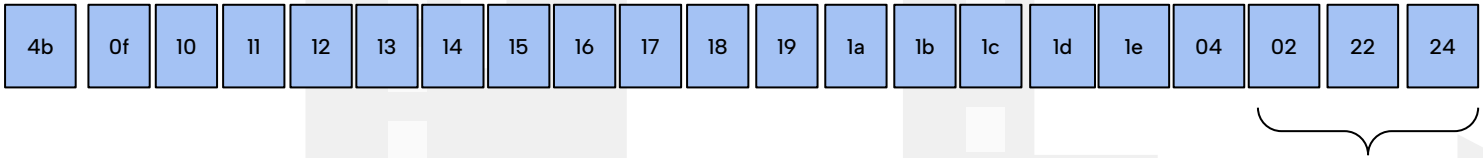
Represents a unmodified portion of length 04. Hence nothing needs to be done

Byte-level granularity approach example - Decoding end

Old Buffer



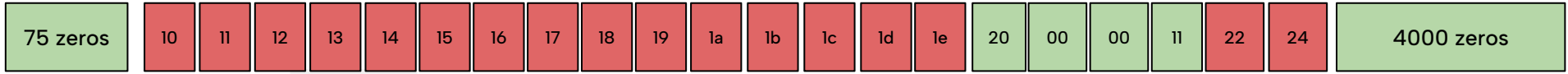
Encoded Buffer – Just 21 bytes long



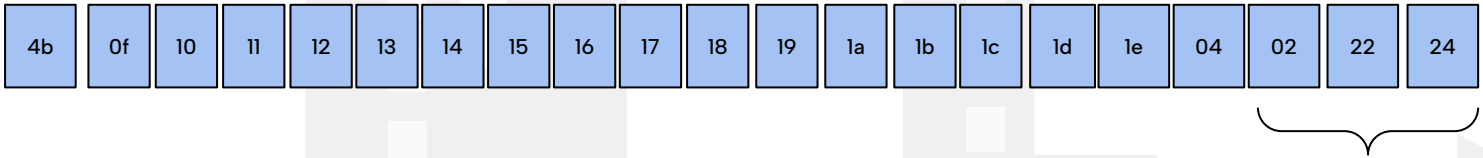
Represents a modified portion of length 02. Therefore the older buffer is updated with the new values

Byte-level granularity approach example - Decoding end

Old Buffer



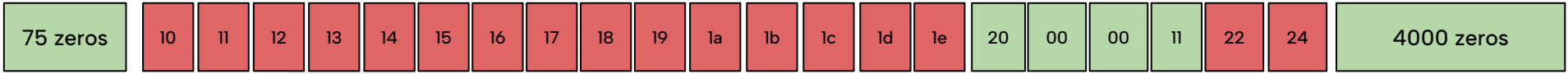
Encoded Buffer – Just 21 bytes long



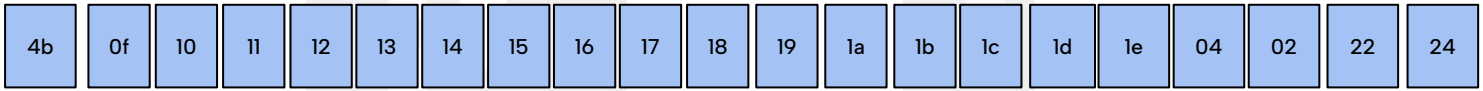
Represents a modified portion of length 02. Therefore the older buffer is updated with the new values

Byte-level granularity approach example - Decoding end

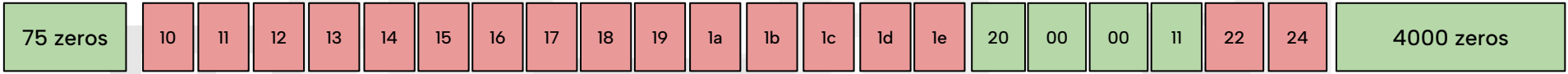
Old Buffer



Encoded Buffer – Just 21 bytes long



New Buffer – What we didn't send



Compression techniques integration

With compression techniques integrated, the flow happens in a format such that

Checkpointing

1. **Modified page is identified**
2. **Page is compressed using a compression algorithm and sent to the checkpoint store**

Recovery End

1. **Page is retrieved from the checkpoint store**
2. **Page is decompressed to the original size before loading.**

Compression techniques integration

Three lossless compression techniques were integrated to the model which are namely

1. **Gzip**
   - **Combined effort of Lz77 and Huffman coding**
   - **Focusses on having a higher compression ratio**

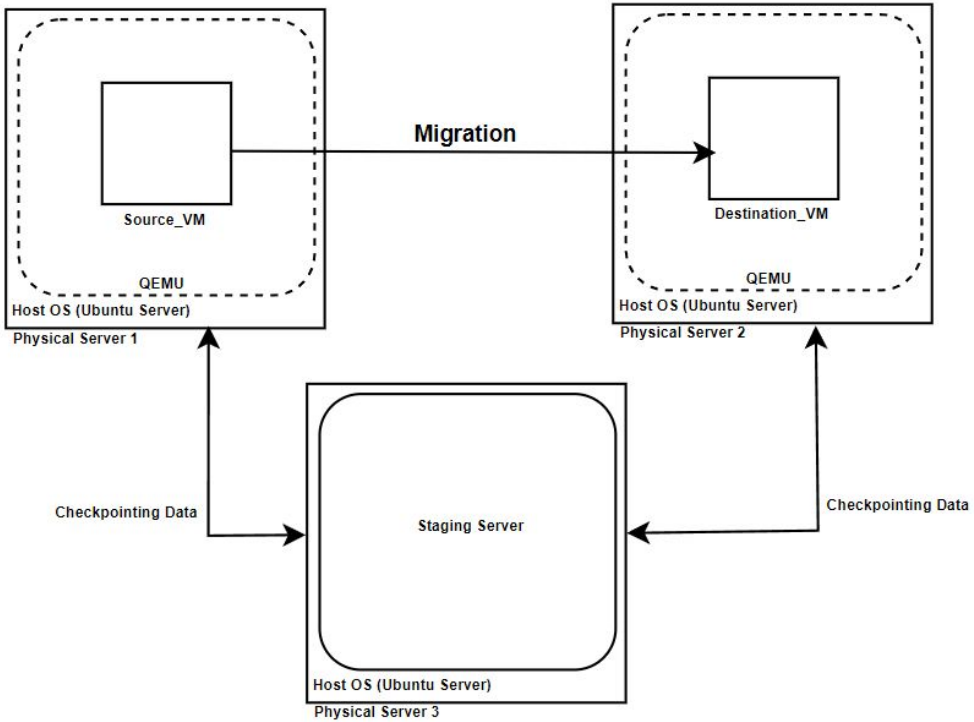
2. **LZ4**
   - **Focuses on being fast/ low latency**
   - **Applicable in real-time data processing applications**

3. **ZSTD**
   - **Developed by Facebook**
   - **Combined effort of dictionary based compression and Huffman coding**

10 Progress

Progress so far

- **Test-bed setup completed**



Progress so far

- Created and migrated VM's using pre-copy, post-copy and hybrid techniques.
- Incremental checkpointing baseline model (Fernando et al, 2019) was successfully reproduced in QEMU/KVM.
- The byte level granularity method was successfully designed and fully implemented on top of the baseline model.
- The compression techniques Gzip, Lz4 and Zstd were successfully integrated to the checkpointing flow in the baseline model.
- Experimented on the content reduction of the byte level granularity mechanism and the compression technique integration when compared to baseline model

Experiments - Byte Level Granularity Approach

- The proposed byte-level granularity mechanism was fully implemented on top of the baseline model Fernando et al, 2019.

