Throughput and Memory Optimization for Parallel Implementations of Dataflow Networks using Multi-Reader Buffers

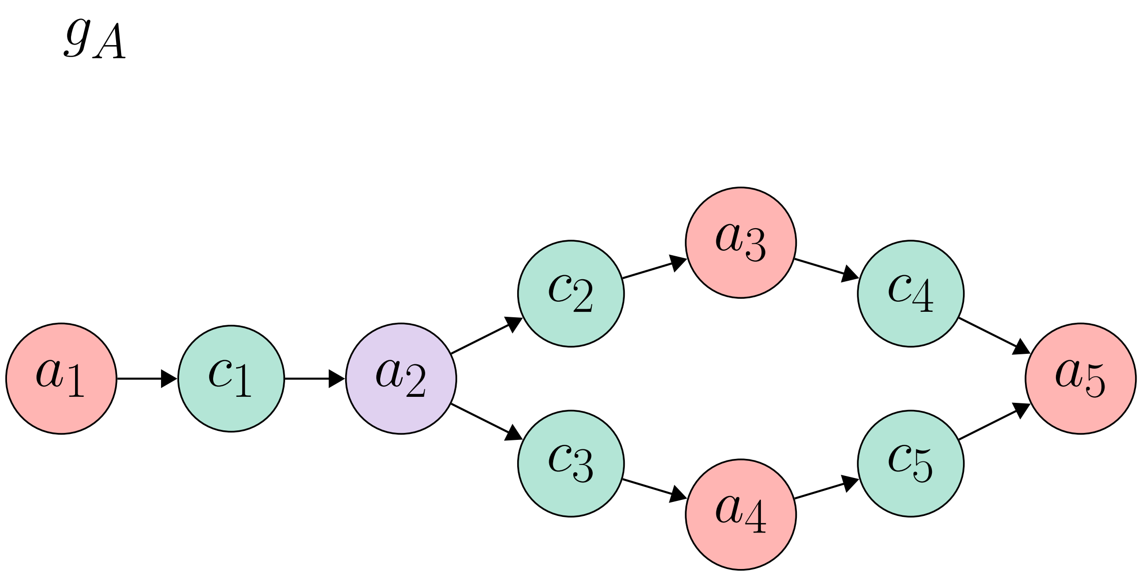
NG-RES: Workshop on Next Generation Real-Time Embedded Systems

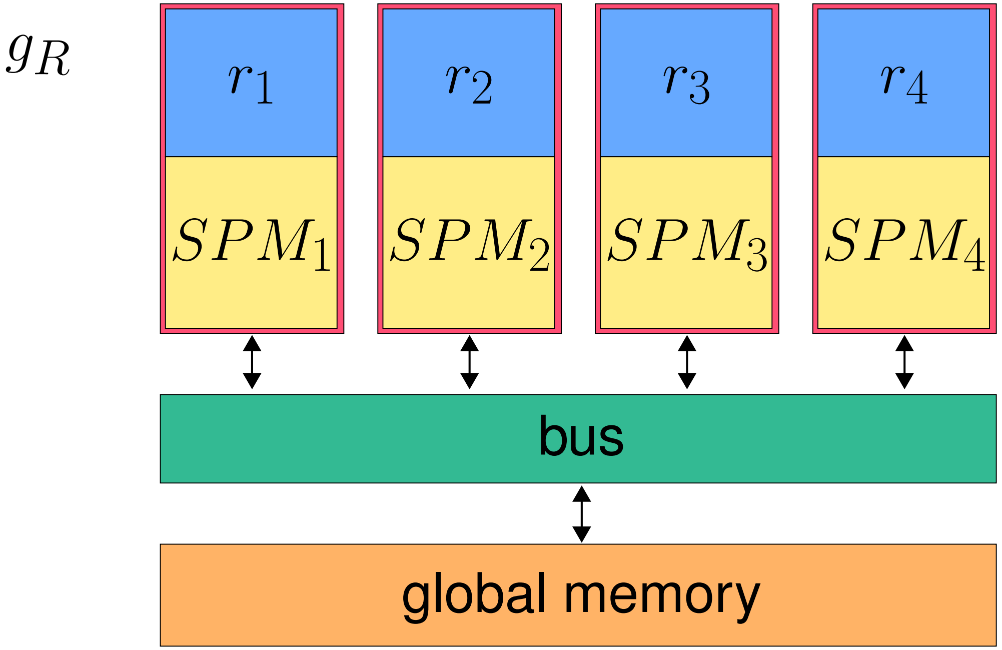
Martín Letras*, **Joachim Falk and Jürgen Teich**

Friedrich-Alexander-Universität Erlangen-Nürnberg, Hardware-Software-Co-Design

January 18, 2023

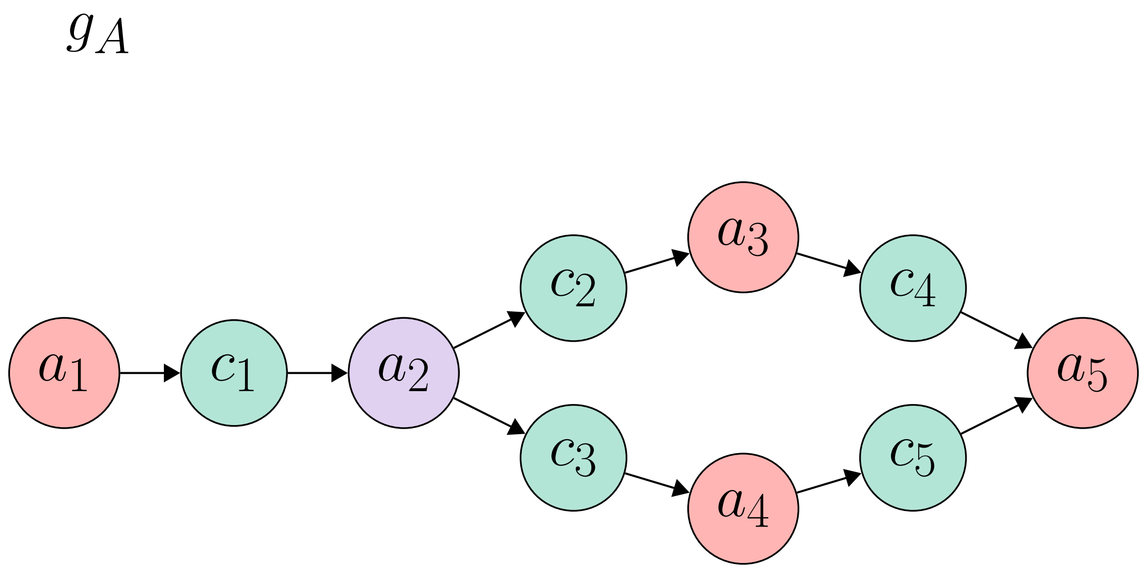
Introduction

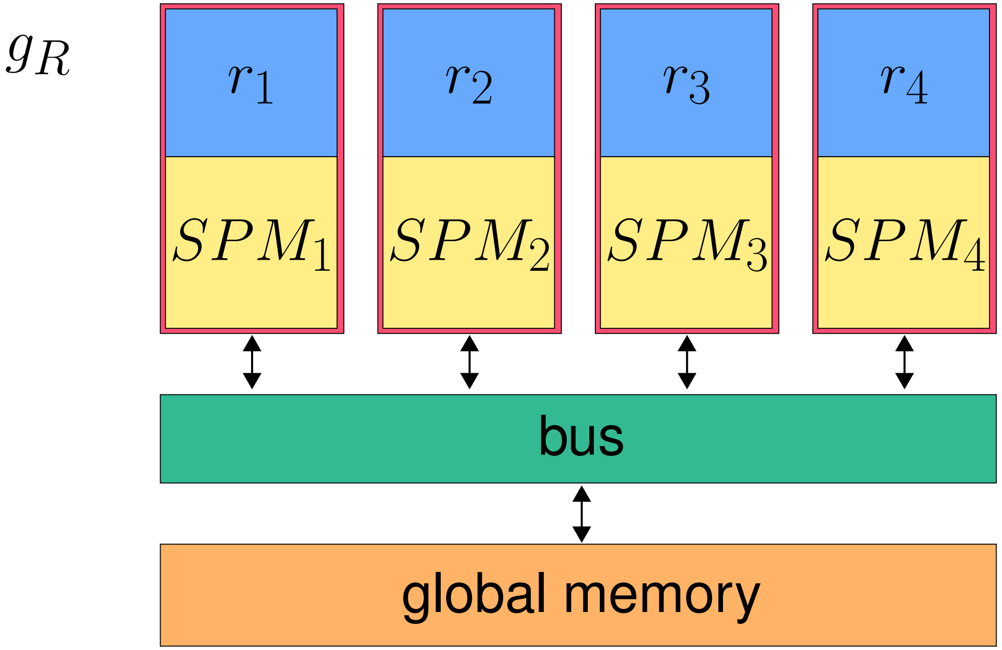




- A **specification graph** g_{S} is composed of an application graph g_{A}, an architecture g_{R} and a set of mappings M

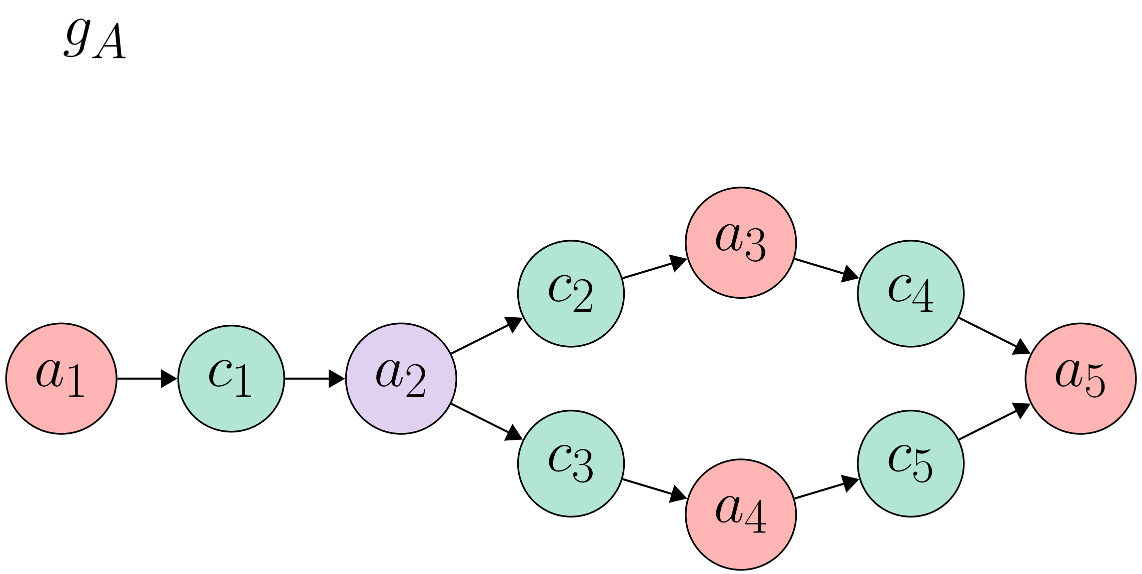
Introduction

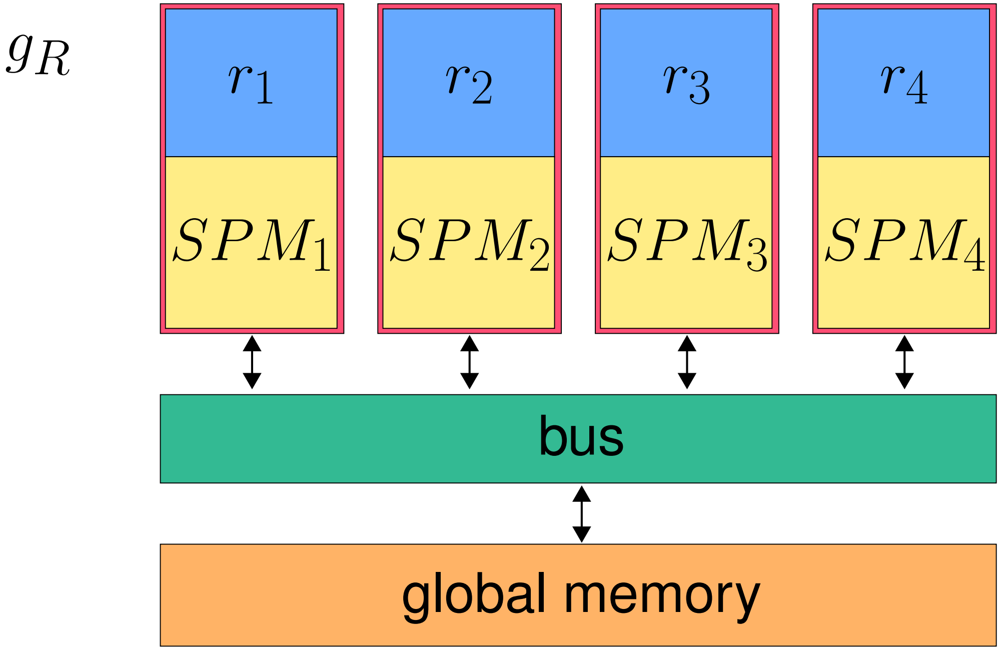




- A **specification graph** g_S is composed of an **application graph** g_A, an architecture g_R and a set of mappings M

