ReCoNodes – Optimization Methods and Platform Design for Reconfigurable Hardware Systems

Prof. Dr. Sándor Fekete

Dr. **Tom Kamphans**

Dipl.-Math. Jan van der Veen

Dipl.-Math.oec. Nils Schweer

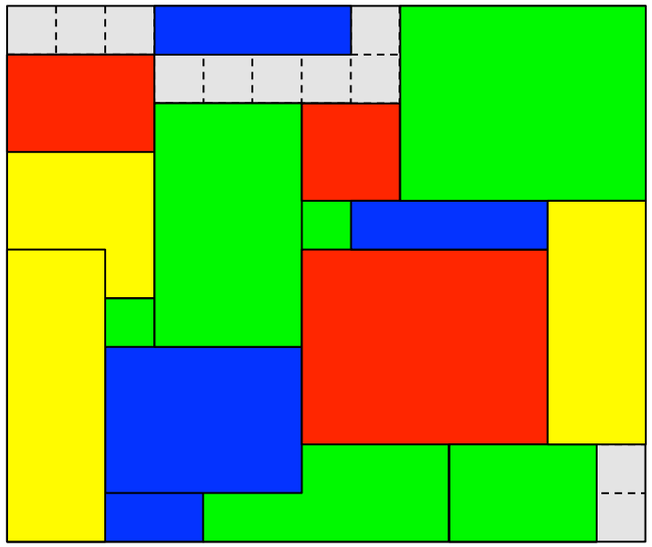
Prof. Dr.-Ing. Jürgen Teich

Dipl.-Inf. Josef Angermeier

Dipl.-Ing. Mateusz Majer

ReCoNodes Part II

Optimization Methods for Module Scheduling and Placement on Reconfigurable Hardware Devices

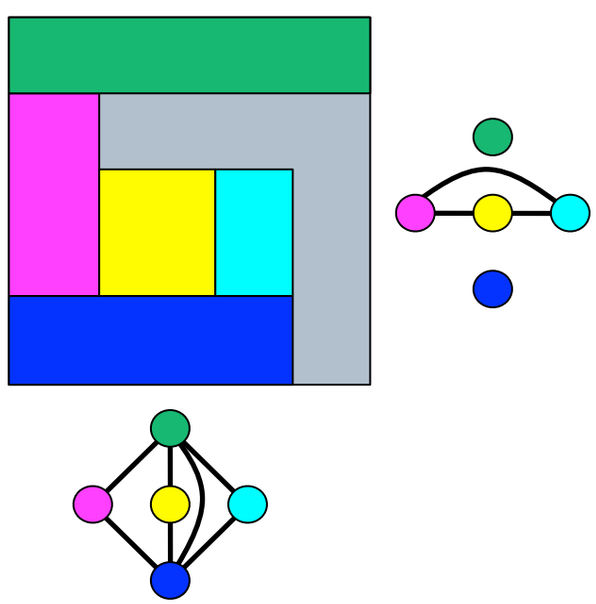


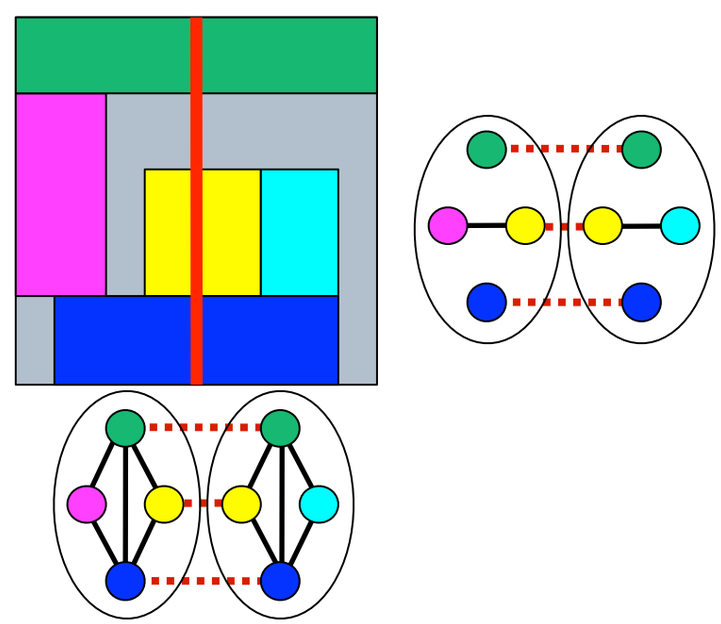
Packing

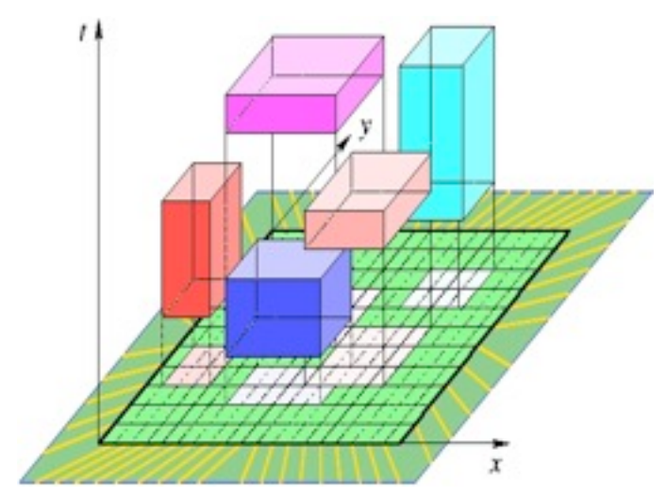
Placing modules onto an FPGA

Orthogonal Placement

- Placement of orthogonal modules ➜ intervall graphs
- Find optimal placement by enumeration of valid interval graphs
- Works also with
  - heterogenities (RAM, DSPs, ...)
  - higher dimensional packings

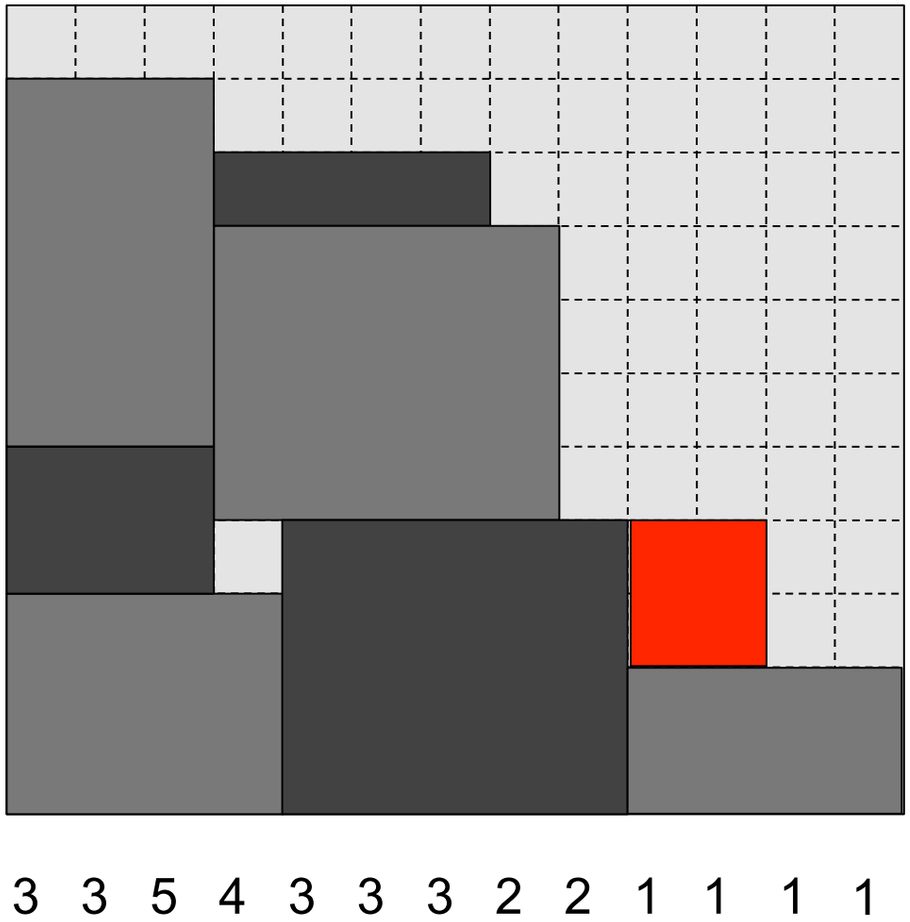






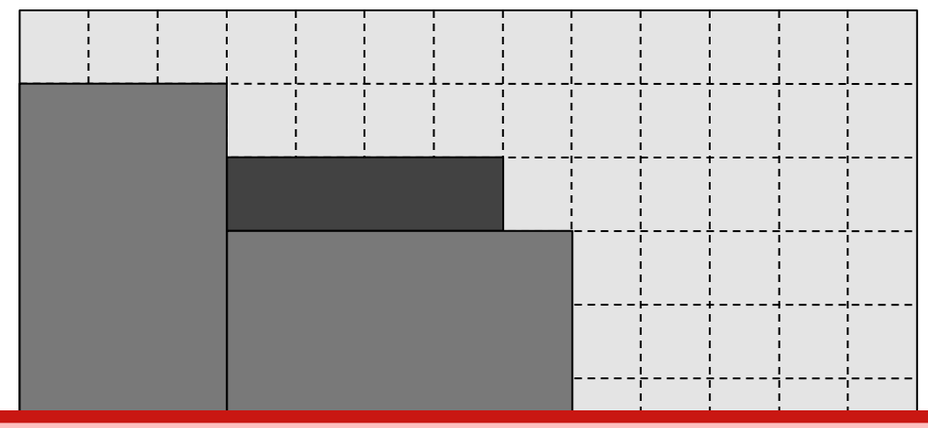
Orthogonal Placement: Online

- Least Interference Fit:
  Place modules in those columns, that are used by as few other modules as possible