- With the mechanism enabled,
  - Multiple migrations tested
  - Recovery of VM using checkpoints

- Successfully recovered the VM using the fine granular checkpoints.

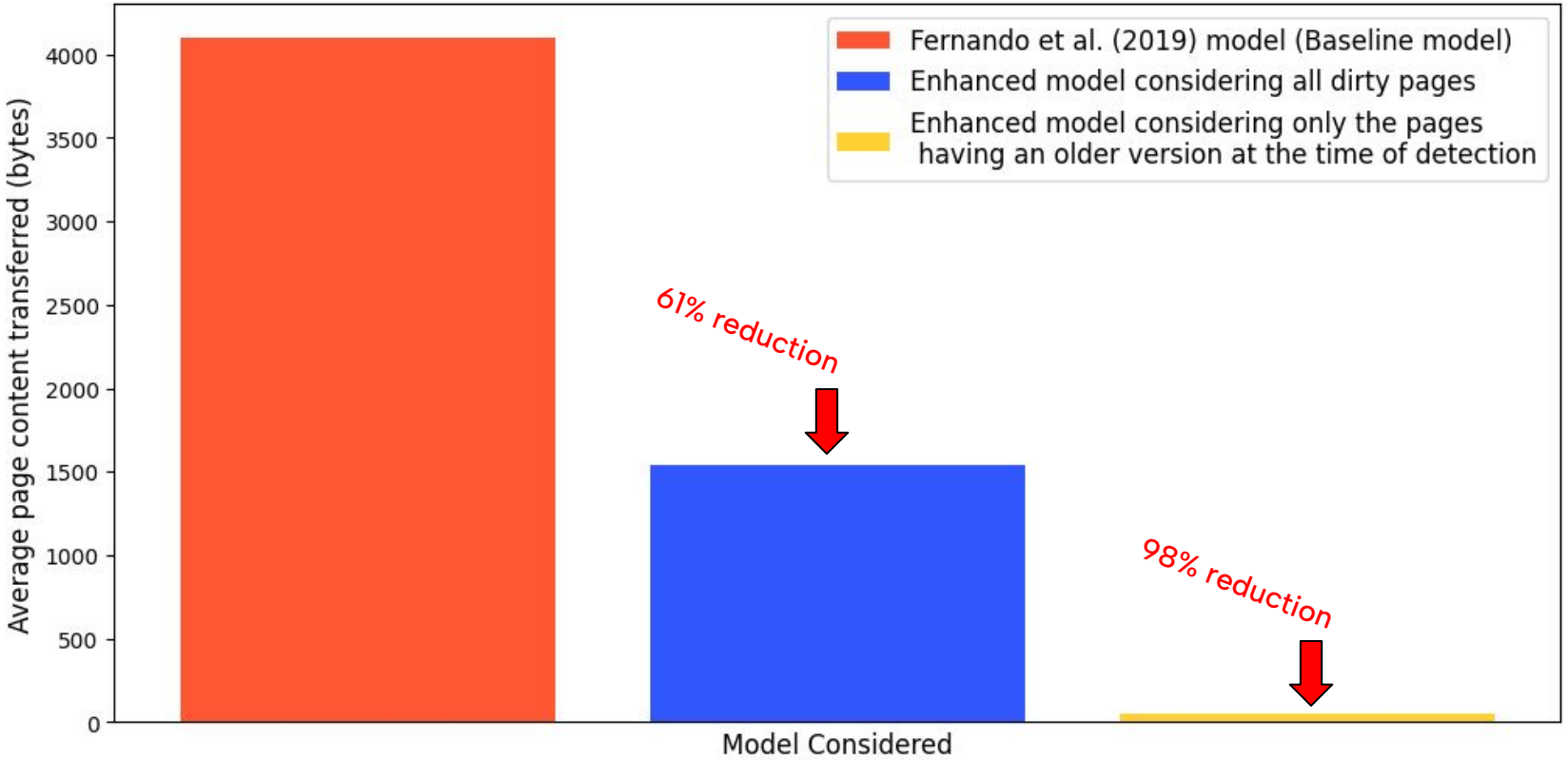
Experiments - Byte Level Granularity Approach

Page content overhead reduction experiment

- With the fine granular checkpointing enabled, the average content of the pages transmitted as checkpoints were observed.
- Compared with the baseline model Fernando et al. 2019.
- Tested with idle and 7000MB working set workloads
- Showed drastic decrements in the average page content transmitted

Experiments - Byte Level Granularity Approach

Page content overhead reduction experiment



Experiments - Byte Level Granularity Approach

Page content overhead reduction experiment

Workload nature	Idle	7000MB - working set
Average dirty pages found	6168	6480
Average page content(bytes) transmitted including pages without having an earlier older version. (All pages considered)	1563.17	1540.1
Percentage of content reduced (including pages without older version)	61.84(%)	62.40(%)
Average page content(bytes) considering only the pages having an older version	47.41	54.47
Percentage of content reduced (considering pages with older versions only)	98.84(%)	98.67(%)