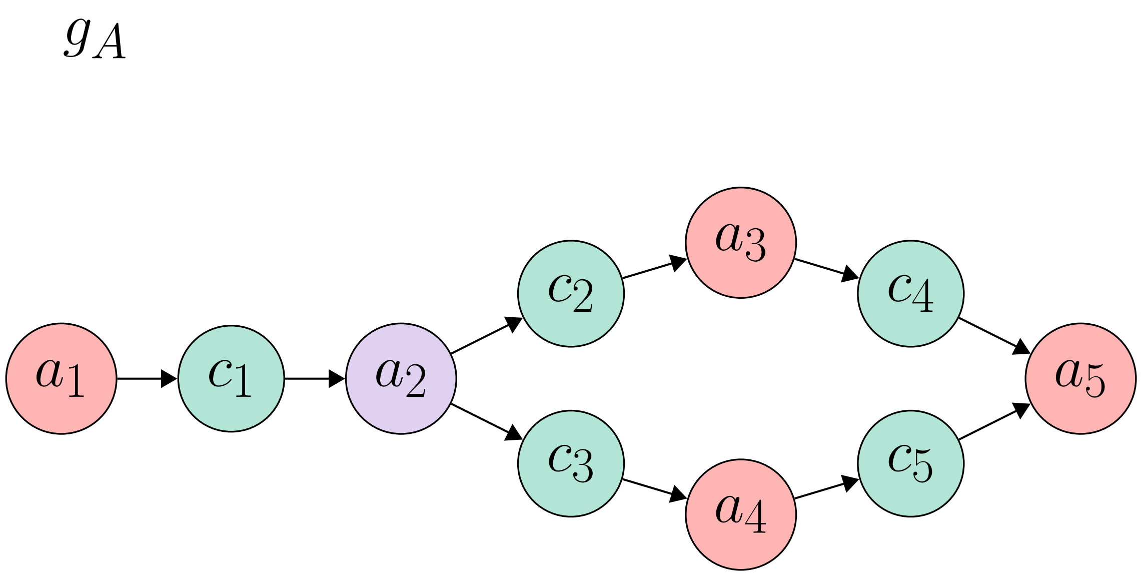
Introduction

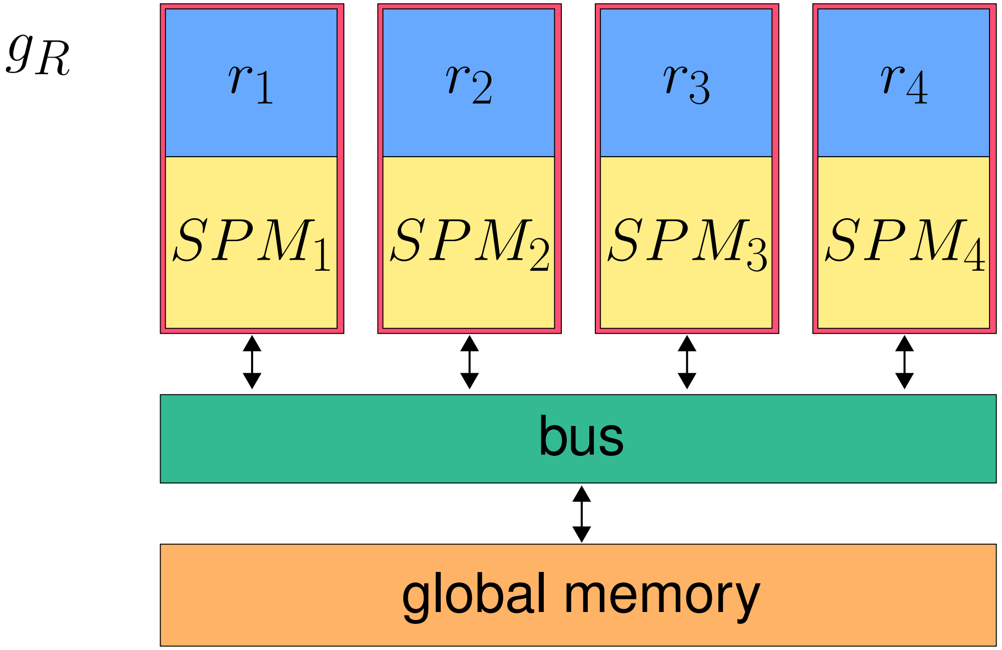




- A **specification graph** g_{S} is composed of an **application graph** g_{A}, an **architecture** g_{R} and a set of mappings M

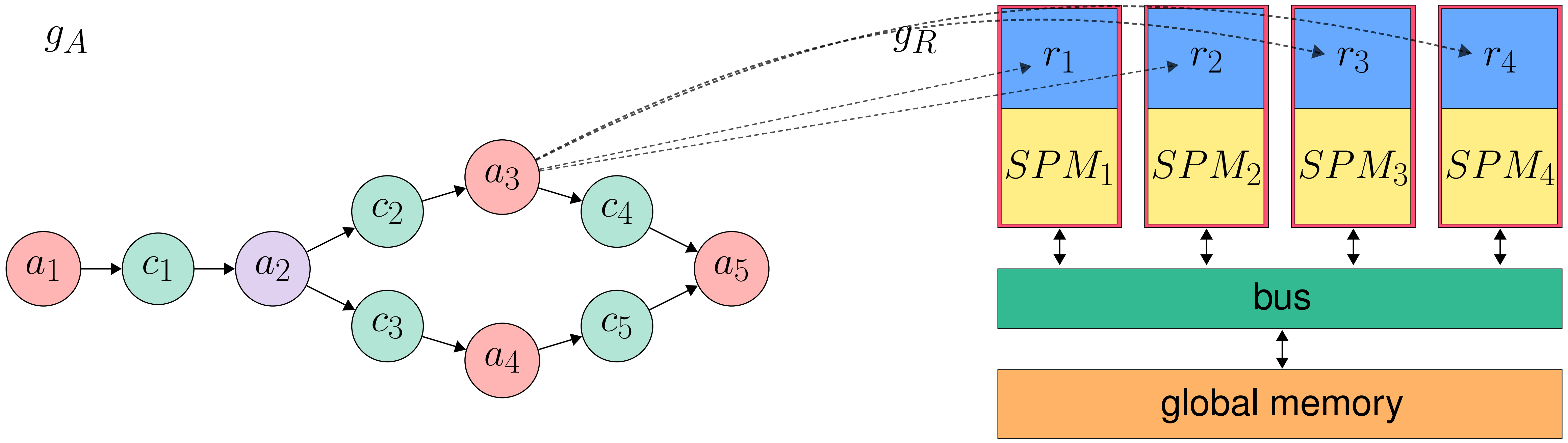
Introduction





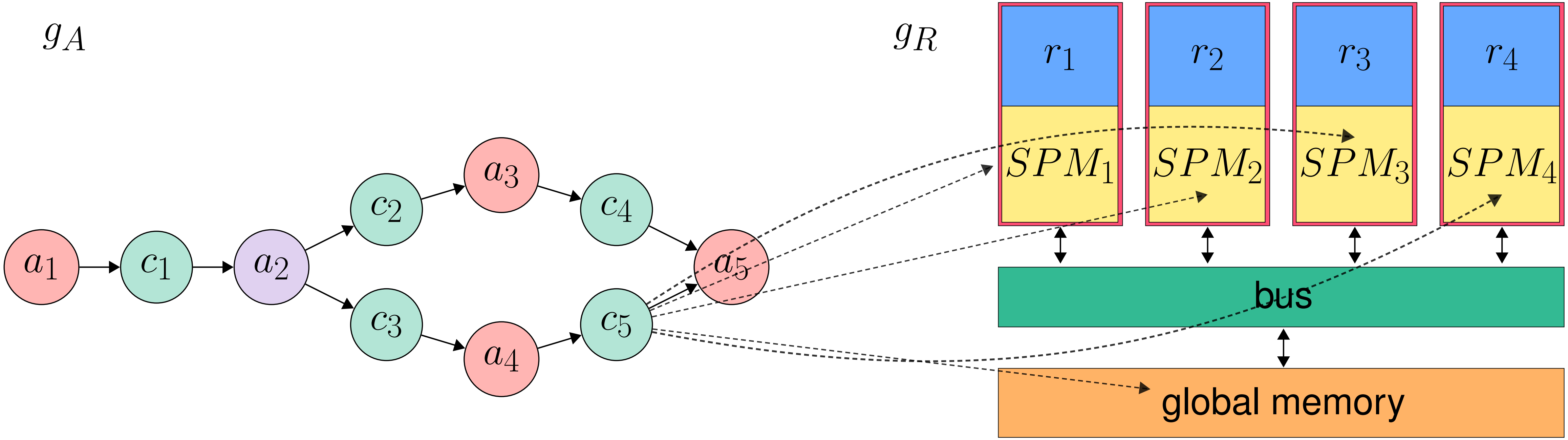
- A **specification graph** g_S is composed of an **application graph** g_A, an **architecture** g_R and a **set of mappings** M

Introduction



- A **specification graph** g_{S} is composed of an **application graph** g_{A}, an **architecture** g_{R} and a **set of mappings** M
  - Each **actor** $a \in A$ can be mapped to **any processor** $r \in R$