Orthogonal Placement: Online

- Least Interference Fit:
  Place modules in those columns, that are used by as few other modules as possible



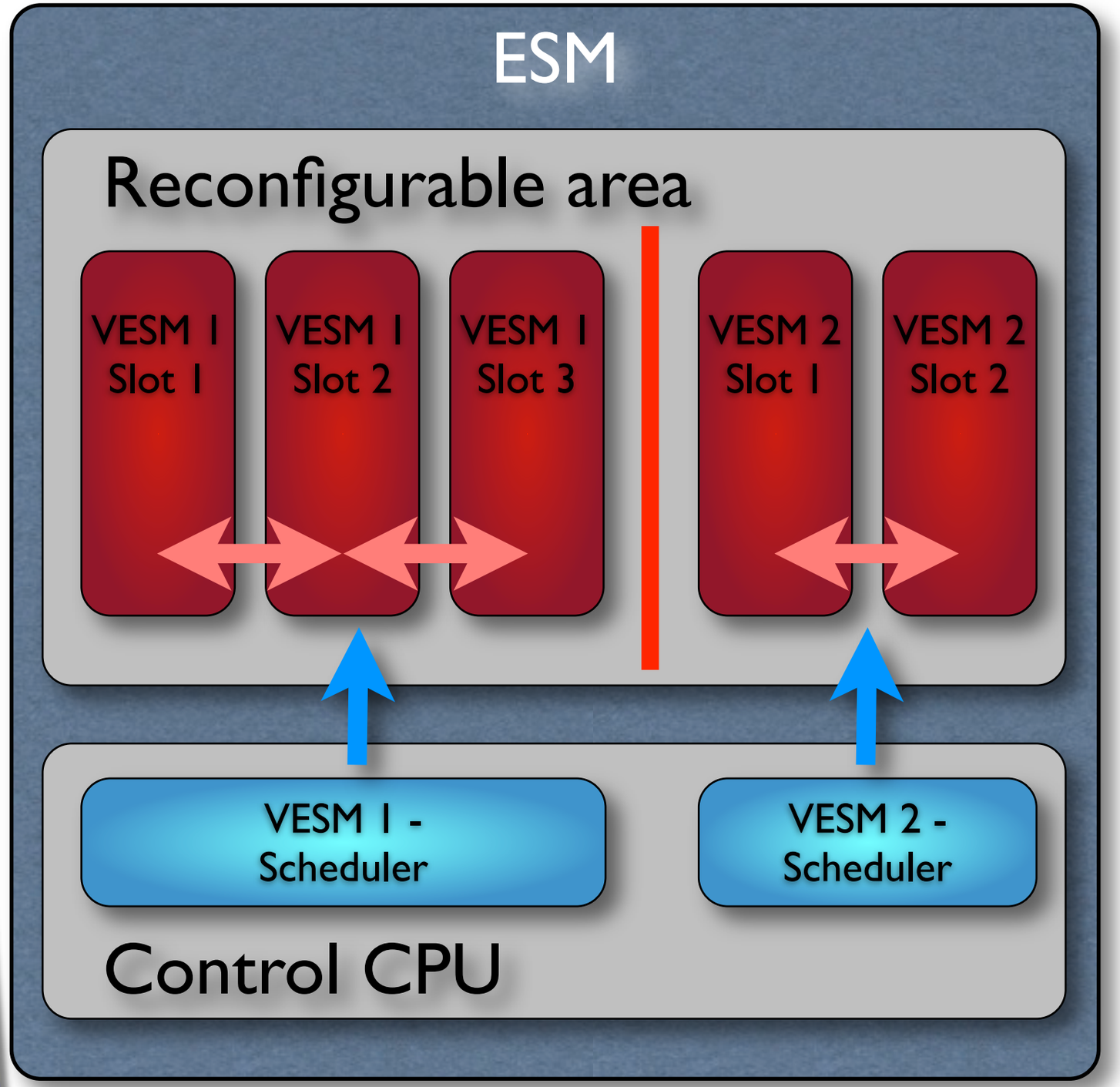
Theorem

If all rectangles have the same width, then *LIF* is 1-competitive

3 3 5 4 3 3 3 2 2 1 1 1 1

Virtual Areas

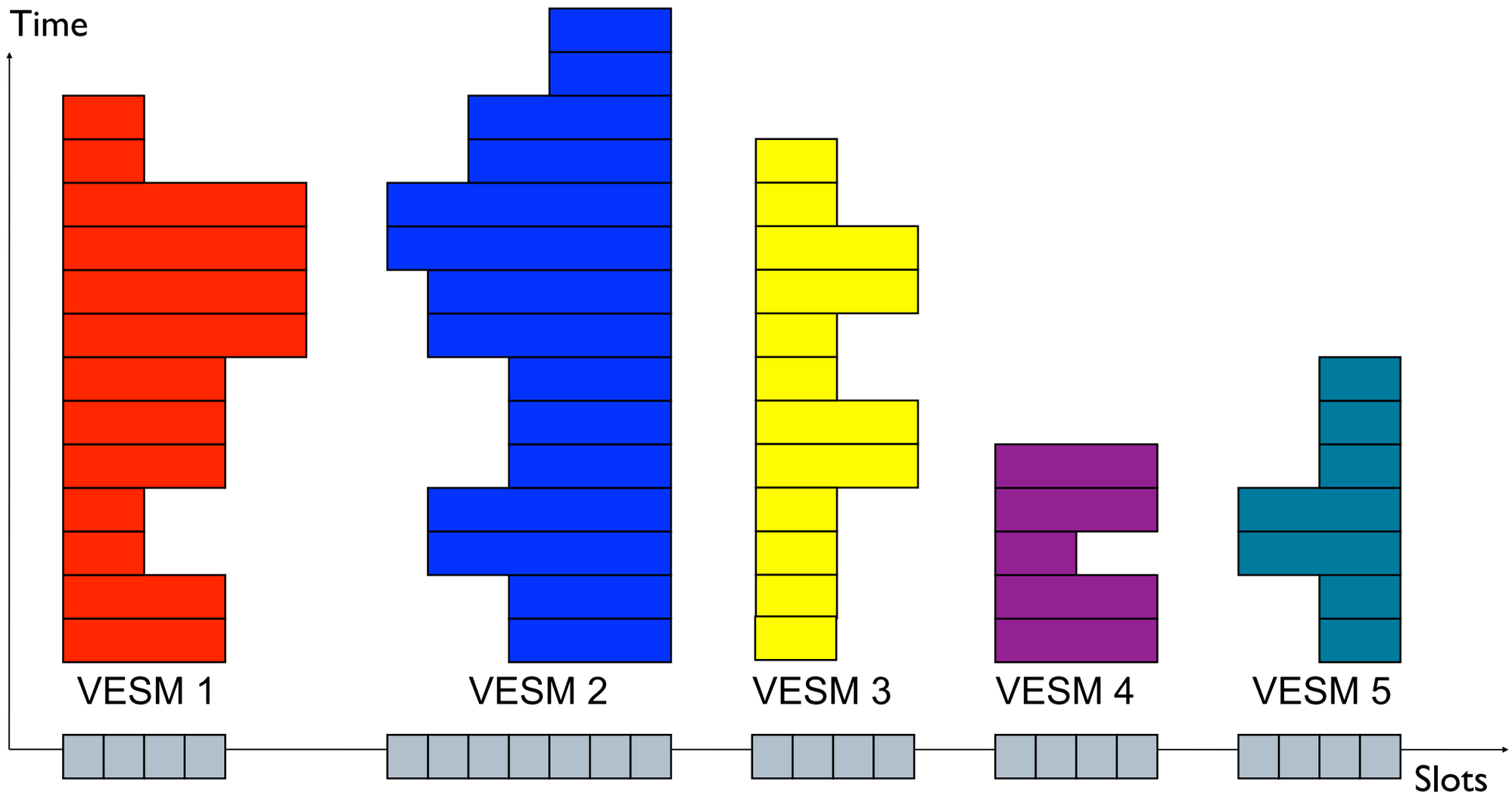
- Divide an ESM into several virtual areas
- Advantages
  - Restrict resources
  - Protect applications
  - Different schedulers for each VESM