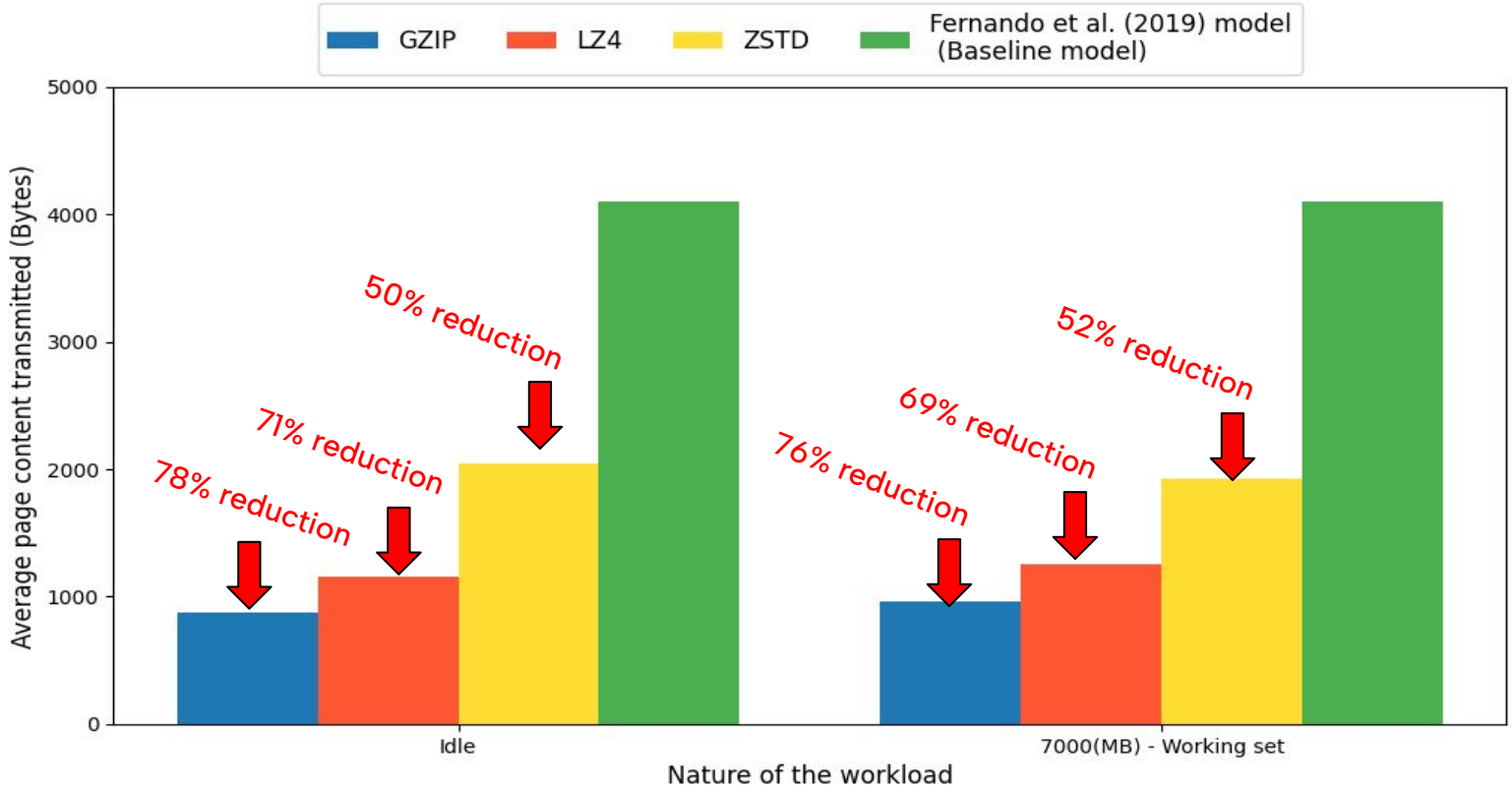
Experiments - Compression Integration

Page content overhead reduction experiment

- With the compression techniques enabled, the average content of the pages transmitted as checkpoints were observed.
- All three compression algorithms Gzip, Lz4, Zstd were tested.
- Compared with the baseline model Fernando et al. 2019.
- Tested with idle and 7000MB working set workloads
- Showed drastic decrements in the average page content transmitted

Experiments - Compression Integration

Page content overhead reduction experiment



Experiments - Compression Integration

Page content overhead reduction experiment

Compression technique	Idle	7000MB - working set
GZIP	877.47	957.49
LZ4	1159.11	1255.94
Zstd	2046.60	1926.20
Fernando et al, 2019 baseline model (No compression)	4096	4096

Compression technique	GZIP	LZ4	Zstd
Page Content Reduction(Idle)	78.57(%)	71.70(%)	50.03(%)
Page Content Reduction(7000 (MB)- workingset)	76.62(%)	69.33(%)	52.97(%)

11 Evaluation & Next Steps

Evaluation - Methodology Accuracy

With the Integration of byte-level granularity and compression techniques into the model from Fernando et al. (2019),

Process

- **VMs are migrated, recovered, and checkpoints are validated for usability**
- **QEMU rebuilt and re-installed with each new technique**
- **Initiated recovery from captured chcekpoints**

Success Indicator:

- **Smooth VM operation post-recovery confirms accuracy of memory and CPU checkpoints**

Evaluation - Whole System Evaluation

Focuses on the metrics

- ✓ **Data reduction:** Reduced checkpoint data and network bandwidth usage (Already done)
- ❑ **Replication time:** Reduction in checkpoint replication time
- ❑ **Faulty window:** Reduced time in faulty window during migration
- ❑ **Migration overheads:** Migration and downtime overheads with proposed approach compared to the baseline model
- ❑ **Recovery time overhead:** Time to recover the VM from checkpoints during failure
- ❑ **Performance impact:** Assessment of VM performance degradation

Evaluation - Whole System Evaluation

Comparison

- Against baseline model (Fernando et al., 2019) which uses page-level tracking

Workloads

- Standard and synthetic benchmarks: working set, iperf, sysbench, quicksort

Project Timeline

Task	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Literature survey											
Background research											
Project proposal											
Background incremental checkpointing analysis and base model implementation											
Design the modifications for the granular byte level methodology											
Implement the design in KVM/QEMU platform											
Test and analyze the overhead reduction											
Integrate compression techniques and evaluate the results											
Overall performance evaluation											
Thesis writing											
Research publication											

Summary .....

- Current incremental checkpointing tracks only the modified pages in successive checkpointing iterations.
- Network load, prolonged total migration time and error prone for failures.
- This research reduces network loads by diving deeper than page level and identify the modified bits/bytes in successive checkpointing iterations.
- Compressing the checkpointing data before transmitting to further reduce the overheads.
- Both models shows a drastic checkpoint content decrement.

Thank You!

References

- Wang, L., Kalbarczyk, Z., Iyer, R. K. & Iyengar, A. (2014), 'Vm-µcheckpoint: Design, modeling, and assessment of lightweight in-memory vm checkpointing', IEEE Transactions on Dependable and Secure Computing 12(2), 243–255. Ω´ I˜ nigo Goiri et al
- Fernando, D., Bagdi, H., Hu, Y., Yang, P., Gopalan, K., Kamhoua, C. & Kwiat, K. (2016), Quick eviction of virtual machines through proactive live snapshots, in 'Proceedings of the 9th International Conference on Utility and Cloud Computing', pp. 99–107.
- Fernando, D., Terner, J., Gopalan, K. & Yang, P. (2019), Live migration ate my vm: Recovering a virtual machine after failure of post-copy live migration, in 'IEEE INFOCOM 2019-IEEE Conference on Computer Communications', IEEE, pp. 343–351.
- Fernando, D., Terner, J., Yang, P. & Gopalan, K. (2023), 'V-recover: Virtual machine recovery when live migration fails', IEEE Transactions on Cloud Computing .
- Hou, K.-Y., Shin, K. G., Turner, Y. & Singhal, S. (2013), Tradeoffs in compressing virtual machine checkpoints, in 'Proceedings of the 7th international workshop on Virtualization technologies in distributed computing', pp. 41–48.