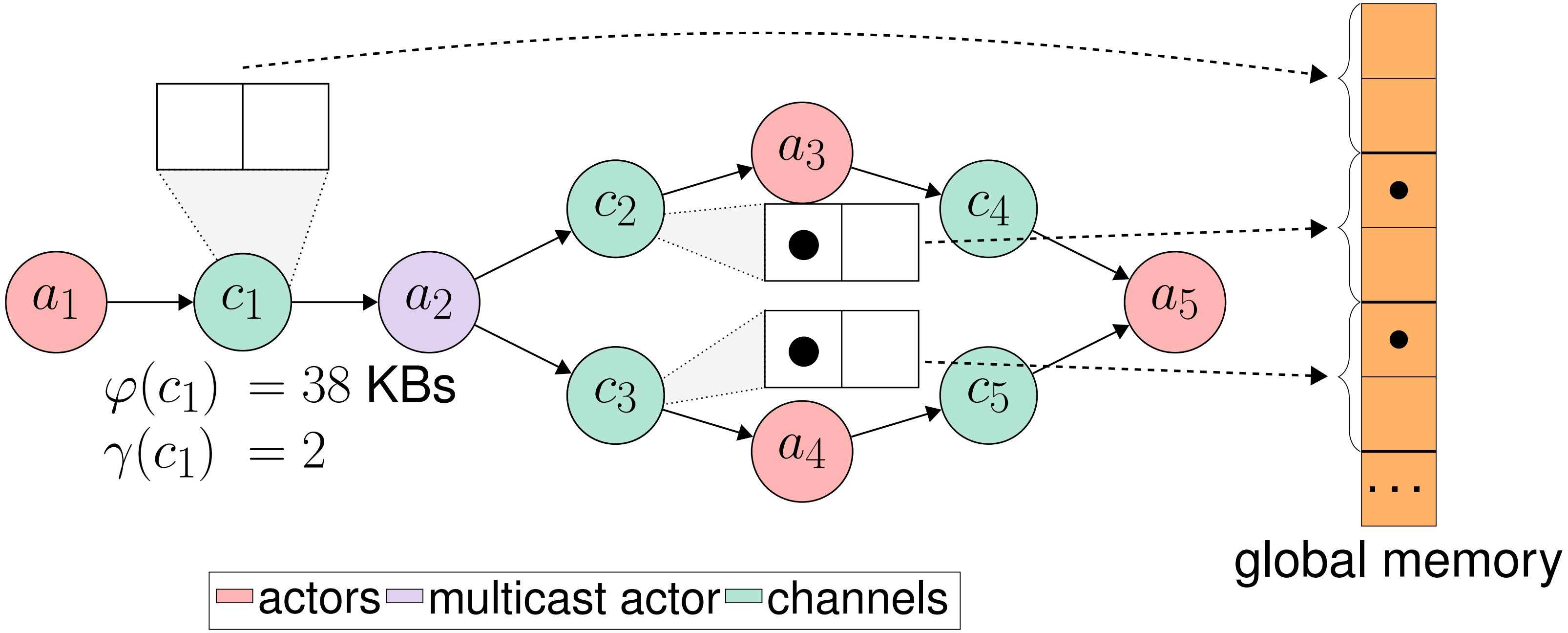
Introduction



- A **specification graph** g_S is composed of an **application graph** g_A, an **architecture** g_R and a **set of mappings** M
  - Each **actor** $a \in A$ can be mapped to **any processor** $r \in R$
  - Each **communication channel** $c \in C$ can be mapped to a **global memory** or to a **scratchpad memory**

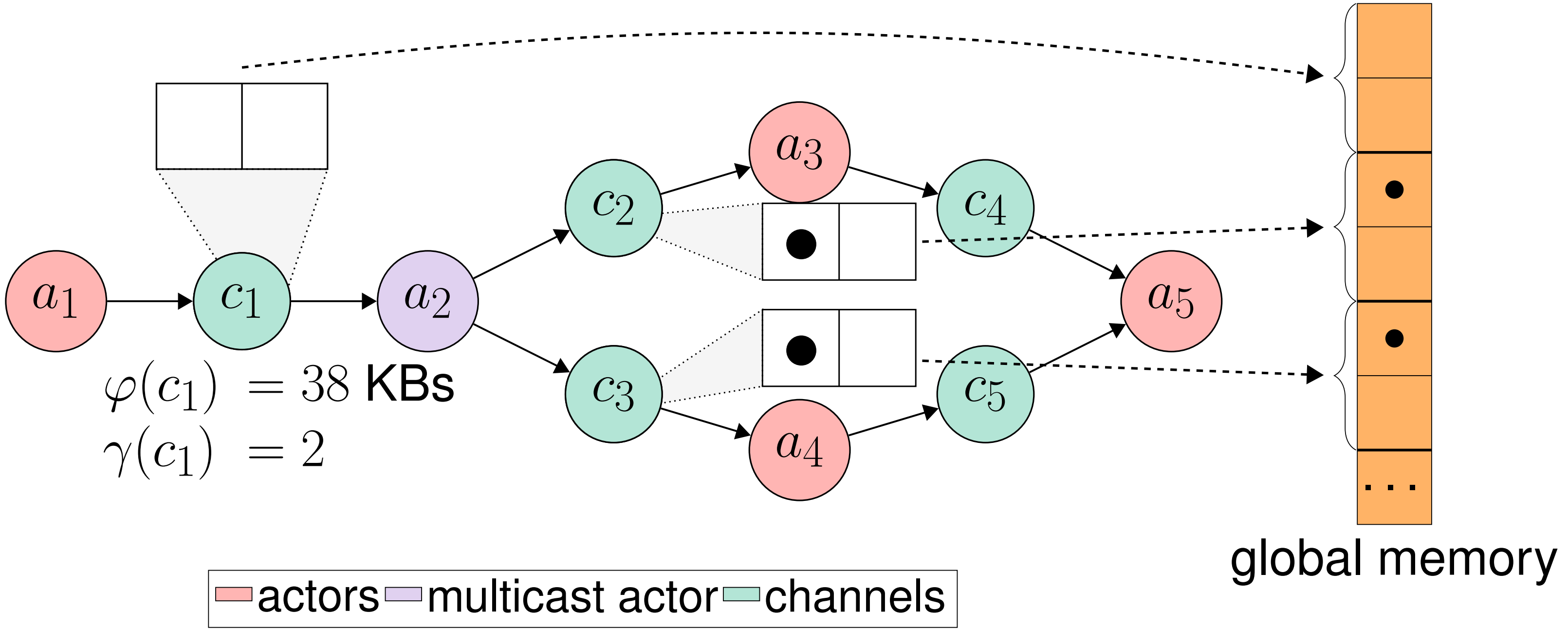
Introduction

- Each **communication channel** is implemented as a **FIFO allocated in memory**



- Exploration of memory footprint reductions of multicast actors
- We introduce the concept of **Multi-Reader Buffer** to replace multicast actors in stream processing applications to reduce memory footprints.

- Each **communication channel** is implemented as a **FIFO allocated in memory**

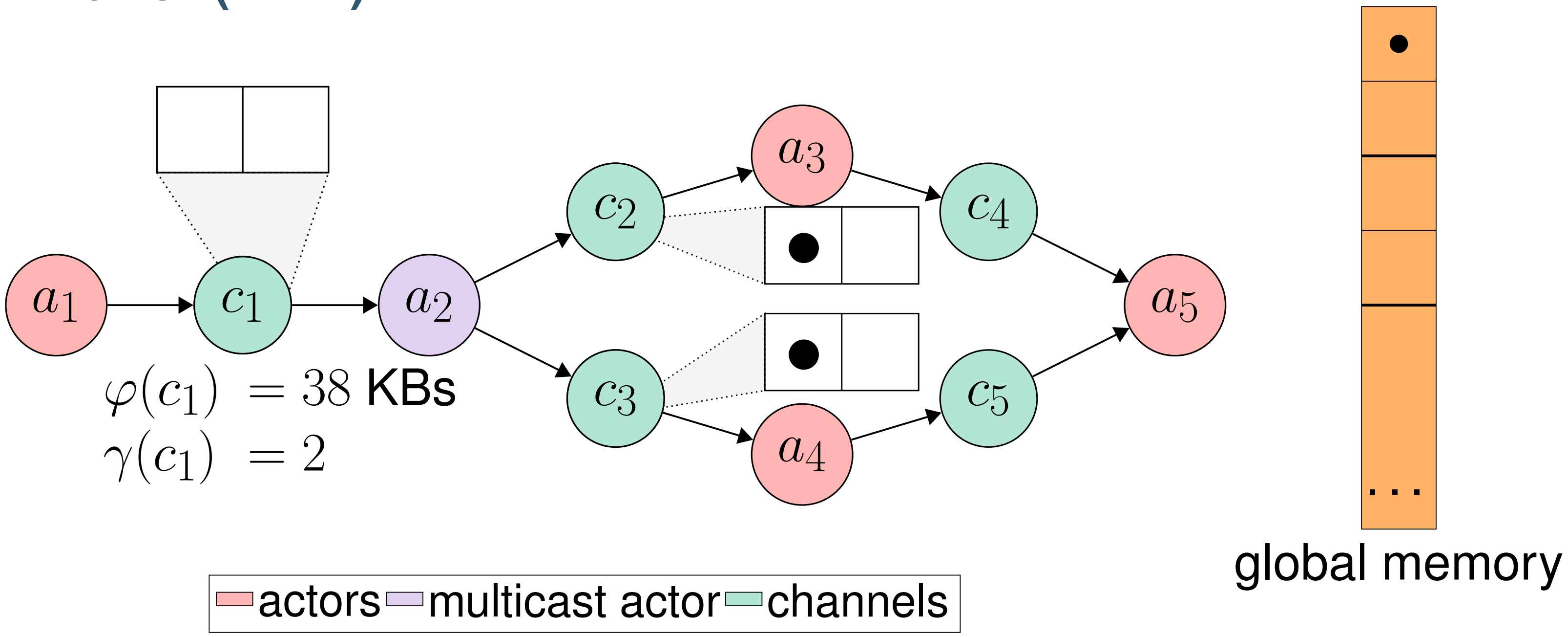


- The **memory footprint** of a given application is calculated as:

$$M_F = \sum_{\forall c \in C} \gamma(c) \times \varphi(c) = 5 \times 2 \times 38 \text{ KBs} = 380 \text{ KBs}$$

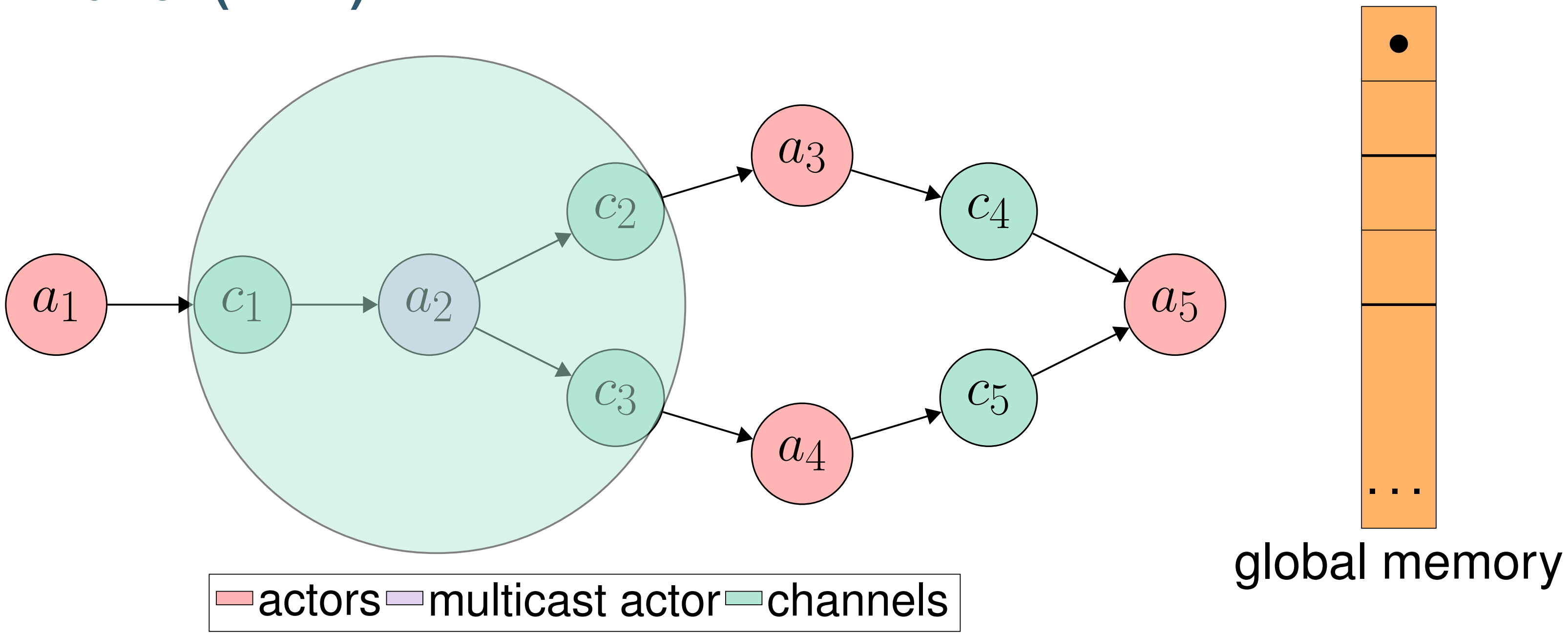
Exploiting Data Redundancy to Reduce Memory Footprint