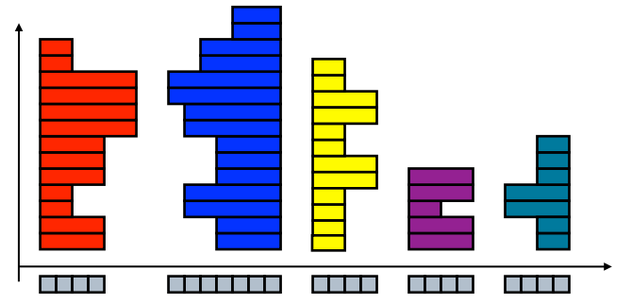
Task: Pack modules with time-varying resource requests

Virtual Areas: Example

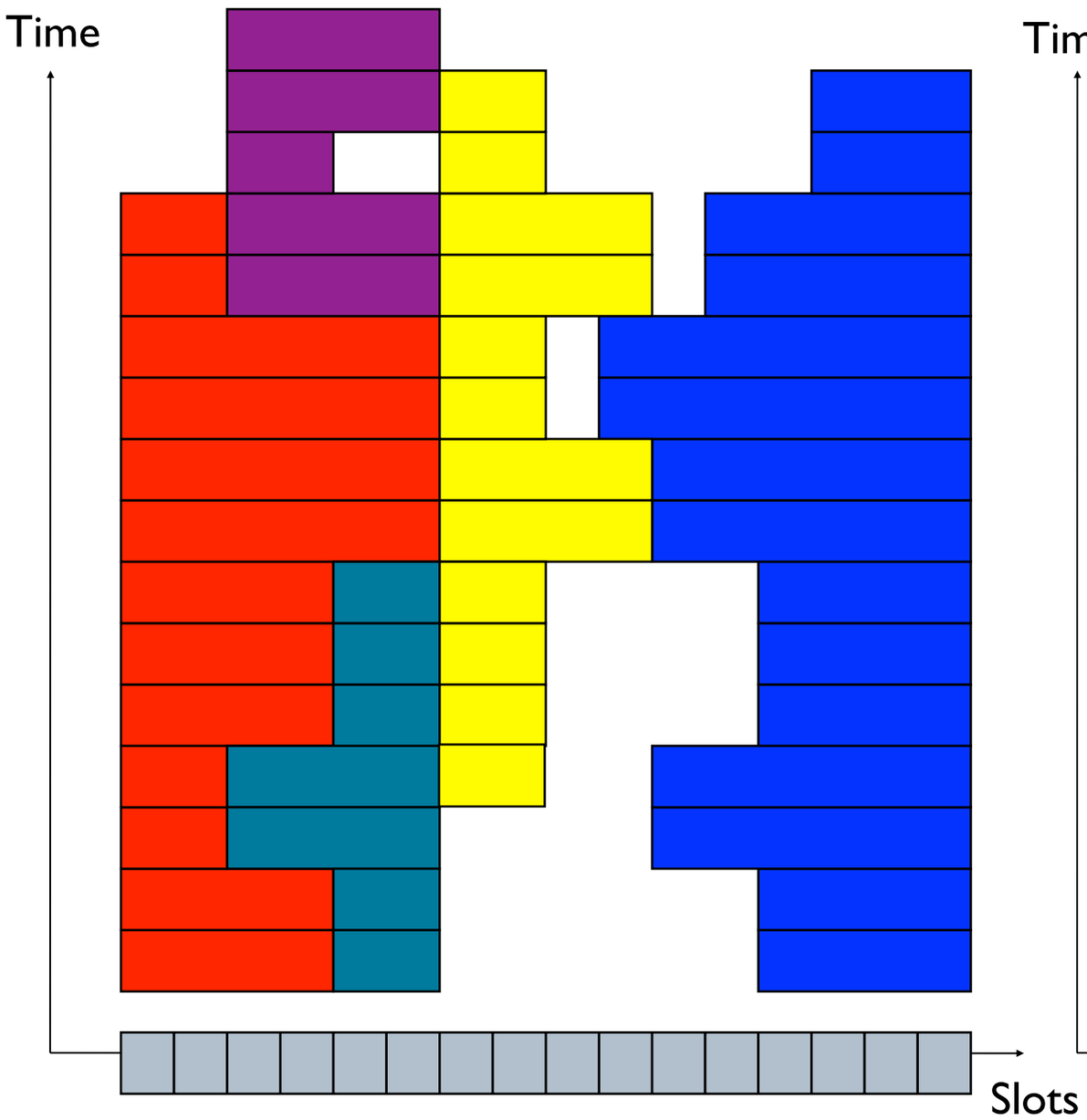
Example: Schedule for 5 VESMs on 16 Slots