References

- Cully, B., Lefebvre, G., Meyer, D., Feeley, M., Hutchinson, N. & Warfield, A. (2008), Remus: High availability via asynchronous virtual machine replication, in 'Proceedings of the 5th USENIX symposium on networked systems design and implementation', San Francisco, pp. 161–174.
- Liu, H., Jin, H., Liao, X., Hu, L. & Yu, C. (2009), Live migration of virtual machine based on full system trace and replay, in 'Proceedings of the 18th ACM international symposium on High performance distributed computing', pp. 101–110.
- Gelenbe, E. (1976), A model of roll-back recovery with multiple checkpoints, in 'Proceedings of the 2nd international conference on Software engineering', pp. 251–255.
- Moody, A., Bronevetsky, G., Mohror, K. & De Supinski, B. R. (2010), Design, modeling, and evaluation of a scalable multi-level checkpointing system, in 'SC'10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis', IEEE, pp. 1–11
- ´ Inigo Goiri, Juli` a, F., Guitart, J. & Torres, J. (2010), Checkpoint-based fault tolerant infrastructure for virtualized service providers, IEEE Computer Society, pp. 455–462.
- Kannan, S., Gavrilovska, A., Schwan, K. & Milojicic, D. (2013), Optimizing check points using nvm as virtual memory, in '2013 IEEE 27th International Symposium on Parallel and Distributed Processing', IEEE, pp. 29–40.
- Knauth, T. & Fetzer, C. (2015), Vecycle: Recycling vm checkpoints for faster migrations, in 'Proceedings of the 16th Annual Middleware Conference', pp. 210 221.
- Li, G., Pattabiraman, K., Cher, C.-Y. & Bose, P. (2015), Experience report: An application-specific checkpointing technique for minimizing checkpoint corruption, in '2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)', IEEE, pp. 141–152.

Thanks!

Any Questions?

Objectives

Our aim

An aim in a corporate context is a goal or desired result that the organization seeks to achieve. The aim should be clear and achievable, and often serves as the basis for further planning actions inside the organization

The goal

Goals inside a company are usually specific and measurable, with clearly defined deadlines and outcomes. The company's goals help focus the actions of the organization and ensure resources are used effectively

Out Scope

- **Limited to KVM-QEMU Hypervisor-** The implementations will be limited to KVM-QEMU hypervisor.
- **Multiple VM failures-** Only single VM failures will be handled. Can be extended to multiple VM failure scenarios.
- **Non-Linux based OSs-** The implementations will be limited to linux based Ubuntu operating systems.

Resources

Human resources

The project team is responsible for the successful execution of the project. Our team is composed of experienced professionals with the necessary skills and expertise to complete the project on time and within budget

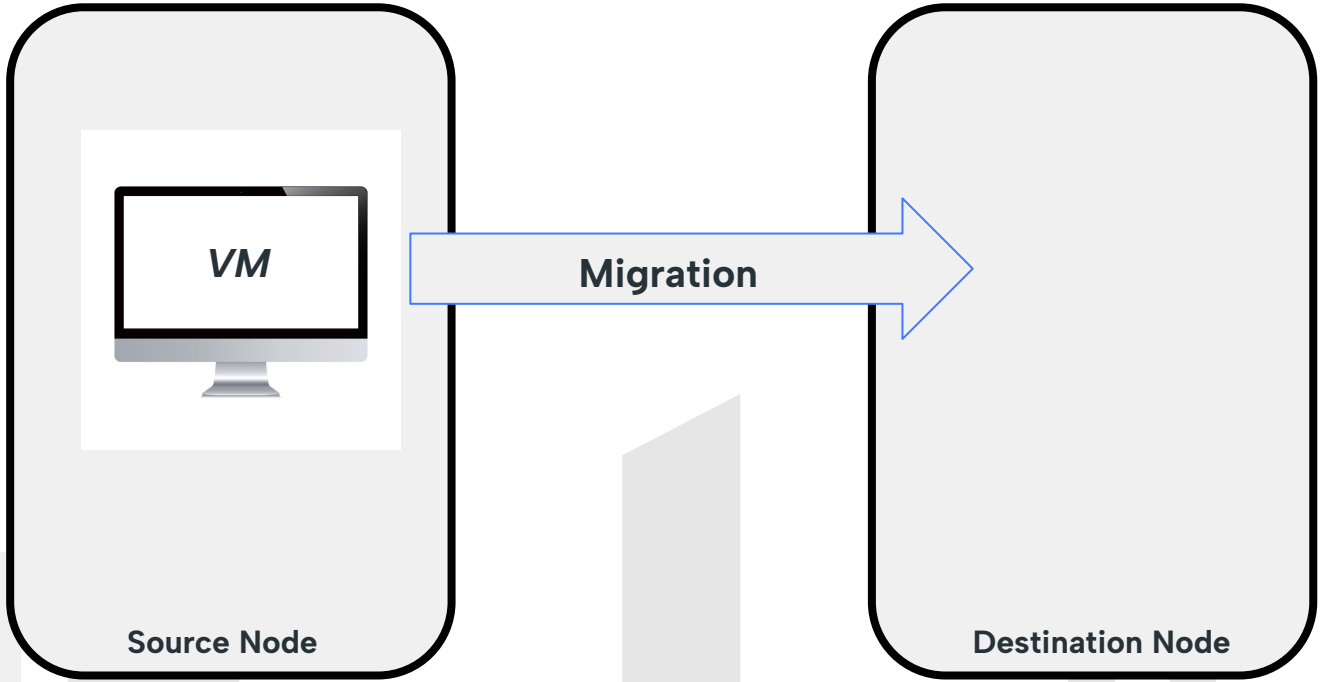
Financial resources

Our project budget is $100,000. This budget includes all expenses associated with the project, including salaries and equipment. We have allocated these resources to ensure that we are able to complete the project within budget

Physical resources

Our project requires a number of specialized pieces of equipment. We will be using [insert equipment and their functions]. All equipment is in good working condition and has been tested and calibrated prior to use

Live VM Migration



Budget

Sources of funding

Funding for a company can come from personal savings or investments, bank loans and other loan options, venture capital and angel investors, grants, competitions or programs, crowdfunding...

Personnel costs

Personnel costs refer to the expenses incurred in hiring, training and retaining staff for a company. This can include salaries, bonuses, benefits and other payroll-related costs

Equipment and materials

Equipment and materials costs refer to all the expenses related to the purchasing, maintenance and upkeep of any physical items used in production or other business processes inside the company

Travel and miscellaneous

Travel and miscellaneous costs refer to expenses related to any travel-related activities, such as conferences, trainings or business trips. It may also include office supplies, communications services, licenses and other miscellaneous expenses

Thanks!

Do you have any questions?

Do you have any questions?

CREDITS: This presentation template was created by Slidesgo, and includes icons by Flaticon, and infographics & images by Freepik

Content

01 Introduction & Background

02 Motivation

03 Related Work

04 Research Gap

05 Research Questions

06 Objectives

07 Scope

08 Research Approach

09 Progress

Contents of this template

You can delete this slide when you're done editing the presentation

Fonts	To view this template correctly in PowerPoint, download and install the fonts we used
Used and alternative resources	An assortment of graphic resources that are suitable for use in this presentation
Thanks slide	You must keep it so that proper credits for our design are given
Colors	All the colors used in this presentation
Icons and infographic resources	These can be used in the template, and their size and color can be edited
Editable presentation theme	You can edit the master slides easily. For more info, click **here**

For more info:
SLIDESGO | BLOG | FAQs

You can visit our sister projects:
FREEPIK | FLATICON | STORYSET | WEPIK | VIDEVO

Solutions

Solution 1

Implementing a new CRM (Customer Relationship Management) system to improve customer data management and sales tracking

Solution 2

Outsourcing specific business functions (such as accounting or IT) to a third-party provider to reduce costs and increase time efficiency

Solution 3

Launching an e-commerce platform to expand the reach of the business and increase online sales

Solution 4

Developing a new product or service to diversify the business and increase revenue streams