- Memory footprint might be reduced by **merging those redundant channels into a Multi-Reader Buffer (MRB)**



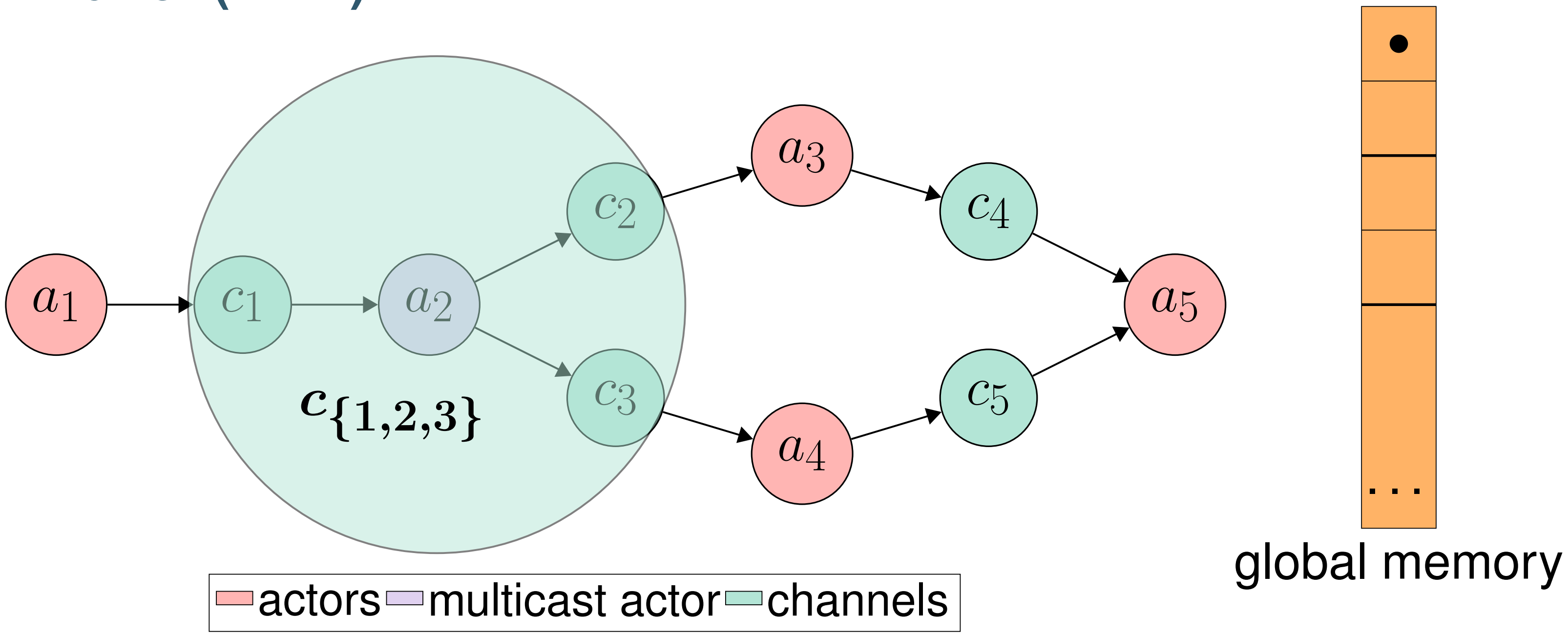
Exploiting Data Redundancy to Reduce Memory Footprint

- Memory footprint might be reduced by **merging those redundant channels into a Multi-Reader Buffer (MRB)**



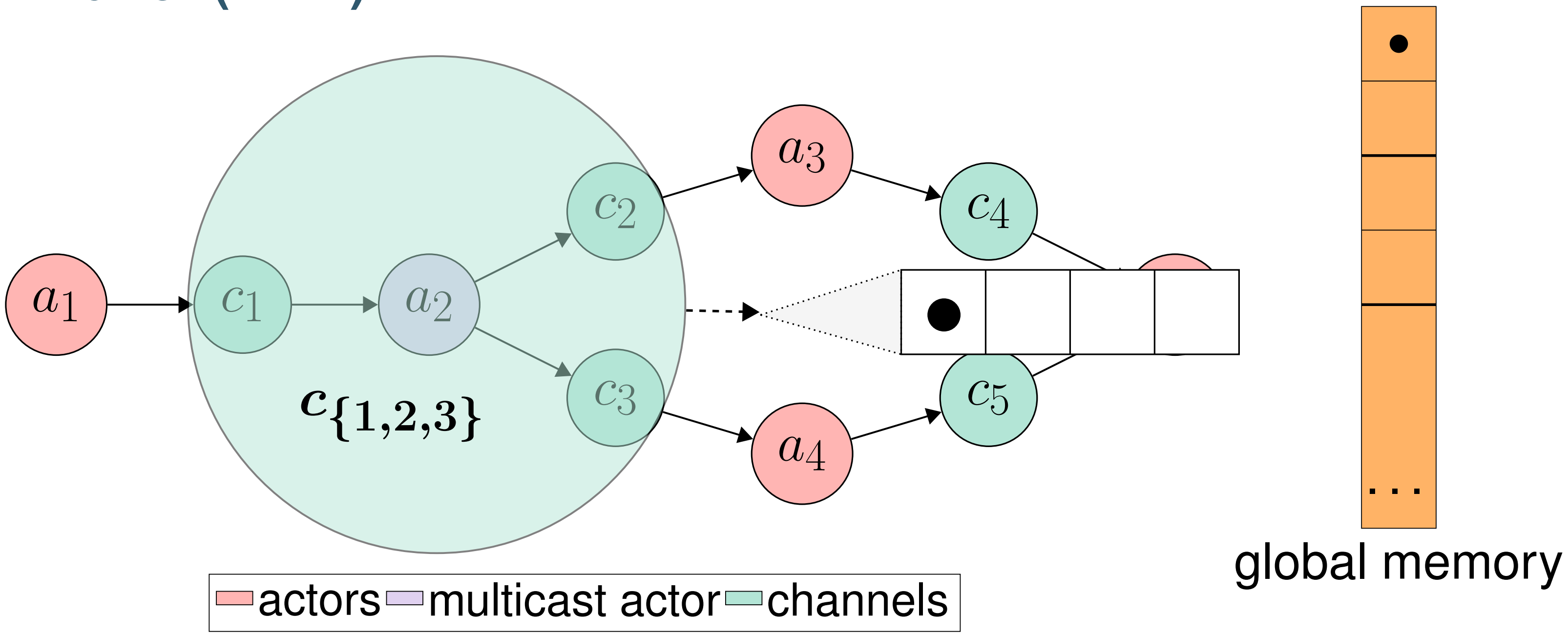
Exploiting Data Redundancy to Reduce Memory Footprint

- Memory footprint might be reduced by **merging those redundant channels into a Multi-Reader Buffer (MRB)**



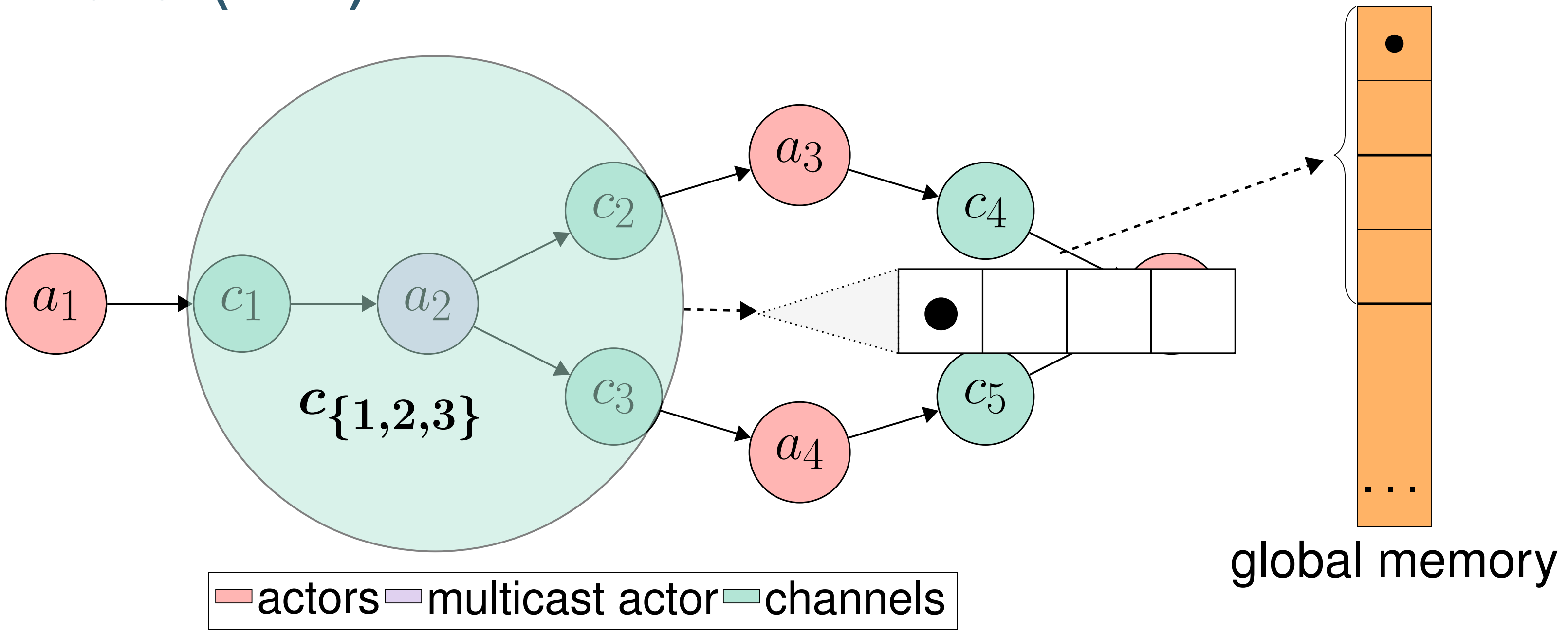
Exploiting Data Redundancy to Reduce Memory Footprint

- Memory footprint might be reduced by **merging those redundant channels into a Multi-Reader Buffer (MRB)**