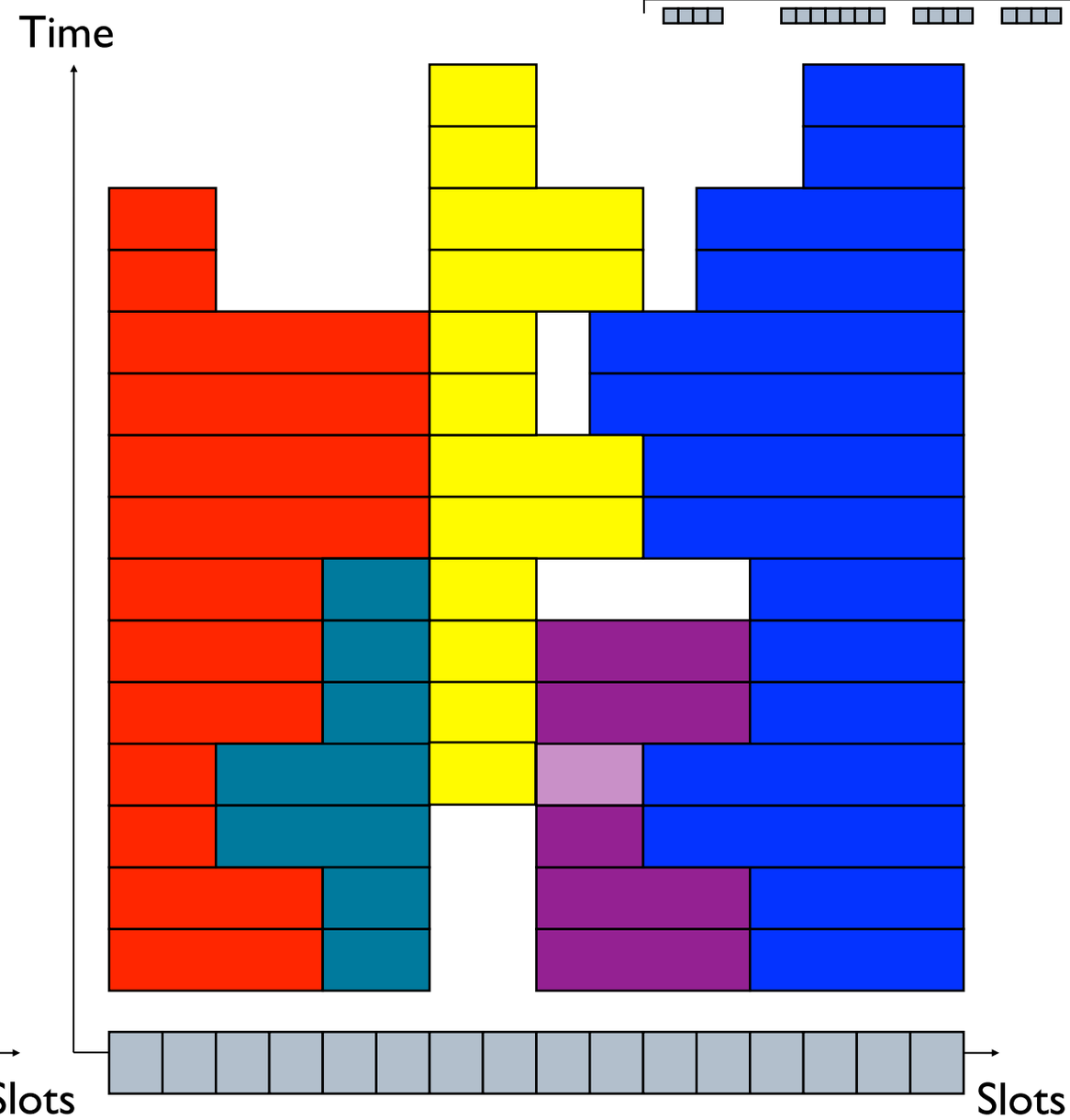
Virtual Areas: Delaying Requests



Without delaying



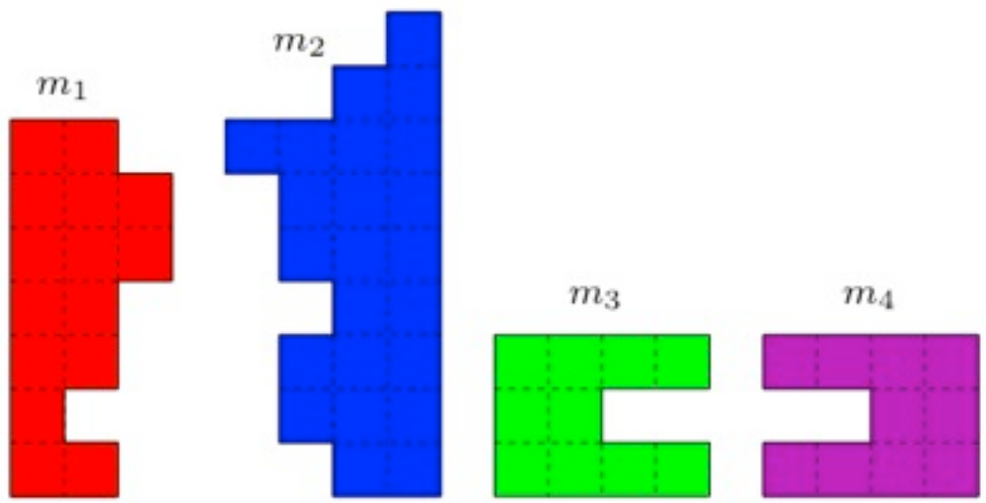
With delaying



FPGAtris: offline

Requests known a priori

- Solvable using an ILP
- Example:



- 6120 Constraints
- 2537 Variables
- 6h with CPLEX (3.2 GHz Xeon)

Variables

Slot assignment variables (x_{si}) module m_i is scheduled in slot s
Time assignment variables (y_{tij}) request (i, j) is scheduled at time t
Occupancy variables (z_{stij}) keep track of occupied slots
Usage variables (u_t) indicates which times steps are used

Constraints

Assignment Constraints $\sum_{s=1}^{N} x_{si} = 1 \qquad \forall i = 1, \ldots, M\,,$ (1)

$\sum_{t=1}^{T} y_{tij} = 1 \qquad \forall i = 1, \ldots, M,\ j = 1, \ldots, \ell_i\,.$ (2)

Boundary Constraints $\forall i, j,\ s = s_{\text{low}}, \ldots, s_{\text{up}} : \quad x_{si} = 0$ (3)

Order Constraints $\sum_{t=1}^{T} t\, y_{tij} - \sum_{t=1}^{T} t\, y_{tij-1} > 0 \quad \forall i, j > 0\,.$ (4)

Occupancy Constraints $\forall i = 1, \ldots, M,\ j = 1, \ldots, \ell_i,\ s = 1, \ldots, N,\ t = 1, \ldots, T,$
$s' = s_{\text{low}}, \ldots, s_{\text{up}} : x_{si} + y_{tij} - z_{s'tij} \leq 1$ (5)

Exclusive Constraints $\forall t = 1, \ldots, T,\ s = 1, \ldots, N : \sum_{i=1}^{M} \sum_{j=1}^{\ell_i} z_{stij} \leq 1$ (6)

Delay Constraints $\forall i = 1, \ldots M,\ j = 1, \ldots, \ell_i - 1,\ s = 1, \ldots, N,$
$t = 1, \ldots, T-1 : z_{stij} - z_{s(t+1)ij} - y_{(t+1)i(j+1)} \leq 0$ (7)

Usage Constraints $\forall t = 1, \ldots T,\ i = 1, \ldots, M,\ j = 1, \ldots, \ell_i : u_t - y_{tij} \geq 0$ (8)
$\forall t = 2, \ldots, T : u_{t-1} - u_t \geq 0\,.$ (9)

Objective Function

$$\min \sum_{i=1}^{T} i u_i$$

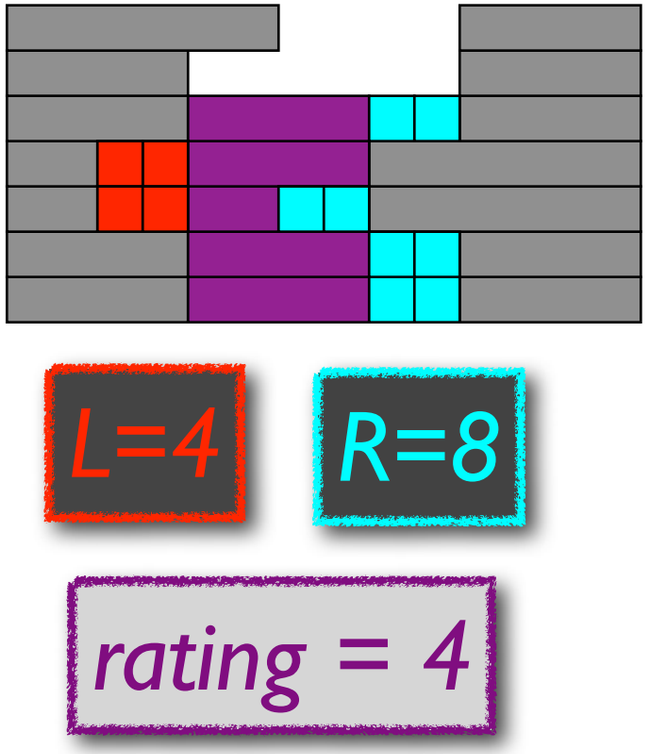
subject to Eq. (1)–(9)
$x_{si} \in \{0, 1\}, y_{tij} \in \{0, 1\}, z_{stij} \in \{0, 1\}, u_t \in \{0, 1\}$

FPGAtris: Simple Heuristics

- FirstFit
  - Choose the lowest, leftmost, overlapping-free position
- FirstFit with delays
  - Same as FF, but stretch modules

FPGAtris: Simple Heuristics

- BestFit
  - L (R) := unoccupied cells left (right) to placed module
  - rating := min{L, R}
  - take position with minimal rating
  - Quality of BestFit packing depends on insertion order

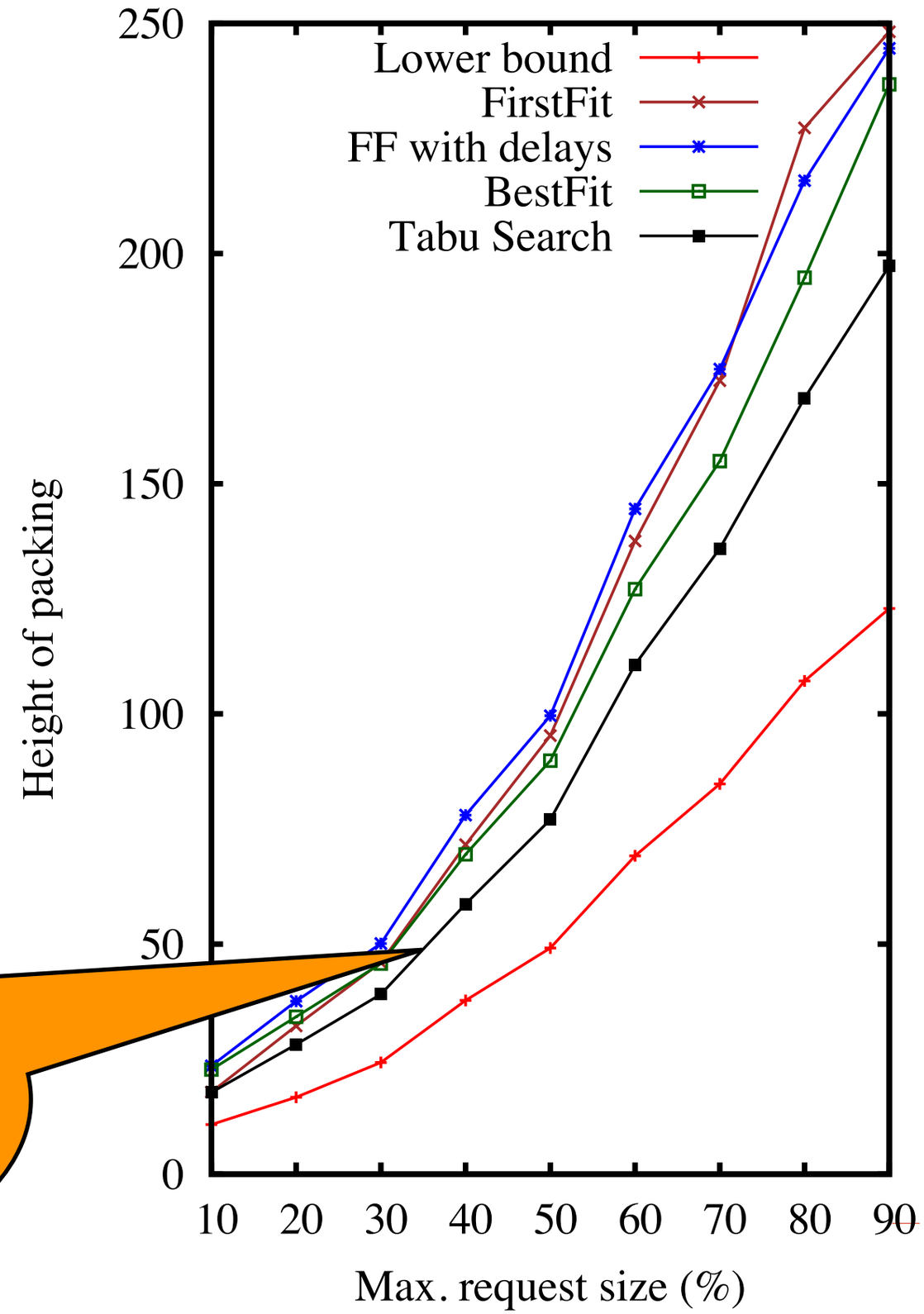


- TabuSearch
  - Try different insertion orders with BestFit

FPGAtris: Simple Heuristics

- Experiments with randomly generated input
- Lower bound: total area / width

$$\mathrm{LB} = \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{\ell_i} r_{ij}$$