Solution 5

Implementing a cost-saving initiative, such as energy-efficient practices or process automation, to reduce expenses

Solution 6

Establishing strategic partnerships with other businesses to gain access to new markets or innovative technologies

200

Units break-even point

$500

Net profit of the project

25%

Market share in the industry

10m

first-year revenue of the project

What to show in a mockup

1. Product/website description: A brief overview of the product/website, including its key features, dimensions, and materials used
2. Features and benefits: A detailed explanation of the product's/website's features and how they will benefit the user
3. Technical specifications: A list of the product's/website's technical specifications, such as dimensions, weight, power requirements, connectivity options and hosting platform

Project timeline

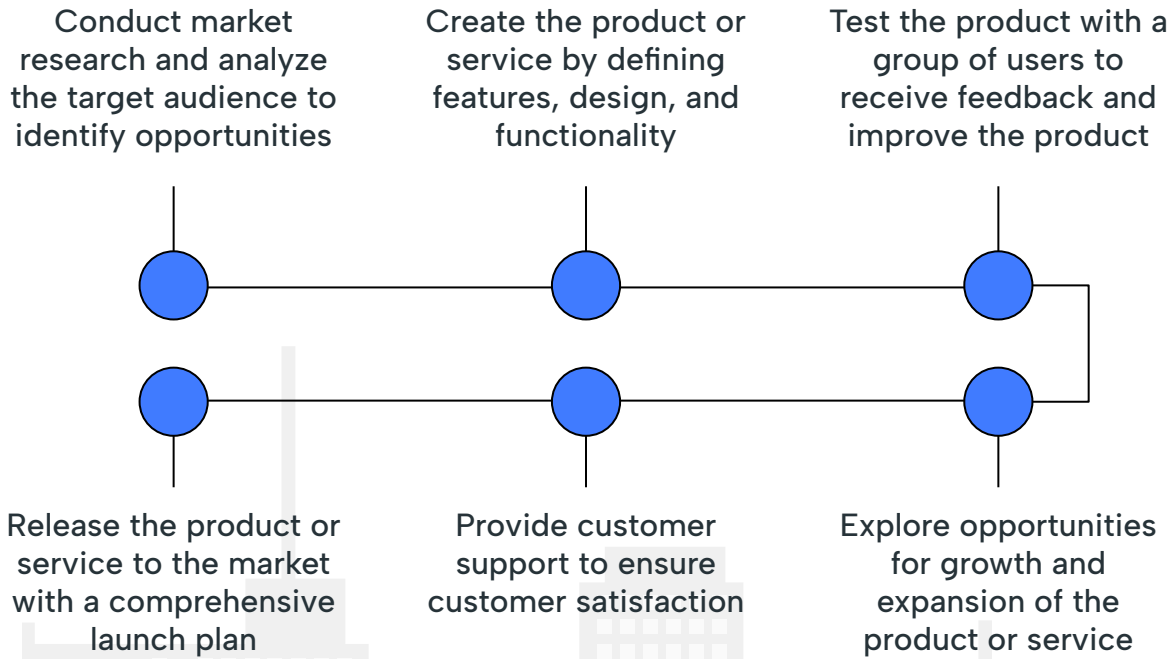
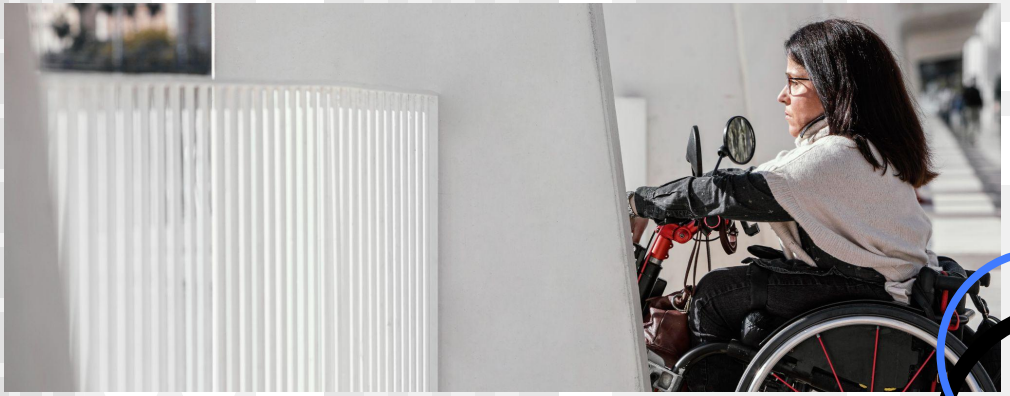
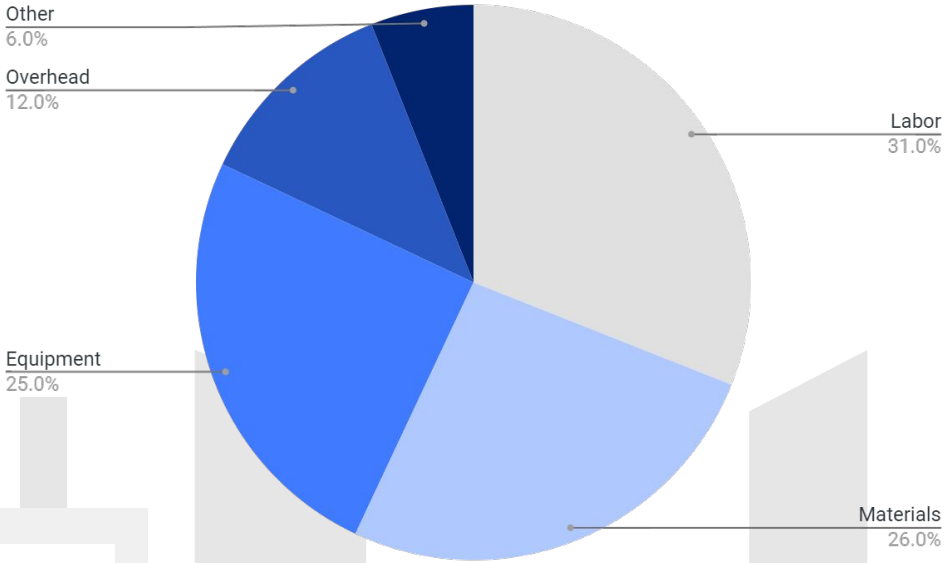


Photo showcase

A photo showcase can be a useful addition to a business project proposal as it can help to visually communicate the concept or idea being proposed