Exploiting Data Redundancy to Reduce Memory Footprint

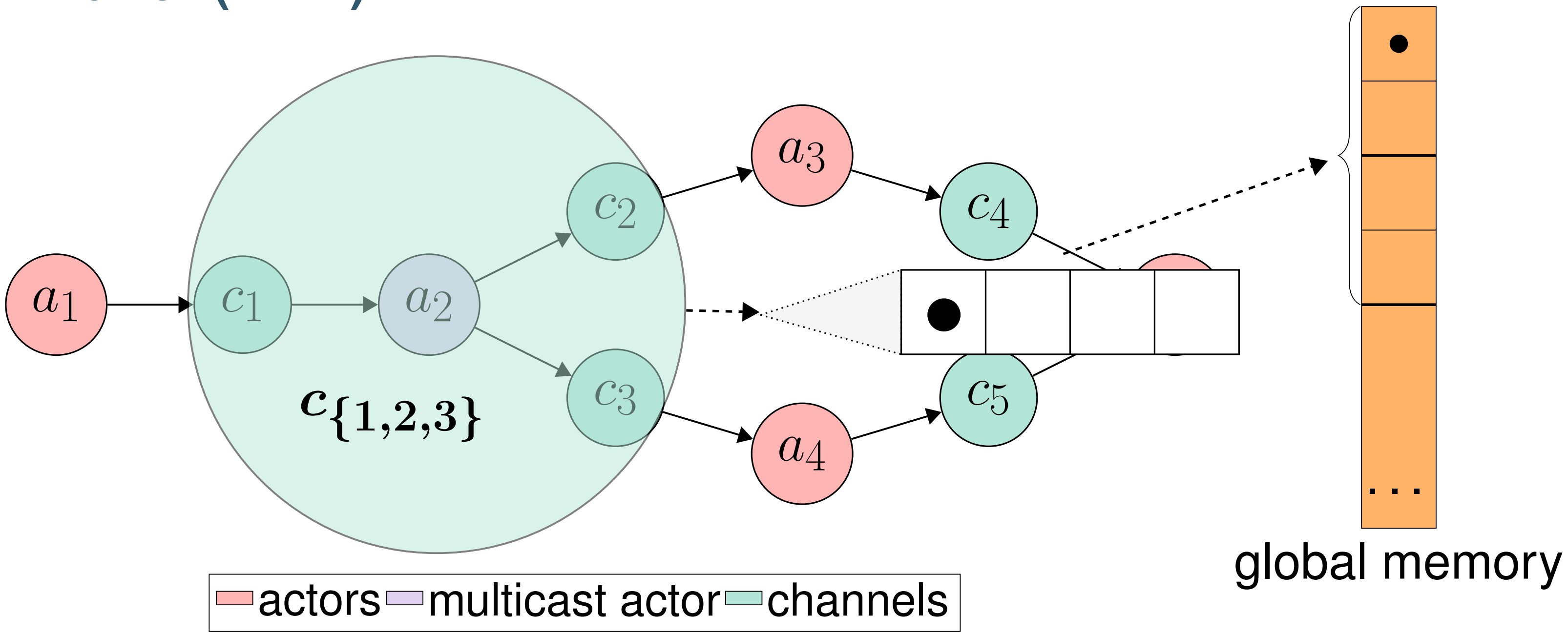
- Memory footprint might be reduced by **merging those redundant channels into a Multi-Reader Buffer (MRB)**



- Here, $c_{\{1,2,3\}}$ stores **only one copy of live data** into memory

Exploiting Data Redundancy to Reduce Memory Footprint

- Memory footprint might be reduced by **merging those redundant channels into a Multi-Reader Buffer (MRB)**



- Here, $c_{\{1,2,3\}}$ stores **only one copy of live data** into memory

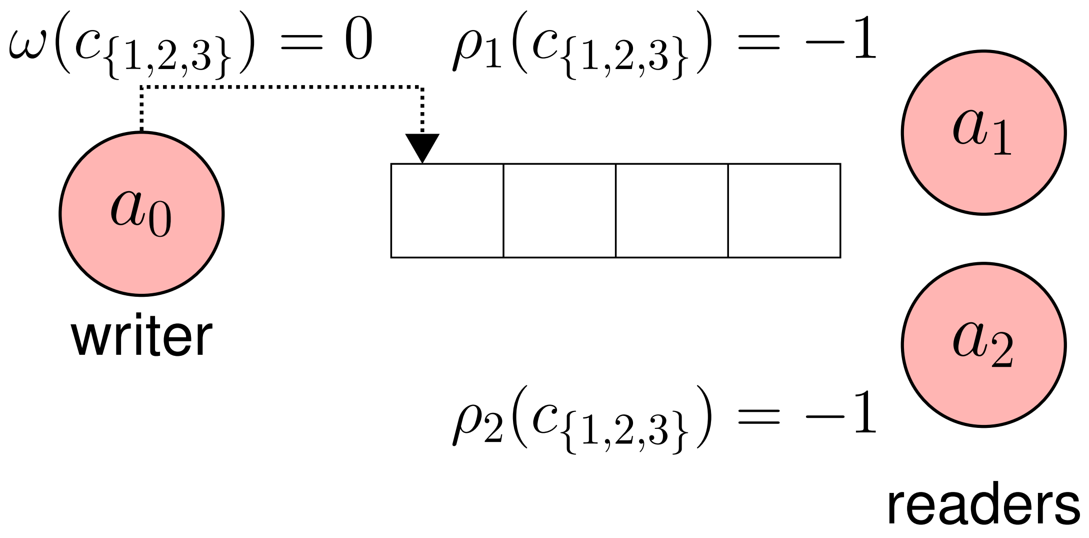
$$M_F = \sum_{\forall c \in C} \gamma(c) \times \varphi(c) = (4 \times 38 \text{ KBs}) + (2 \times 2 \times 38 \text{ KBs}) = 304 \text{ KBs}$$

Multi-Reader Buffer (MRB)

Definition

By definition, an MRB $c_\mathbf{m}$ has **one writer** $a_\mathbf{w}$ and **multiple readers** $a_{\mathbf{r}_i} \in A_\mathbf{r} \subseteq A$

- Each $c_\mathbf{m}$ has a write index $\omega(c_\mathbf{m}) \in \{0, 1, \ldots, \gamma(c_\mathbf{m}) - 1\}$
- Similarly, each $c_\mathbf{m}$ manages read indices $\rho_i(c_\mathbf{m}) \in \{-1, 0, 1, \ldots, \gamma(c_\mathbf{m}) - 1\}$



Multi-Reader Buffer (MRB)

Semantics

By definition, an MRB c_m has **one writer** a_w and **multiple readers** $a_{\mathsf{r}_i} \in A_\mathsf{r} \subseteq A$

- MRB's state from a_{r_i} and a_w perspective:

$$\mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \begin{cases} 0 & \text{if } \rho_i(c_\mathsf{m}) < 0 \\ ((\omega(c_\mathsf{m}) - \rho_i(c_\mathsf{m}) - 1) \bmod \gamma(c_\mathsf{m})) + 1 & \text{otherwise} \end{cases} \quad (1)$$

$$\mathsf{F}(c_\mathsf{m}) = \gamma(c_\mathsf{m}) - \max_{a_{\mathsf{r}_i} \in A_\mathsf{r}} \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) \quad (2)$$

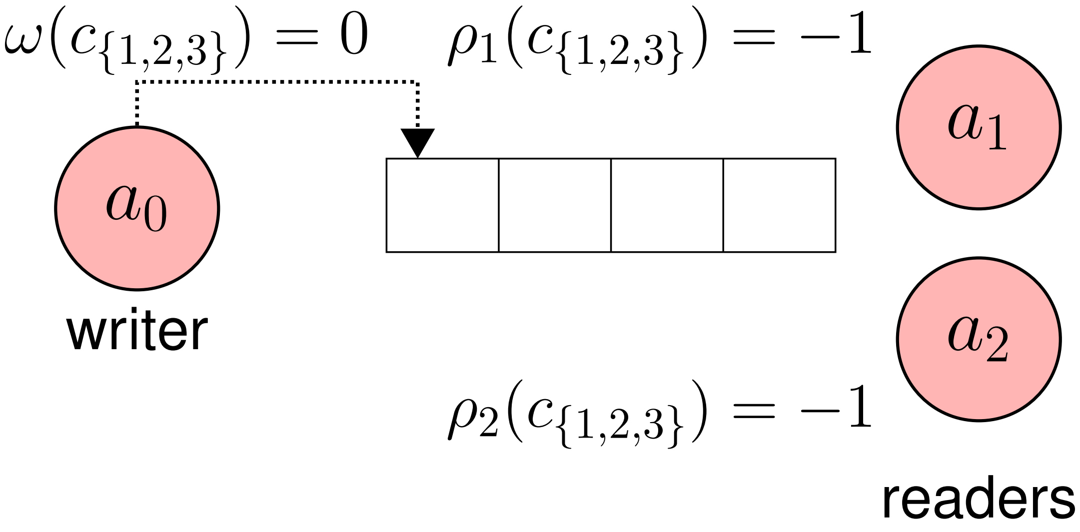
- After reading, the index $\rho_i(c_\mathsf{m})$ is updated:

$$\rho_i(c_\mathsf{m}) \leftarrow \begin{cases} -1 & \text{if } \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \kappa(a_{\mathsf{r}_i}) \\ (\rho_i(c_\mathsf{m}) + \kappa(a_{\mathsf{r}_i})) \bmod \gamma(c_\mathsf{m}) & \text{otherwise} \end{cases} \quad (3)$$

- After writing, the indices are updated as:

$$\forall_{1 \le i \le |A_\mathsf{r}|} \ \rho_i(c_\mathsf{m}) \leftarrow \begin{cases} \omega(c_\mathsf{m}) & \text{if } \rho_i(c_\mathsf{m}) = -1 \\ \rho_i(c_\mathsf{m}) & \text{otherwise} \end{cases} \quad (4)$$

$$\omega(c_\mathsf{m}) \leftarrow (\omega(c_\mathsf{m}) + \psi(a_\mathsf{w})) \bmod \gamma(c_\mathsf{m}) \quad (5)$$



By definition, an MRB c_{m} has **one writer** a_{w} and **multiple readers** $a_{\mathsf{r}_i} \in A_{\mathsf{r}} \subseteq A$

- MRB's state from a_{r_i} and a_{w} perspective:

$$\mathsf{T}(c_{\mathsf{m}}, a_{\mathsf{r}_i}) = \begin{cases} 0 & \text{if } \rho_i(c_{\mathsf{m}}) < 0 \\ ((\omega(c_{\mathsf{m}}) - \rho_i(c_{\mathsf{m}}) - 1) \bmod \gamma(c_{\mathsf{m}})) + 1 & \text{otherwise} \end{cases} \tag{1}$$

$$\mathsf{F}(c_{\mathsf{m}}) = \gamma(c_{\mathsf{m}}) - \max_{a_{\mathsf{r}_i} \in A_{\mathsf{r}}} \mathsf{T}(c_{\mathsf{m}}, a_{\mathsf{r}_i}) \tag{2}$$

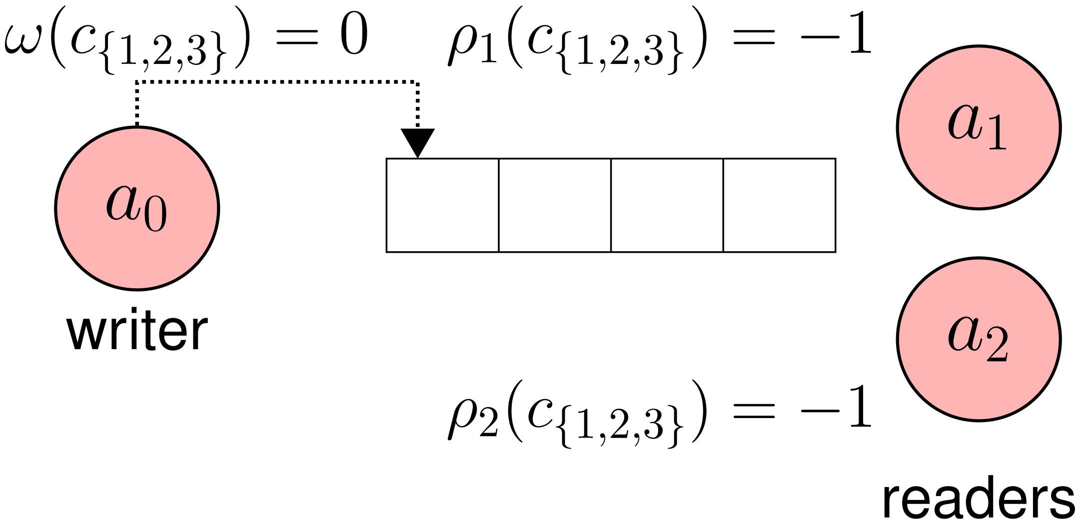
- After reading, the index $\rho_i(c_{\mathsf{m}})$ is updated:

$$\rho_i(c_{\mathsf{m}}) \leftarrow \begin{cases} -1 & \text{if } \mathsf{T}(c_{\mathsf{m}}, a_{\mathsf{r}_i}) = \kappa(a_{\mathsf{r}_i}) \\ (\rho_i(c_{\mathsf{m}}) + \kappa(a_{\mathsf{r}_i})) \bmod \gamma(c_{\mathsf{m}}) & \text{otherwise} \end{cases} \tag{3}$$