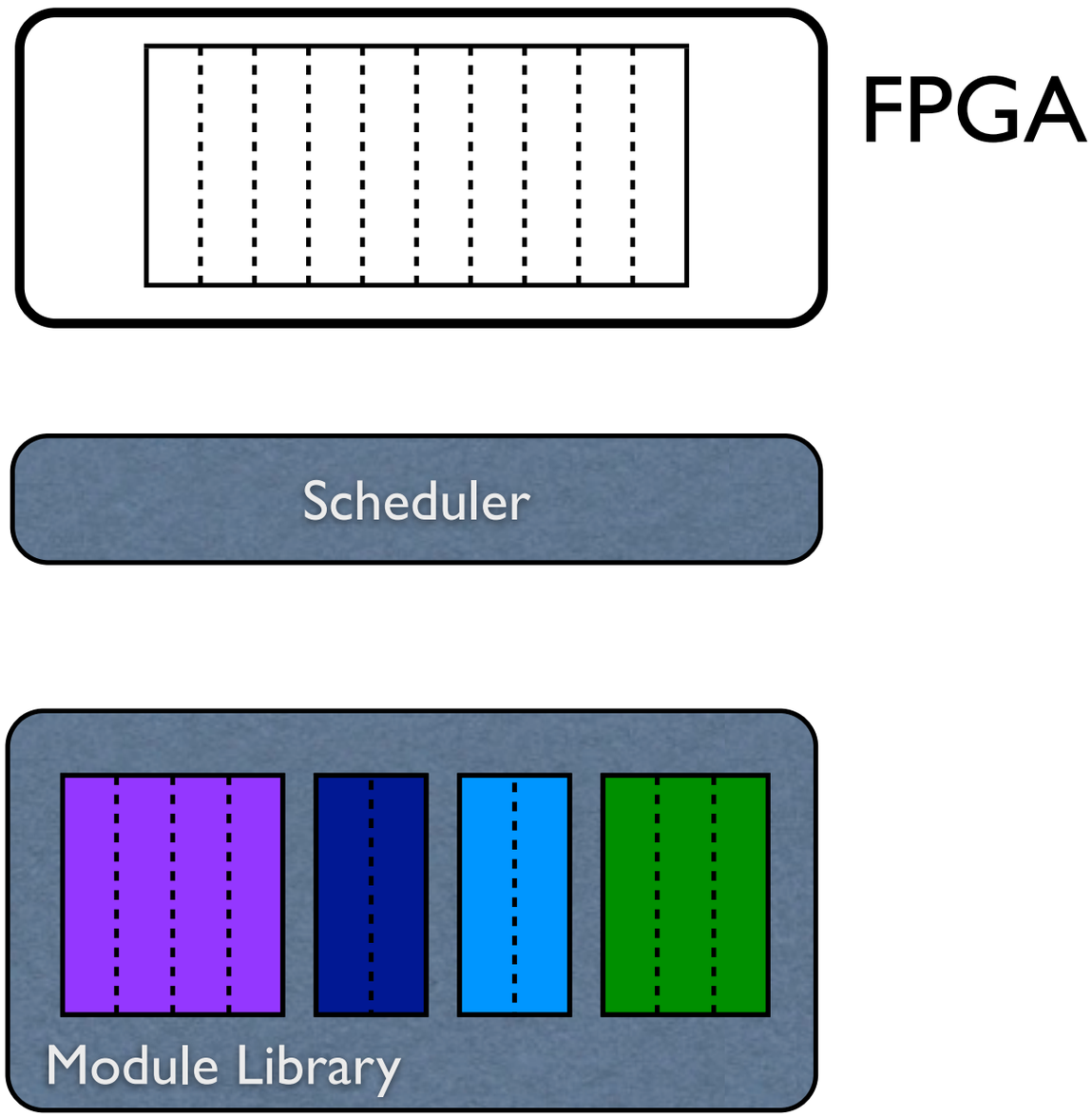


TabuSearch outperforms others

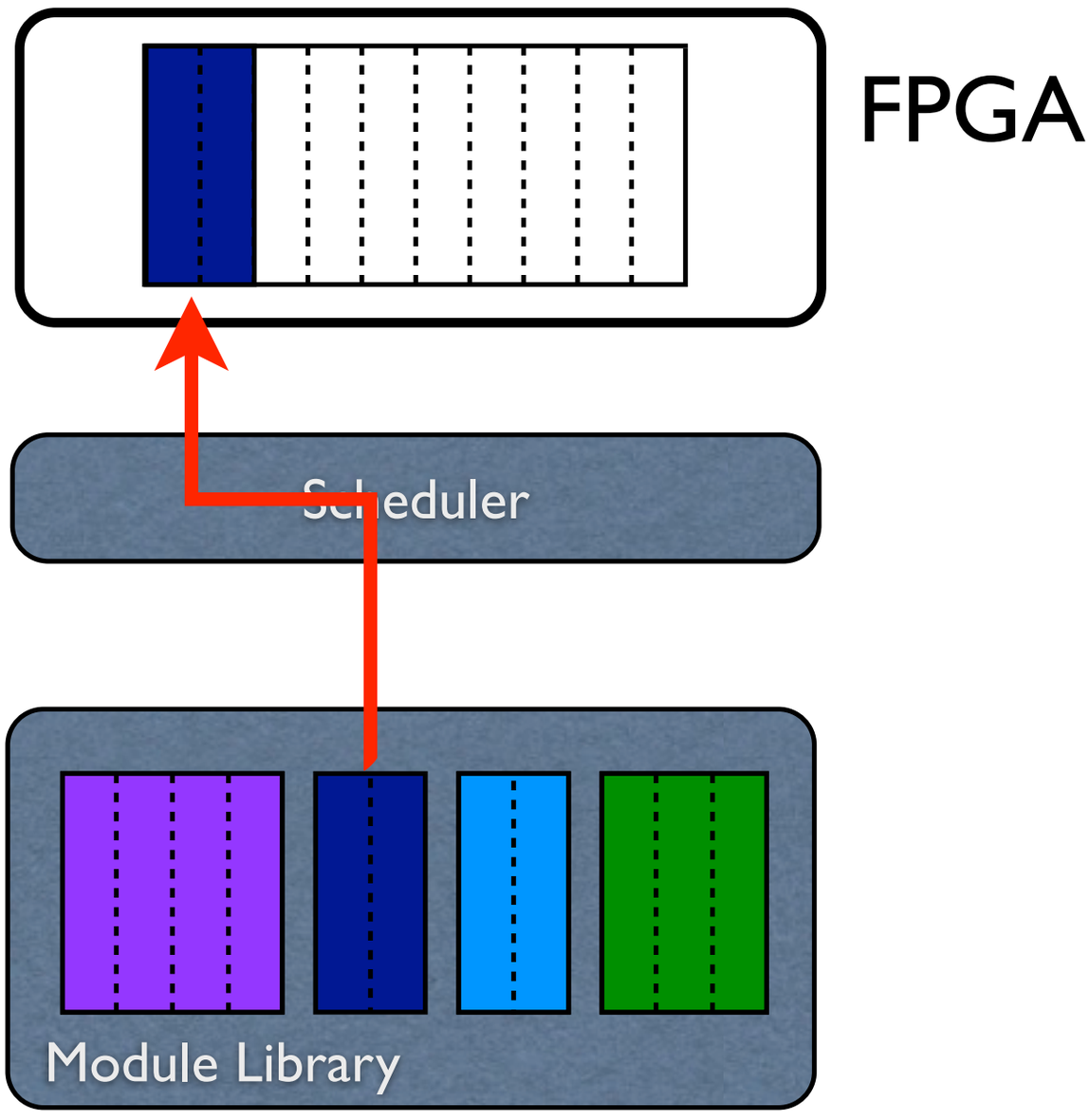
Defragmentation

Cleaning up an FPGA