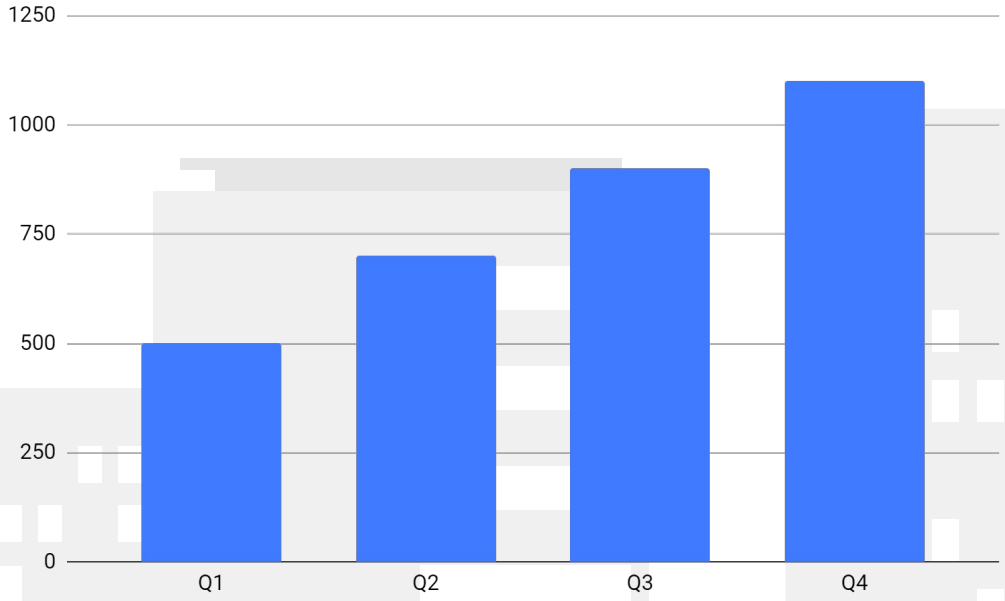


Project expenses



Follow the link in the graph to modify its data and then paste the new one here. **For more info, click here**

Project data



Benefits of using tables

Tables in project proposals offer visual organization, enabling clear presentation of information in a structured format. They enhance the visual appeal, facilitate data comparison and improve overall clarity and professionalism of your project proposal

Follow the link in the graph to modify its data and then paste the new one here. **For more info, click here**

KPI dashboard

Resource	Utilization rate	Cost per unit
Labor	85%	$50
Equipment	70%	$100
Materials	95%	$20
Rent	90%	$1,000
Energy	80%	$80
Software licenses	80%	$200
Advertising	60%	$500

120 u

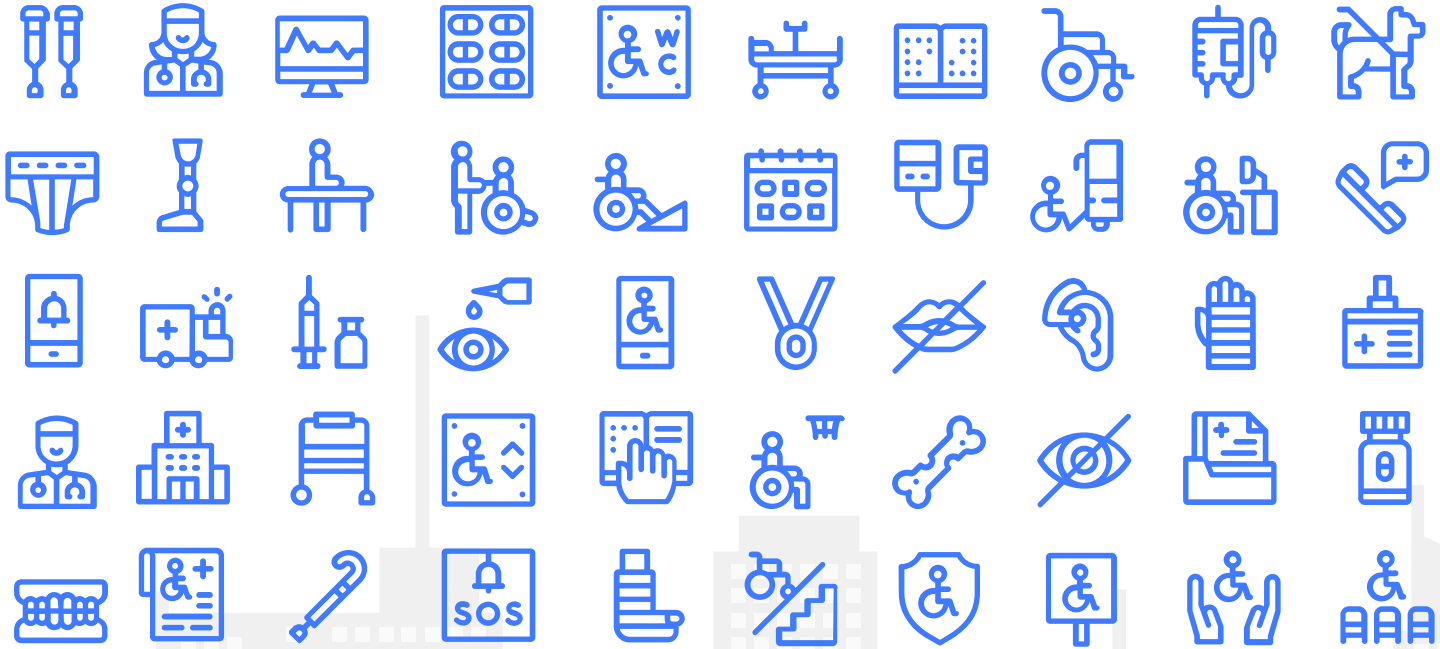
Output per worker

2h

Time to complete a task

Follow the link in the graph to modify its data and then paste the new one here. **For more info, click here**

Icon pack



Alternative resources

Here’s an assortment of alternative resources whose style fits the one of this template:

- Building concept illustration

Resources

Did you like the resources on this template? Get them for free at our other websites:

Vectors

- Family with a disabled parent concept illustration
- Flat design young people helping the elderly illustration
- Life in a city concept illustration
- Twilight city graphic panorama
- Modern urban landscape with flat design
- Colorful and polygonal city
- Flat urban landscape with office buildings

Photos

- Adult having fun with their disabled friend
- Disabled man helping girl learn
- Disabled person in wheelchair on the street
- Close-up hand moving wheel
- Side view of woman in a wheelchair in the city
- Woman having fun with her her disabled friend
- Full shot happy friends outdoors
- Woman having fun with her her disabled friend

Icons

- Icon pack: Disabled people assistance

Instructions for use

If you have a free account, in order to use this template, you must credit **Slidesgo** by keeping the **Thanks** slide. Please refer to the next slide to read the instructions for premium users.

As a Free user, you are allowed to:

- Modify this template.
- Use it for both personal and commercial projects.

You are not allowed to:

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute Slidesgo Content unless it has been expressly authorized by Slidesgo.
- Include Slidesgo Content in an online or offline database or file.
- Offer Slidesgo templates (or modified versions of Slidesgo templates) for download.
- Acquire the copyright of Slidesgo Content.

For more information about editing slides, please read our FAQs or visit our blog:
https://slidesgo.com/faqs and **https://slidesgo.com/slidesgo-school**

Instructions for use (premium users)

As a Premium user, you can use this template without attributing **Slidesgo** or keeping the "**Thanks**" slide.

You are allowed to:

- Modify this template.
- Use it for both personal and commercial purposes.
- Hide or delete the “Thanks” slide and the mention to Slidesgo in the credits.
- Share this template in an editable format with people who are not part of your team.