- After writing, the indices are updated as:

$$\forall_{1 \le i \le |A_{\mathsf{r}}|} \; \rho_i(c_{\mathsf{m}}) \leftarrow \begin{cases} \omega(c_{\mathsf{m}}) & \text{if } \rho_i(c_{\mathsf{m}}) = -1 \\ \rho_i(c_{\mathsf{m}}) & \text{otherwise} \end{cases} \tag{4}$$

$$\omega(c_{\mathsf{m}}) \leftarrow (\omega(c_{\mathsf{m}}) + \psi(a_{\mathsf{w}})) \bmod \gamma(c_{\mathsf{m}}) \tag{5}$$



Before firing $\langle a_1, a_1, a_1 \rangle$

Multi-Reader Buffer (MRB)

Semantics

By definition, an MRB c_m has **one writer a_w** and **multiple readers $a_{\mathsf{r}_i} \in A_\mathsf{r} \subseteq A$**

- MRB's state from a_{r_i} and a_w perspective:

$$\mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \begin{cases} 0 & \text{if } \rho_i(c_\mathsf{m}) < 0 \\ ((\omega(c_\mathsf{m}) - \rho_i(c_\mathsf{m}) - 1) \bmod \gamma(c_\mathsf{m})) + 1 & \text{otherwise} \end{cases} \tag{1}$$

$$\mathsf{F}(c_\mathsf{m}) = \gamma(c_\mathsf{m}) - \max_{a_{\mathsf{r}_i} \in A_\mathsf{r}} \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) \tag{2}$$

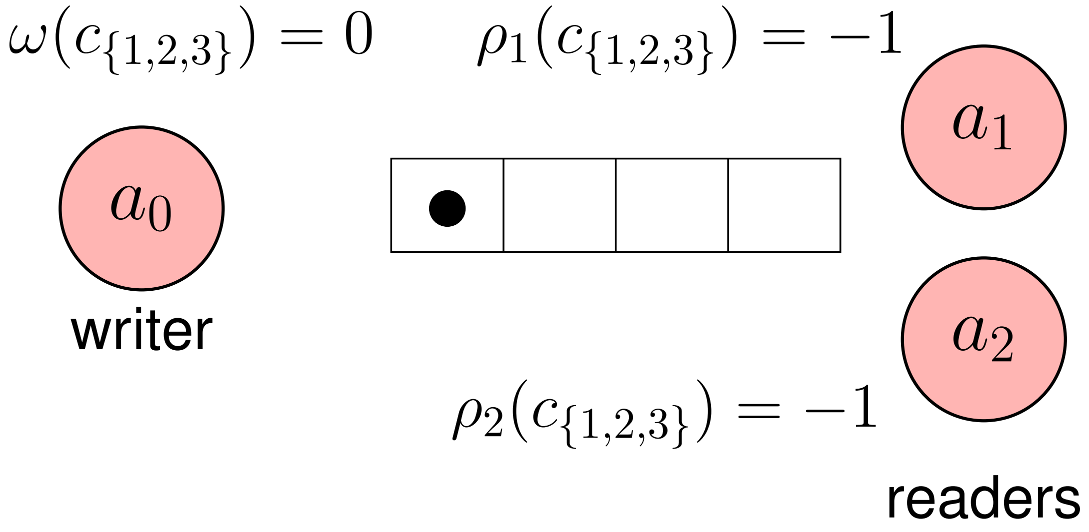
- After reading, the index $\rho_i(c_\mathsf{m})$ is updated:

$$\rho_i(c_\mathsf{m}) \leftarrow \begin{cases} -1 & \text{if } \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \kappa(a_{\mathsf{r}_i}) \\ (\rho_i(c_\mathsf{m}) + \kappa(a_{\mathsf{r}_i})) \bmod \gamma(c_\mathsf{m}) & \text{otherwise} \end{cases} \tag{3}$$

- After writing, the indices are updated as:

$$\underset{1 \le i \le |A_\mathsf{r}|}{\forall} \rho_i(c_\mathsf{m}) \leftarrow \begin{cases} \omega(c_\mathsf{m}) & \text{if } \rho_i(c_\mathsf{m}) = -1 \\ \rho_i(c_\mathsf{m}) & \text{otherwise} \end{cases} \tag{4}$$

$$\omega(c_\mathsf{m}) \leftarrow (\omega(c_\mathsf{m}) + \psi(a_\mathsf{w})) \bmod \gamma(c_\mathsf{m}) \tag{5}$$



Multi-Reader Buffer (MRB)

Semantics

By definition, an MRB $c_{\mathbf{m}}$ has **one writer** $a_{\mathbf{w}}$ and **multiple readers** $a_{\mathbf{r}_i} \in A_{\mathbf{r}} \subseteq A$

- MRB's state from $a_{\mathbf{r}_i}$ and $a_{\mathbf{w}}$ perspective:

$$\mathsf{T}(c_{\mathbf{m}}, a_{\mathbf{r}_i}) = \begin{cases} 0 & \text{if } \rho_i(c_{\mathbf{m}}) < 0 \\ ((\omega(c_{\mathbf{m}}) - \rho_i(c_{\mathbf{m}}) - 1) \bmod \gamma(c_{\mathbf{m}})) + 1 & \text{otherwise} \end{cases} \quad (1)$$

$$\mathsf{F}(c_{\mathbf{m}}) = \gamma(c_{\mathbf{m}}) - \max_{a_{\mathbf{r}_i} \in A_{\mathbf{r}}} \mathsf{T}(c_{\mathbf{m}}, a_{\mathbf{r}_i}) \quad (2)$$

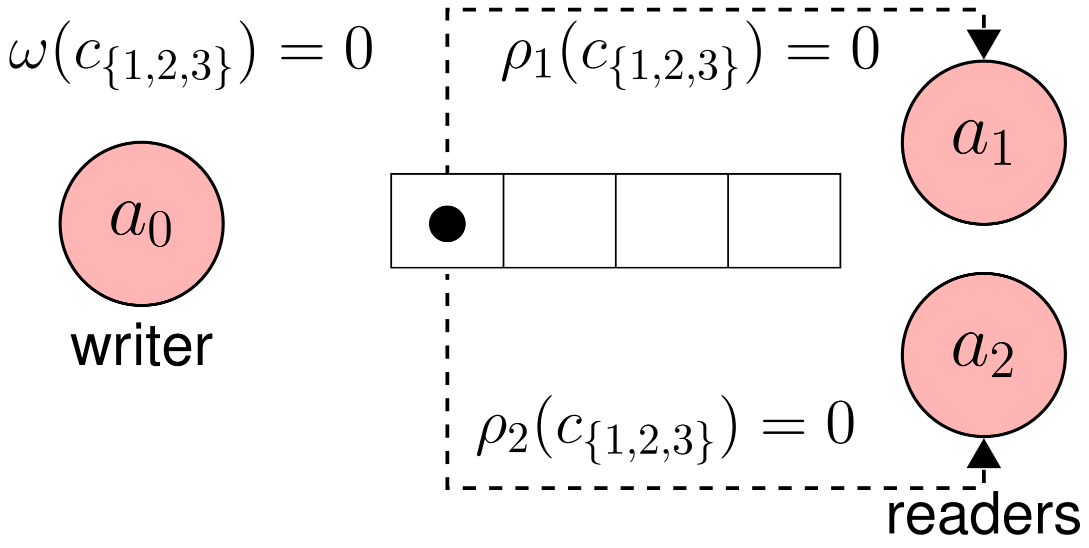
- After reading, the index $\rho_i(c_{\mathbf{m}})$ is updated:

$$\rho_i(c_{\mathbf{m}}) \leftarrow \begin{cases} -1 & \text{if } \mathsf{T}(c_{\mathbf{m}}, a_{\mathbf{r}_i}) = \kappa(a_{\mathbf{r}_i}) \\ (\rho_i(c_{\mathbf{m}}) + \kappa(a_{\mathbf{r}_i})) \bmod \gamma(c_{\mathbf{m}}) & \text{otherwise} \end{cases} \quad (3)$$

- After writing, the indices are updated as:

$$\forall_{1 \le i \le |A_{\mathbf{r}}|} \; \rho_i(c_{\mathbf{m}}) \leftarrow \begin{cases} \omega(c_{\mathbf{m}}) & \text{if } \rho_i(c_{\mathbf{m}}) = -1 \\ \rho_i(c_{\mathbf{m}}) & \text{otherwise} \end{cases} \quad (4)$$

$$\omega(c_{\mathbf{m}}) \leftarrow (\omega(c_{\mathbf{m}}) + \psi(a_{\mathbf{w}})) \bmod \gamma(c_{\mathbf{m}}) \quad (5)$$



Multi-Reader Buffer (MRB)

Semantics

By definition, an MRB c_m has **one writer** a_w and **multiple readers** $a_{\mathsf{r}_i} \in A_\mathsf{r} \subseteq A$

- MRB's state from a_{r_i} and a_w perspective:

$$\mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \begin{cases} 0 & \text{if } \rho_i(c_\mathsf{m}) < 0 \\ ((\omega(c_\mathsf{m}) - \rho_i(c_\mathsf{m}) - 1) \bmod \gamma(c_\mathsf{m})) + 1 & \text{otherwise} \end{cases} \quad (1)$$

$$\mathsf{F}(c_\mathsf{m}) = \gamma(c_\mathsf{m}) - \max_{a_{\mathsf{r}_i} \in A_\mathsf{r}} \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) \quad (2)$$

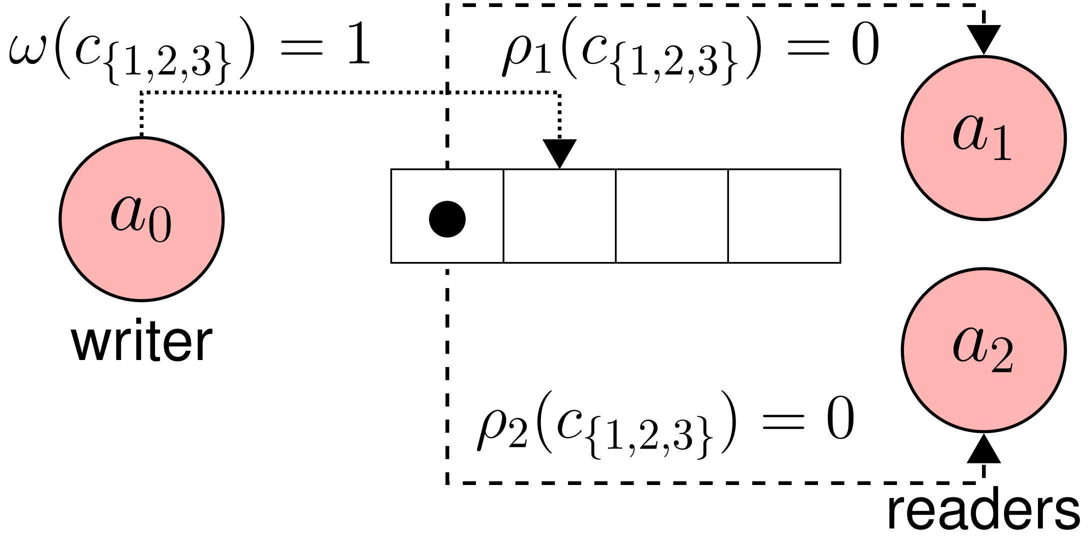
- After reading, the index $\rho_i(c_\mathsf{m})$ is updated:

$$\rho_i(c_\mathsf{m}) \leftarrow \begin{cases} -1 & \text{if } \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \kappa(a_{\mathsf{r}_i}) \\ (\rho_i(c_\mathsf{m}) + \kappa(a_{\mathsf{r}_i})) \bmod \gamma(c_\mathsf{m}) & \text{otherwise} \end{cases} \quad (3)$$