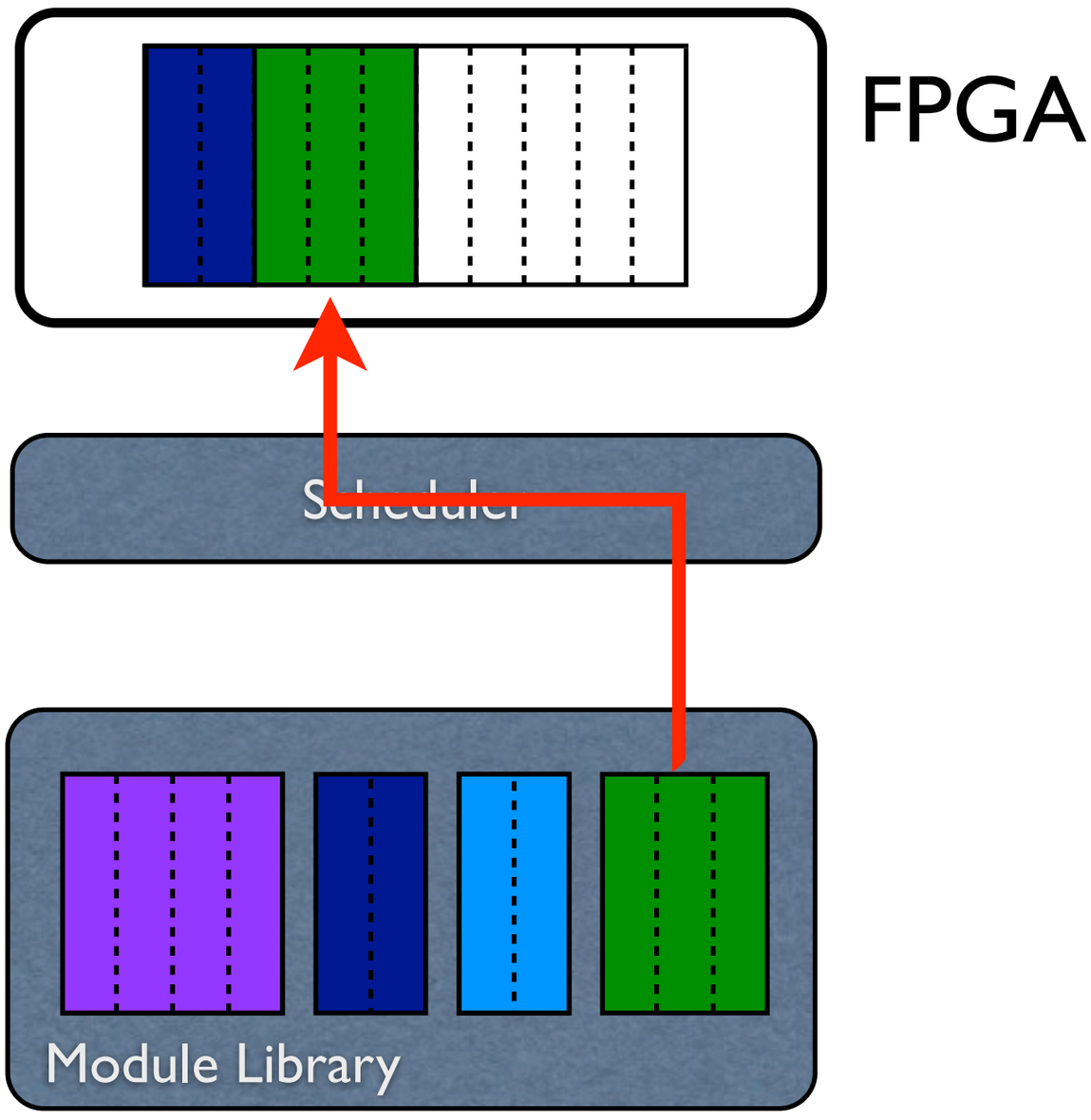
Defragmentation



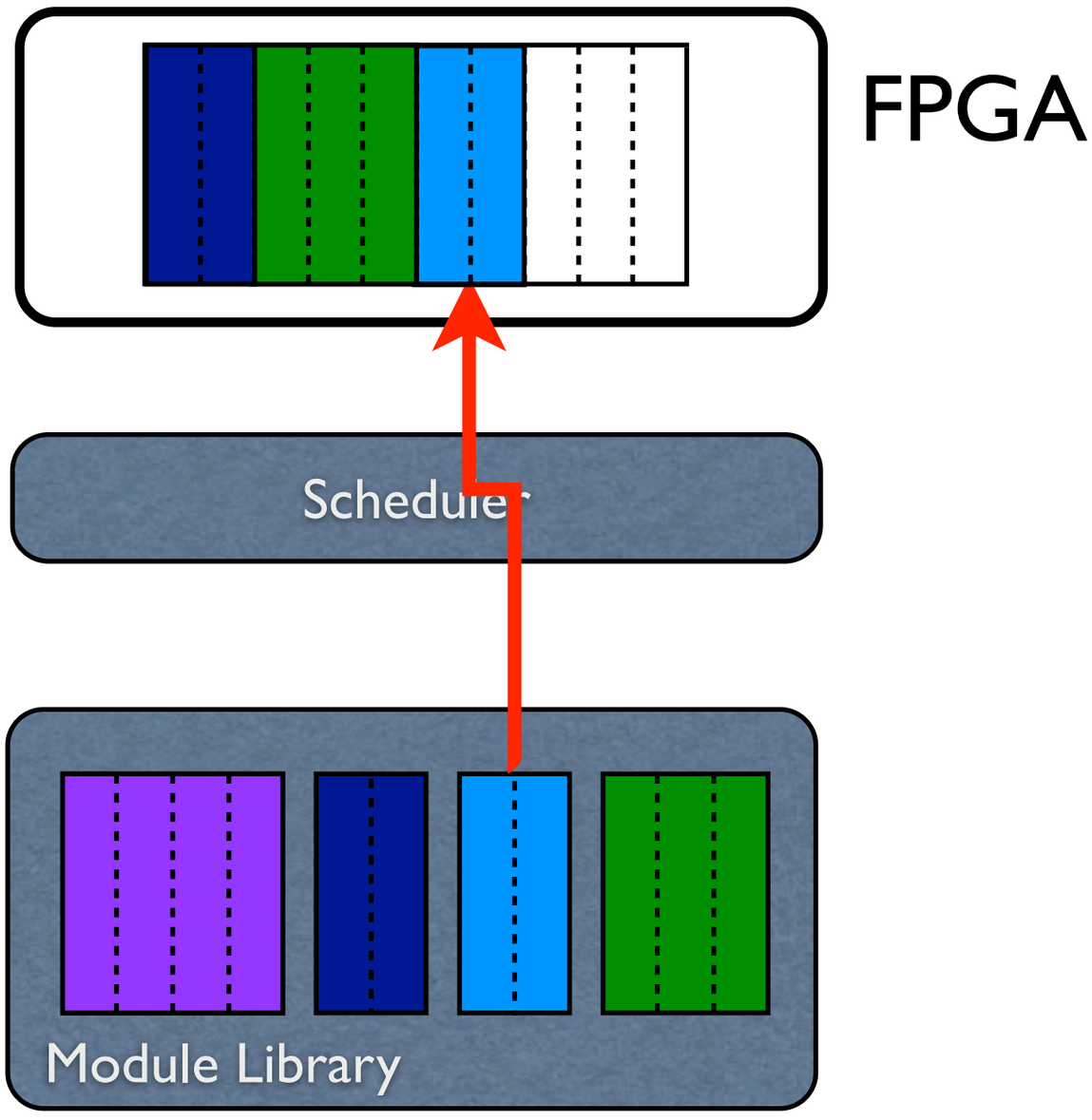
Defragmentation



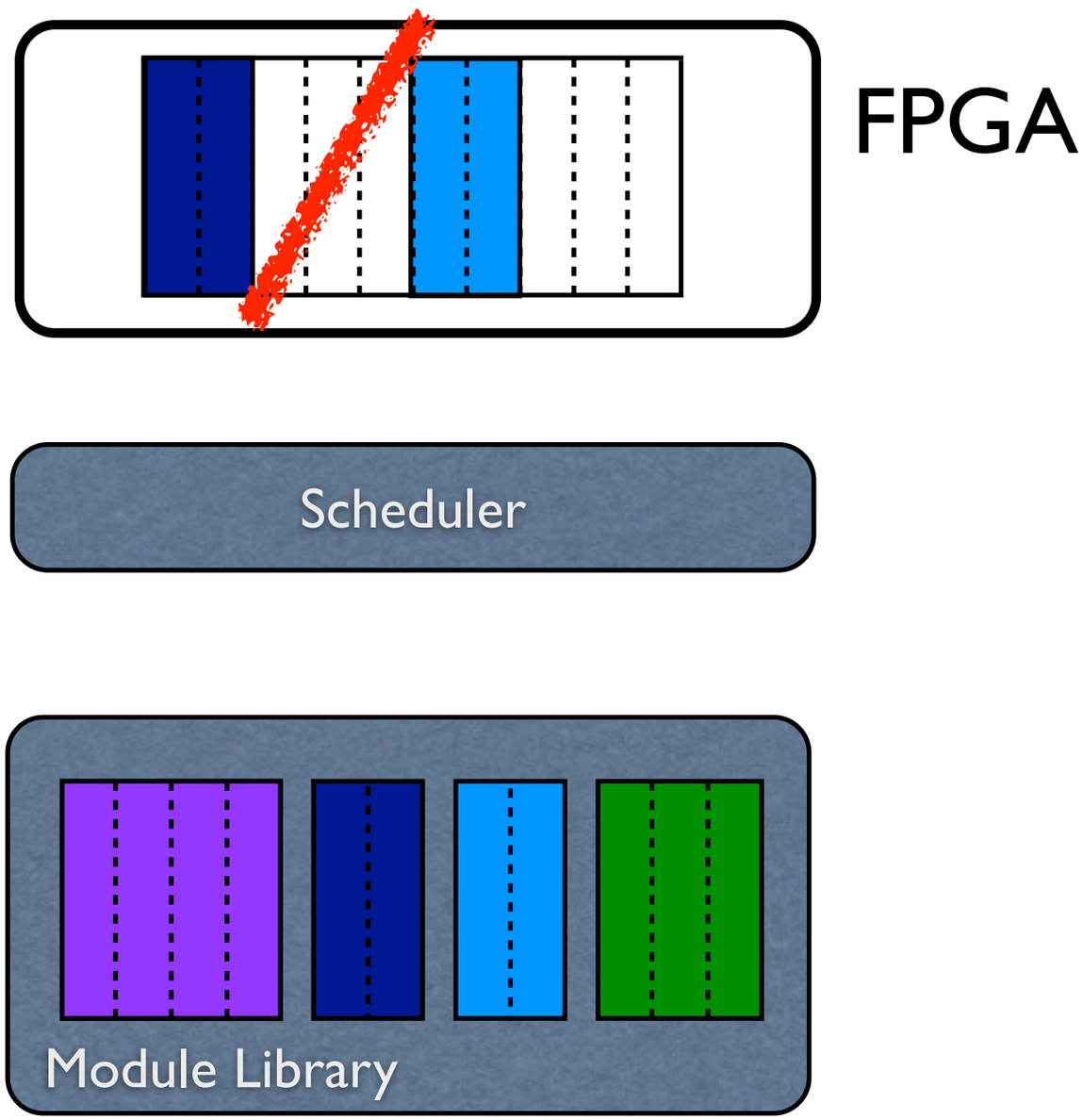