You are not allowed to:

- Sublicense, sell or rent this Slidesgo Template (or a modified version of this Slidesgo Template).
- Distribute this Slidesgo Template (or a modified version of this Slidesgo Template) or include it in a database or in any other product or service that offers downloadable images, icons or presentations that may be subject to distribution or resale.
- Use any of the elements that are part of this Slidesgo Template in an isolated and separated way from this Template.
- Register any of the elements that are part of this template as a trademark or logo, or register it as a work in an intellectual property registry or similar.

For more information about editing slides, please read our FAQs or visit our blog:
https://slidesgo.com/faqs and **https://slidesgo.com/slidesgo-school**

Fonts & colors used

This presentation has been made using the following fonts:

Raleway
(https://fonts.google.com/specimen/Raleway)

Albert Sans
(https://fonts.google.com/specimen/Albert+Sans)

Storyset

Create your Story with our illustrated concepts. Choose the style you like the most, edit its colors, pick the background and layers you want to show and bring them to life with the animator panel! It will boost your presentation. Check out **how it works**.

Pana

Amico

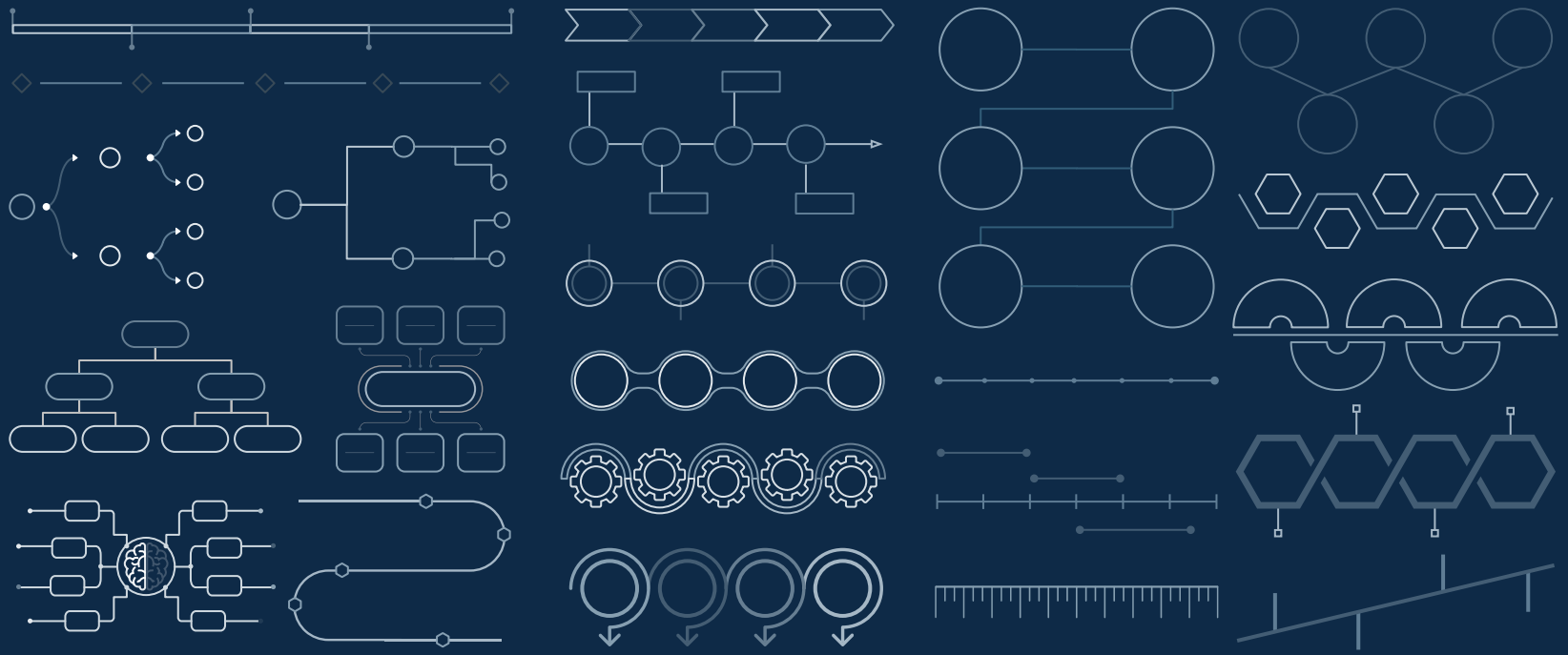
Bro

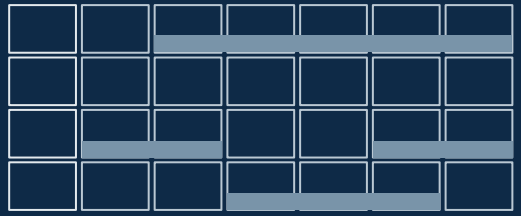
Rafiki

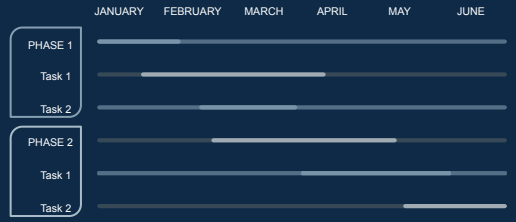
Cuate

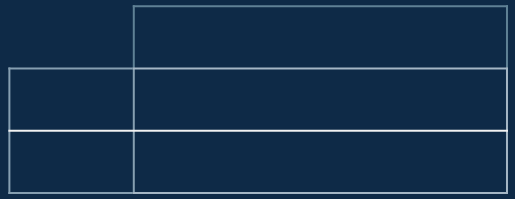
Use our editable graphic resources...

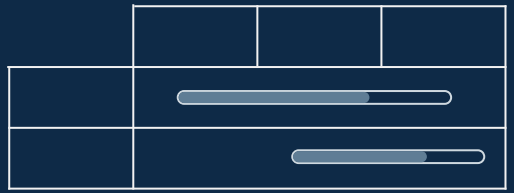
You can easily **resize** these resources without losing quality. To **change the color**, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Group the resource again when you're done. You can also look for more **infographics** on Slidesgo.

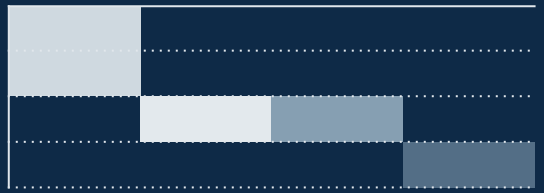


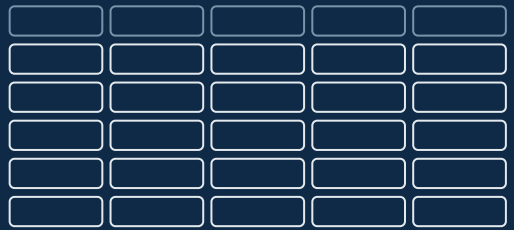


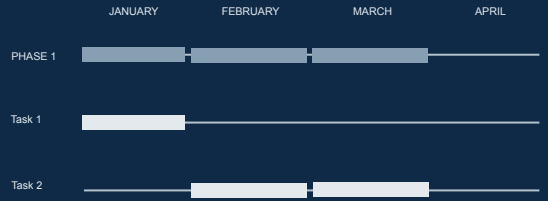

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
PHASE 1
Task 1
Task 2
PHASE 2
Task 1
Task 2