- After writing, the indices are updated as:

$$\forall_{1 \le i \le |A_\mathsf{r}|} \; \rho_i(c_\mathsf{m}) \leftarrow \begin{cases} \omega(c_\mathsf{m}) & \text{if } \rho_i(c_\mathsf{m}) = -1 \\ \rho_i(c_\mathsf{m}) & \text{otherwise} \end{cases} \quad (4)$$

$$\omega(c_\mathsf{m}) \leftarrow (\omega(c_\mathsf{m}) + \psi(a_\mathsf{w})) \bmod \gamma(c_\mathsf{m}) \quad (5)$$



Multi-Reader Buffer (MRB)

Semantics

By definition, an MRB c_{m} has **one writer** a_{w} and **multiple readers** $a_{\mathsf{r}_i} \in A_{\mathsf{r}} \subseteq A$

- MRB's state from a_{r_i} and a_{w} perspective:

$$\mathsf{T}(c_{\mathsf{m}}, a_{\mathsf{r}_i}) = \begin{cases} 0 & \text{if } \rho_i(c_{\mathsf{m}}) < 0 \\ ((\omega(c_{\mathsf{m}}) - \rho_i(c_{\mathsf{m}}) - 1) \bmod \gamma(c_{\mathsf{m}})) + 1 & \text{otherwise} \end{cases} \quad (1)$$

$$\mathsf{F}(c_{\mathsf{m}}) = \gamma(c_{\mathsf{m}}) - \max_{a_{\mathsf{r}_i} \in A_{\mathsf{r}}} \mathsf{T}(c_{\mathsf{m}}, a_{\mathsf{r}_i}) \quad (2)$$

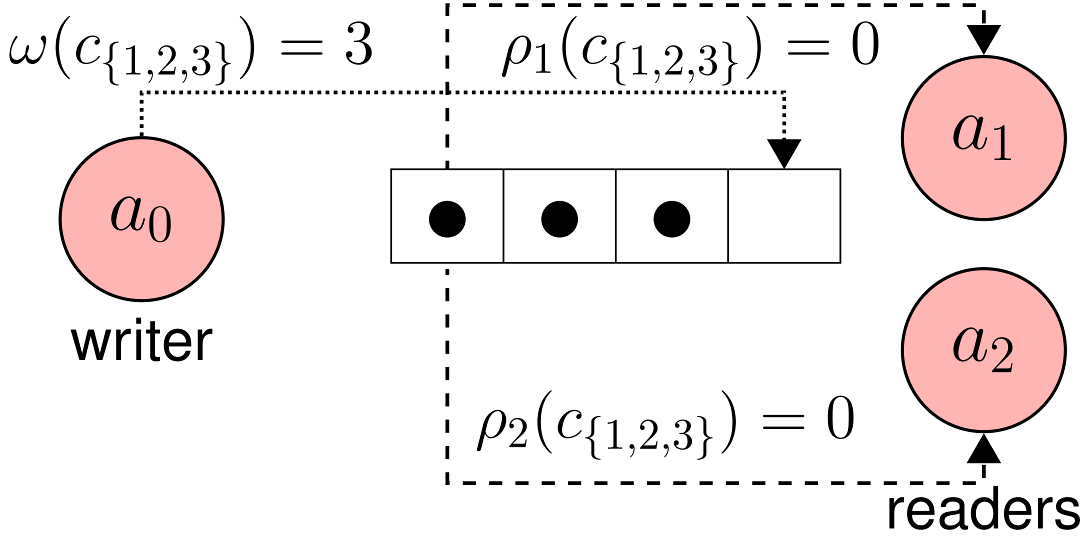
- After reading, the index $\rho_i(c_{\mathsf{m}})$ is updated:

$$\rho_i(c_{\mathsf{m}}) \leftarrow \begin{cases} -1 & \text{if } \mathsf{T}(c_{\mathsf{m}}, a_{\mathsf{r}_i}) = \kappa(a_{\mathsf{r}_i}) \\ (\rho_i(c_{\mathsf{m}}) + \kappa(a_{\mathsf{r}_i})) \bmod \gamma(c_{\mathsf{m}}) & \text{otherwise} \end{cases} \quad (3)$$

- After writing, the indices are updated as:

$$\forall_{1 \le i \le |A_{\mathsf{r}}|} \ \rho_i(c_{\mathsf{m}}) \leftarrow \begin{cases} \omega(c_{\mathsf{m}}) & \text{if } \rho_i(c_{\mathsf{m}}) = -1 \\ \rho_i(c_{\mathsf{m}}) & \text{otherwise} \end{cases} \quad (4)$$

$$\omega(c_{\mathsf{m}}) \leftarrow (\omega(c_{\mathsf{m}}) + \psi(a_{\mathsf{w}})) \bmod \gamma(c_{\mathsf{m}}) \quad (5)$$



After firing $\langle a_1, a_1, a_1 \rangle$

Multi-Reader Buffer (MRB)

Semantics

By definition, an MRB c_m has **one writer** a_w and **multiple readers** $a_{\mathsf{r}_i} \in A_\mathsf{r} \subseteq A$

- MRB's state from a_{r_i} and a_w perspective:

$$\mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \begin{cases} 0 & \text{if } \rho_i(c_\mathsf{m}) < 0 \\ ((\omega(c_\mathsf{m}) - \rho_i(c_\mathsf{m}) - 1) \bmod \gamma(c_\mathsf{m})) + 1 & \text{otherwise} \end{cases} \tag{1}$$

$$\mathsf{F}(c_\mathsf{m}) = \gamma(c_\mathsf{m}) - \max_{a_{\mathsf{r}_i} \in A_\mathsf{r}} \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) \tag{2}$$

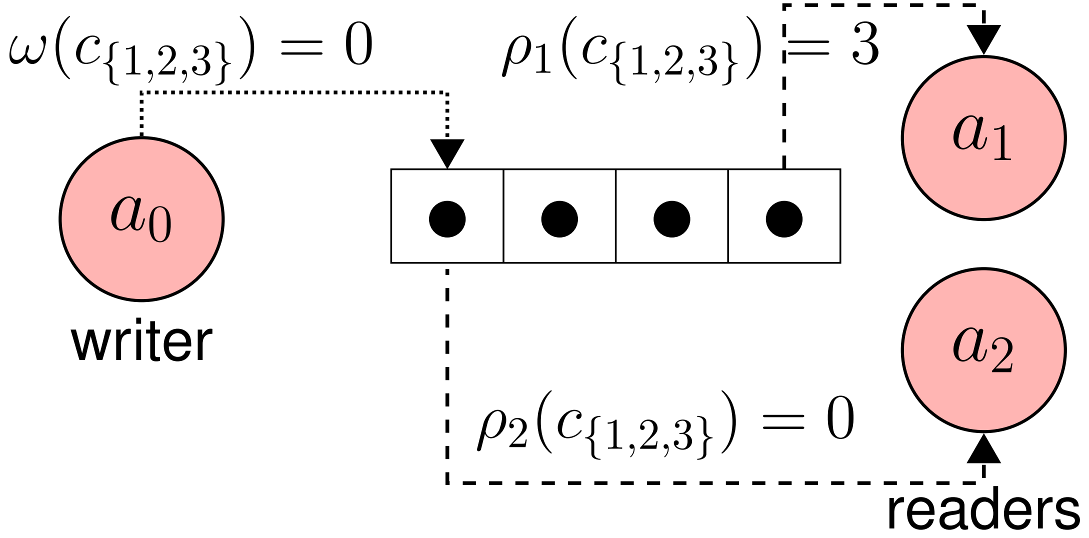
- After reading, the index $\rho_i(c_\mathsf{m})$ is updated:

$$\rho_i(c_\mathsf{m}) \leftarrow \begin{cases} -1 & \text{if } \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \kappa(a_{\mathsf{r}_i}) \\ (\rho_i(c_\mathsf{m}) + \kappa(a_{\mathsf{r}_i})) \bmod \gamma(c_\mathsf{m}) & \text{otherwise} \end{cases} \tag{3}$$

- After writing, the indices are updated as:

$$\forall_{1 \le i \le |A_\mathsf{r}|} \; \rho_i(c_\mathsf{m}) \leftarrow \begin{cases} \omega(c_\mathsf{m}) & \text{if } \rho_i(c_\mathsf{m}) = -1 \\ \rho_i(c_\mathsf{m}) & \text{otherwise} \end{cases} \tag{4}$$

$$\omega(c_\mathsf{m}) \leftarrow (\omega(c_\mathsf{m}) + \psi(a_\mathsf{w})) \bmod \gamma(c_\mathsf{m}) \tag{5}$$



After firing $\langle a_3, a_3, a_3, a_1 \rangle$

Multi-Reader Buffer (MRB)

Semantics

By definition, an MRB c_m has **one writer** a_w and **multiple readers** $a_{\mathsf{r}_i} \in A_\mathsf{r} \subseteq A$

- MRB's state from a_{r_i} and a_w perspective:

$$\mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \begin{cases} 0 & \text{if } \rho_i(c_\mathsf{m}) < 0 \\ ((\omega(c_\mathsf{m}) - \rho_i(c_\mathsf{m}) - 1) \bmod \gamma(c_\mathsf{m})) + 1 & \text{otherwise} \end{cases} \quad (1)$$

$$\mathsf{F}(c_\mathsf{m}) = \gamma(c_\mathsf{m}) - \max_{a_{\mathsf{r}_i} \in A_\mathsf{r}} \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) \quad (2)$$

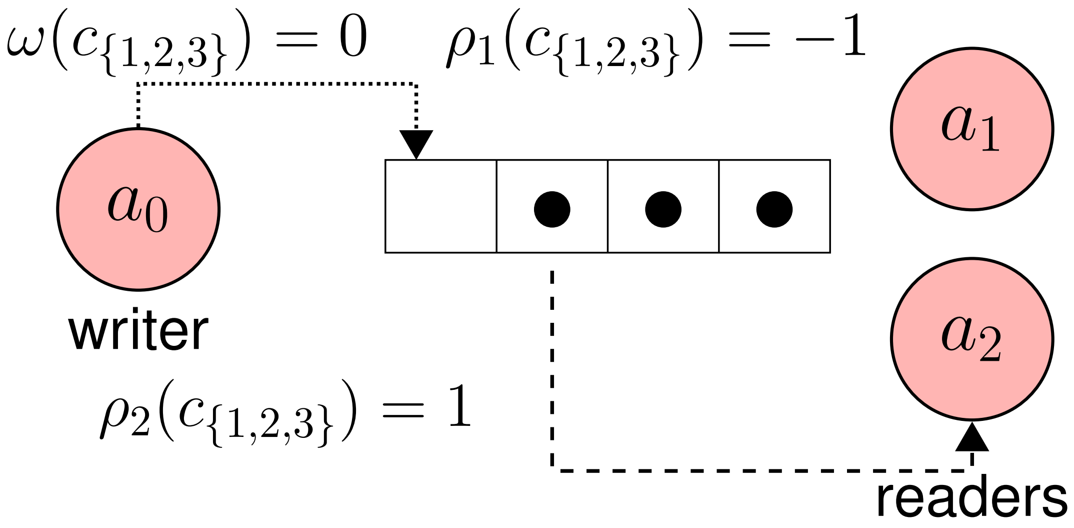
- After reading, the index $\rho_i(c_\mathsf{m})$ is updated:

$$\rho_i(c_\mathsf{m}) \leftarrow \begin{cases} -1 & \text{if } \mathsf{T}(c_\mathsf{m}, a_{\mathsf{r}_i}) = \kappa(a_{\mathsf{r}_i}) \\ (\rho_i(c_\mathsf{m}) + \kappa(a_{\mathsf{r}_i})) \bmod \gamma(c_\mathsf{m}) & \text{otherwise} \end{cases} \quad (3)$$