Defragmentation



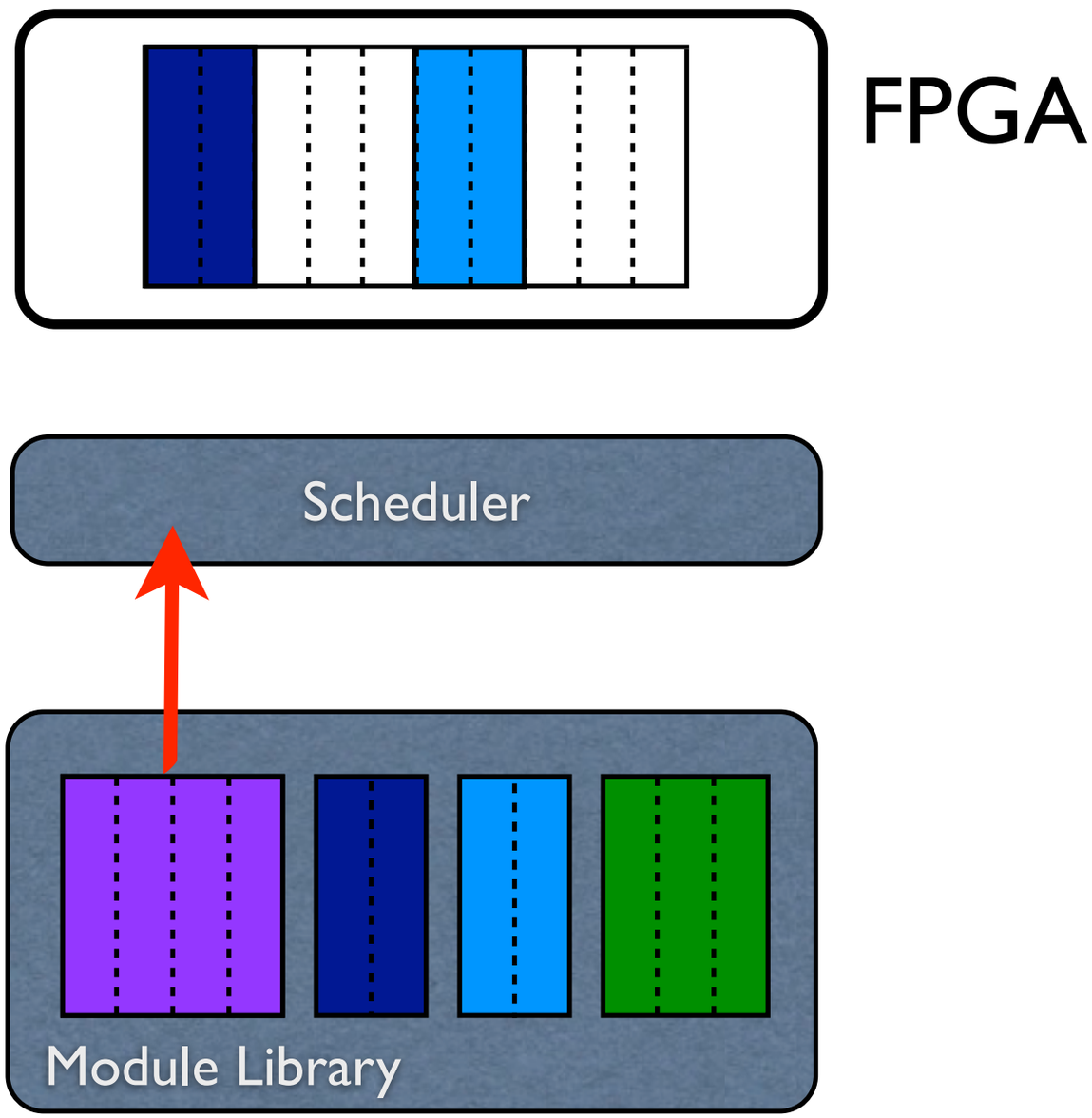
Defragmentation



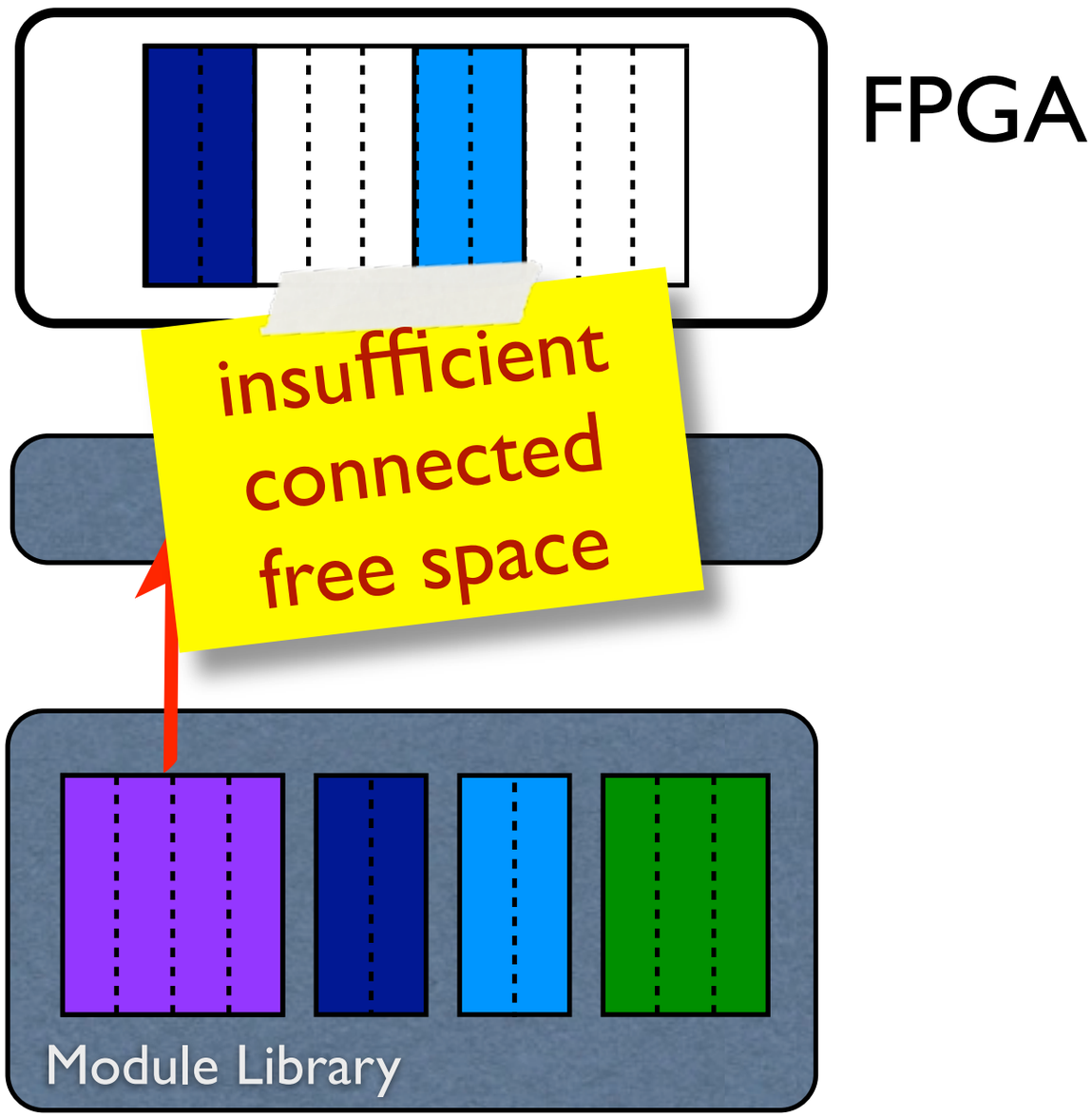
Defragmentation