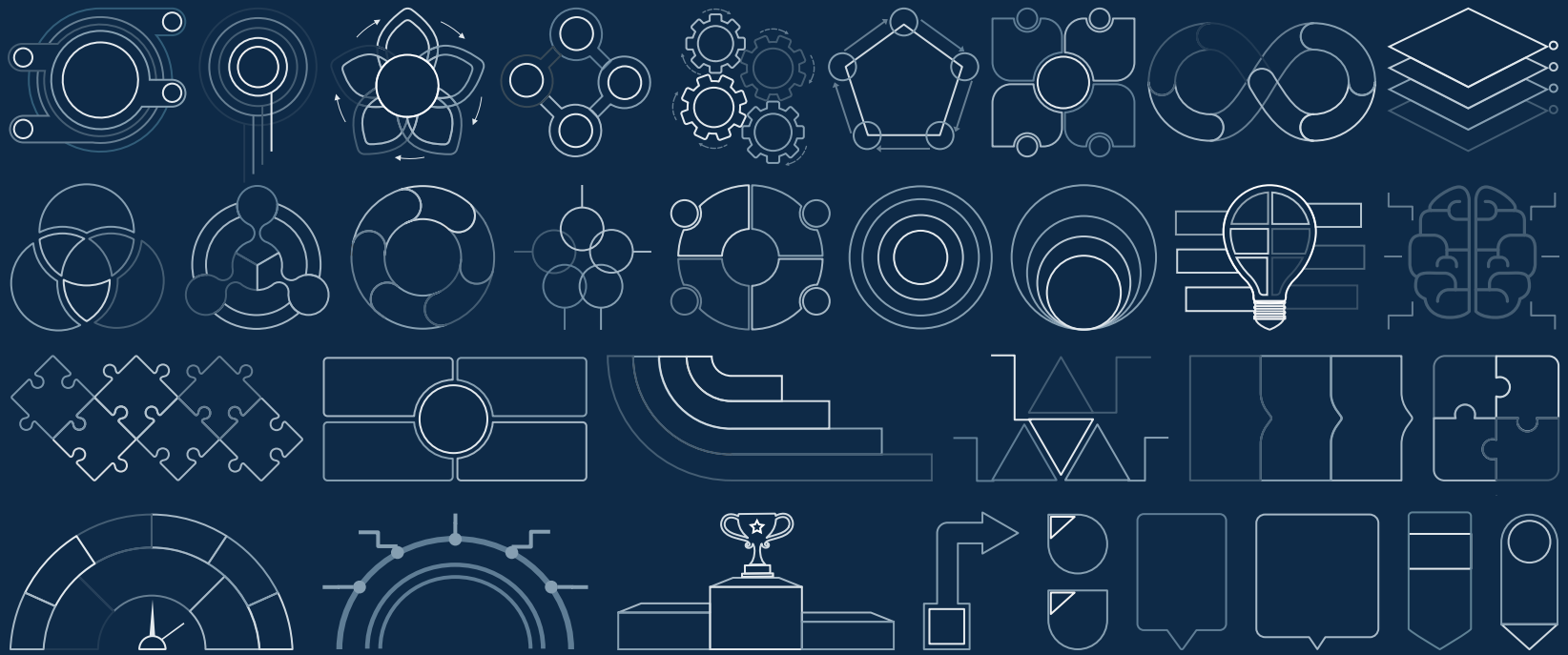


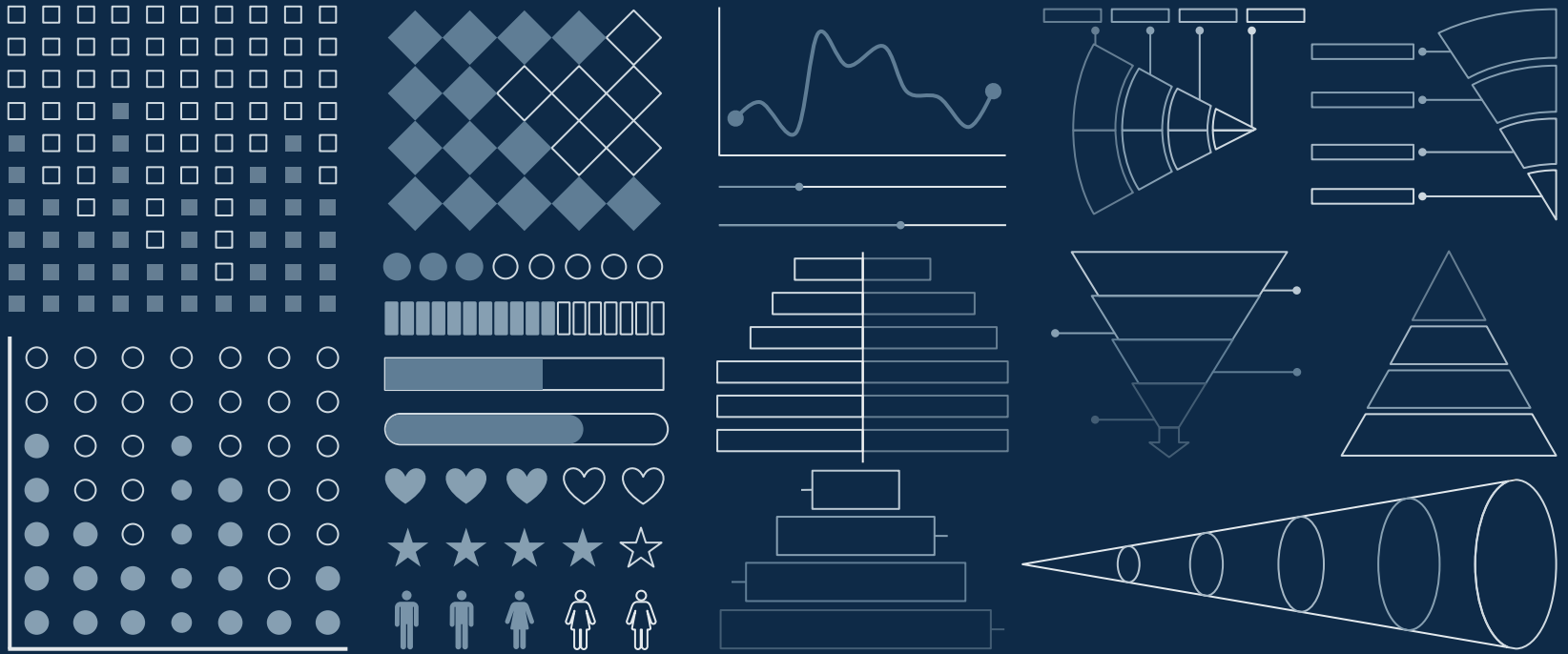







JANUARY
FEBRUARY
MARCH
APRIL
PHASE 1
Task 1
Task 2





...and our sets of editable icons

You can **resize** these icons without losing quality.
You can **change the stroke and fill color**; just select the icon and click on the **paint bucket/pen**.
In Google Slides, you can also use **Flaticon's extension**, allowing you to customize and add even more icons.

Educational Icons

Medical Icons

Business Icons

Teamwork Icons

Help & Support Icons

Avatar Icons

Creative Process Icons

Performing Arts Icons

Nature Icons

SEO & Marketing Icons


slidesgo