- After writing, the indices are updated as:

$$\forall_{1 \le i \le |A_\mathsf{r}|} \; \rho_i(c_\mathsf{m}) \leftarrow \begin{cases} \omega(c_\mathsf{m}) & \text{if } \rho_i(c_\mathsf{m}) = -1 \\ \rho_i(c_\mathsf{m}) & \text{otherwise} \end{cases} \quad (4)$$

$$\omega(c_\mathsf{m}) \leftarrow (\omega(c_\mathsf{m}) + \psi(a_\mathsf{w})) \bmod \gamma(c_\mathsf{m}) \quad (5)$$



After firing $\langle a_4, a_3 \rangle$

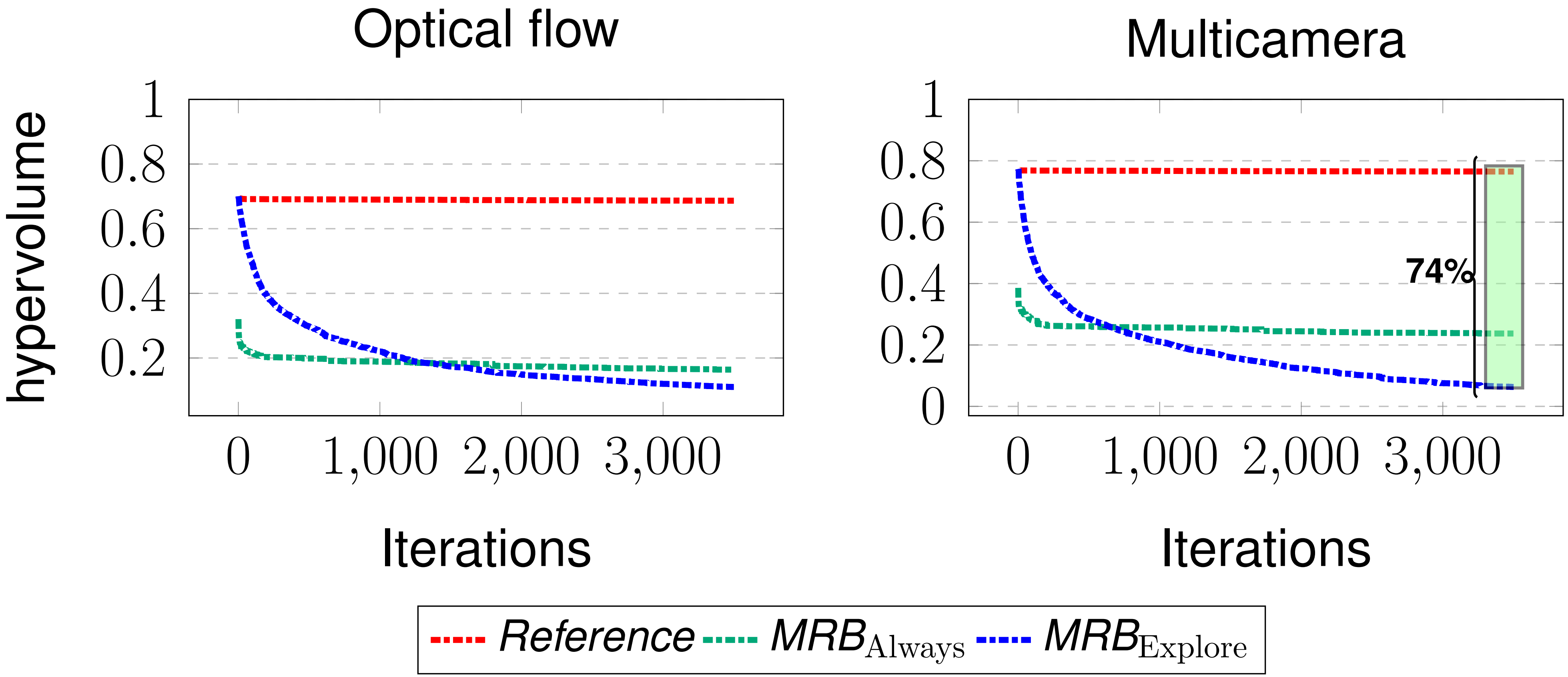
Experimental Setup

Design Space Exploration

We compare three approaches that perform the **multi-objective optimization of throughput, memory footprint and # of allocated cores**

- *Reference* exploring **only the mappings**
- *MRB*$_{\text{Always}}$ exploring the mappings and **merging all the multicast actors**
- *MRB*$_{\text{Explore}}$ exploring the mappings and **exploring the merging of multicast actors**

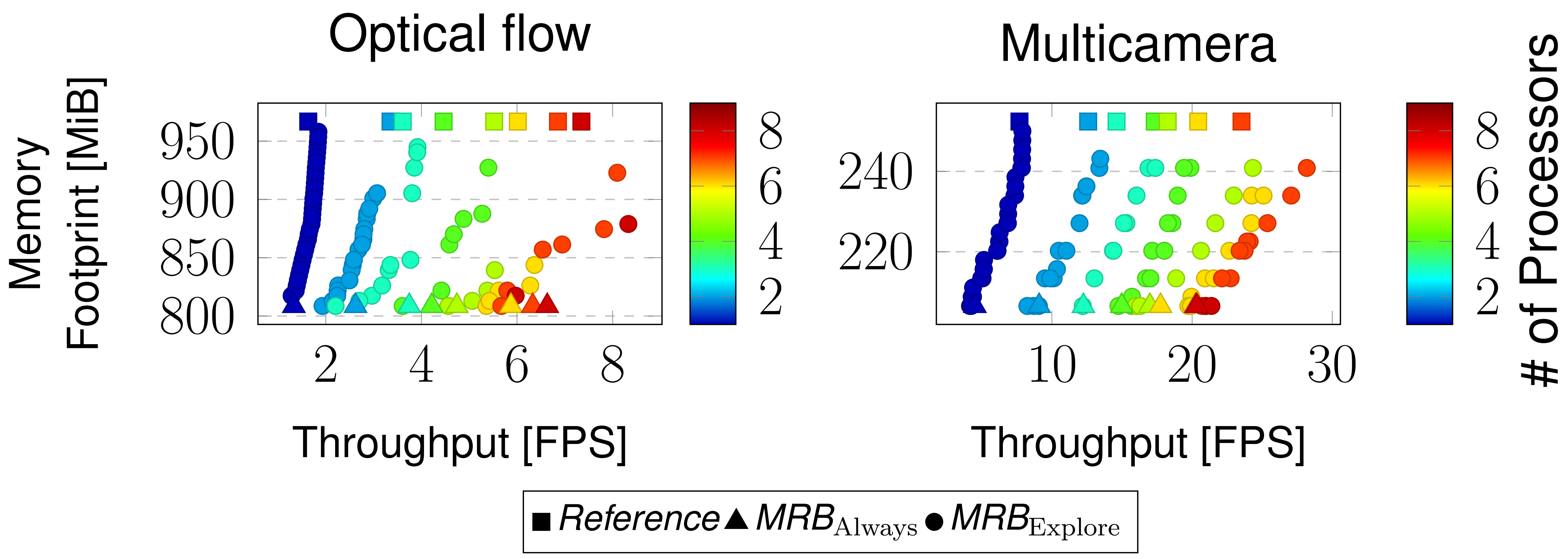
Application	# of instances	$\|A\|$	$\|C\|$	# of multicast	$M_F($*Reference*$)$ [MiB]
Optical flow	4	89	112	15	996.8
Stitching multicamera	2	123	226	46	252.4

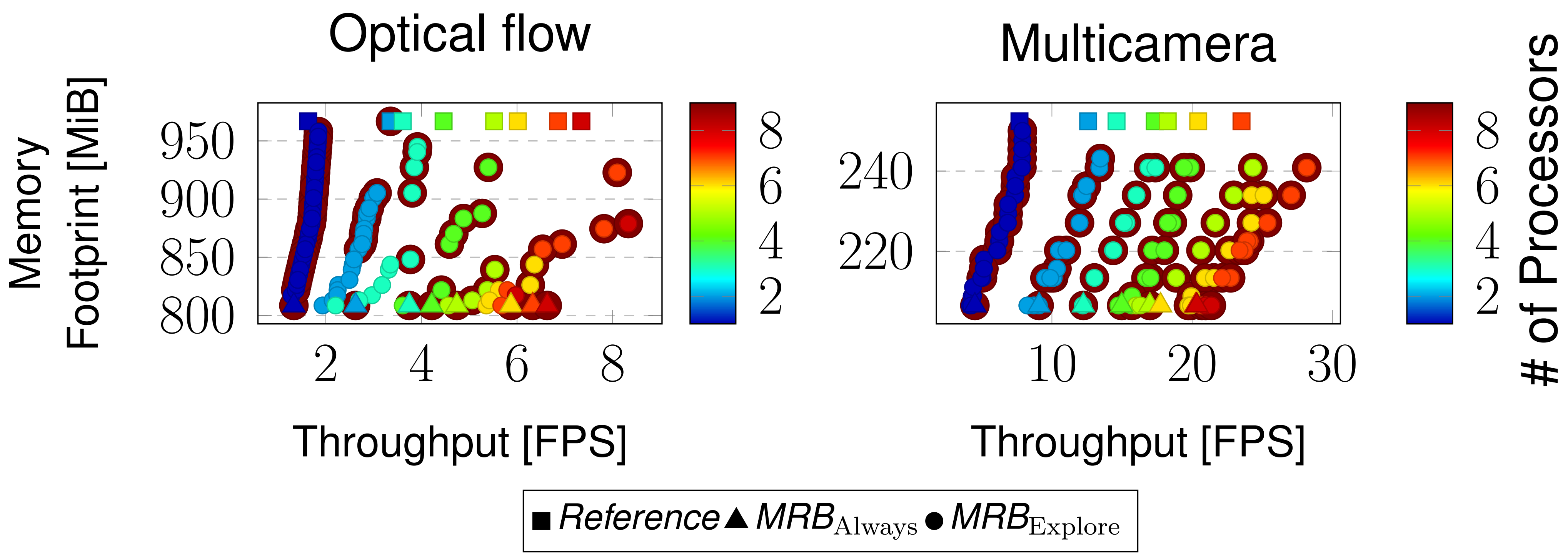


A. P. Guerreiro, C. M. Fonseca, and L. Paquete. “The Hypervolume Indicator: Computational Problems and Algorithms”. In: *ACM Comput. Surv.* 54.6 (July 2021)

Results

Pareto-Front




Optical flow
Multicamera
Memory Footprint [MiB]
950
900
850
800
240
220
2
4
6
8
10
20
30
Throughput [FPS]
Throughput [FPS]
8
6
4
2
# of Processors
Reference
MRB_{Always}
MRB_{Explore}

Summary

- Introduction of the **concept of Multi-Reader Buffers (MRBs)** as a memory-efficient implementation of multi-cast actors and their replacement as a graph transformation
- Rather than replicating produced tokens for all readers, an **MRB stores only one copy of data for all readers**
- MRBs provide **minimal buffer implementations** that are obtained by replacing all multi-cast actors in an application with MRB
- We proposed a DSE approach to **explore the space of selective MRB replacements** which delivers significant improvements in the throughput of applications

Thanks for listening.

Any questions?