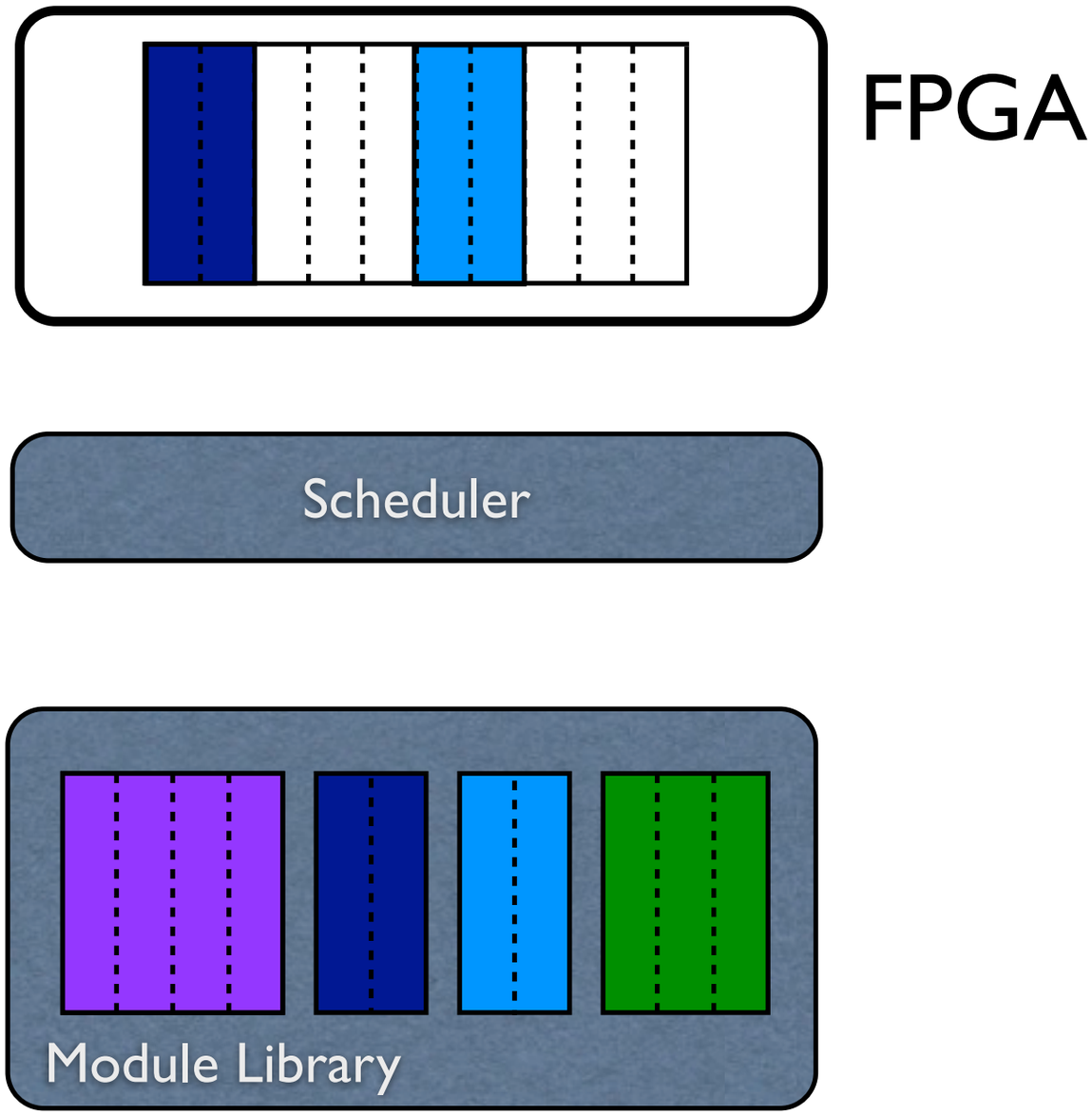
Defragmentation



Defragmentation

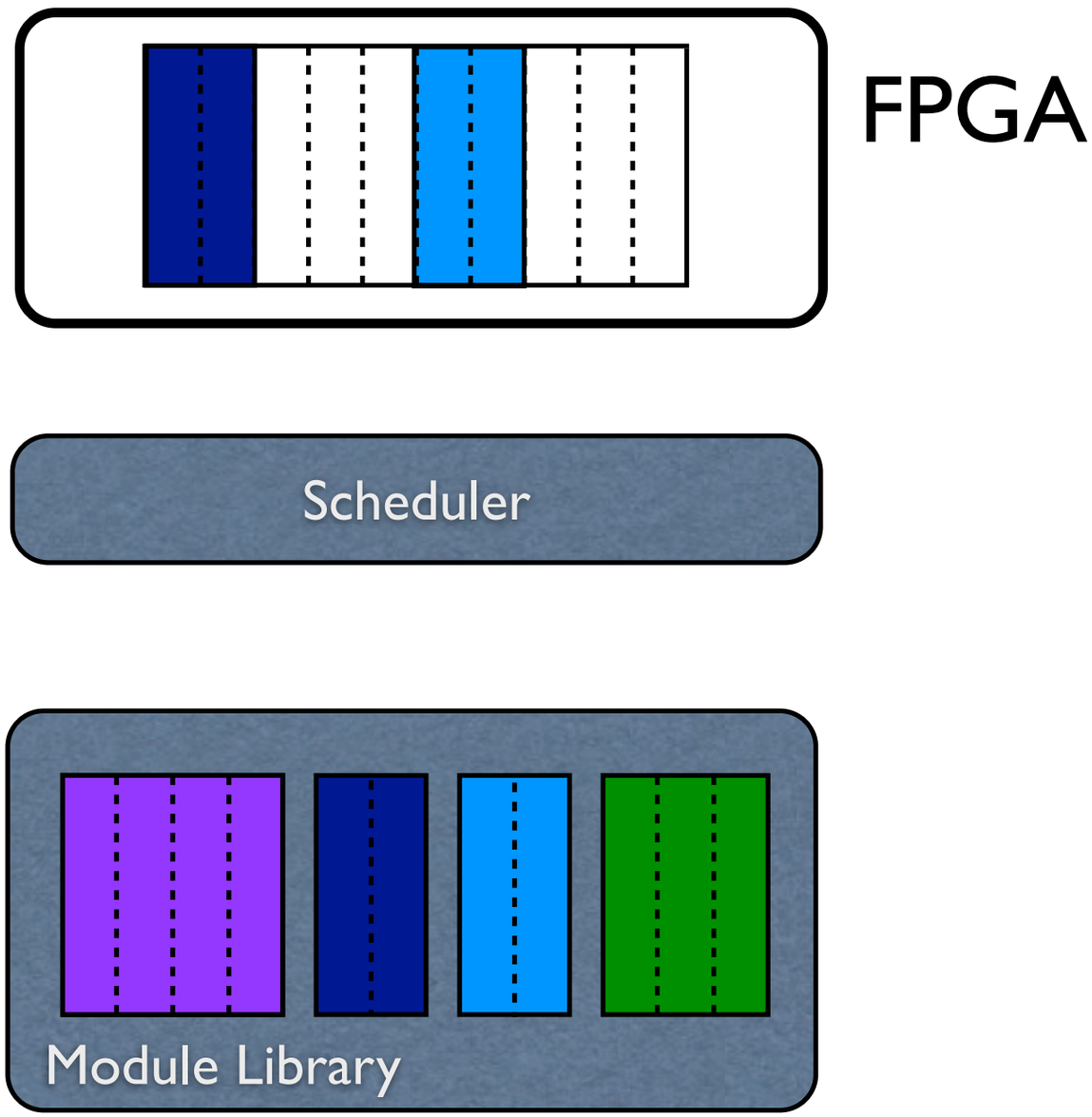


Defragmentation



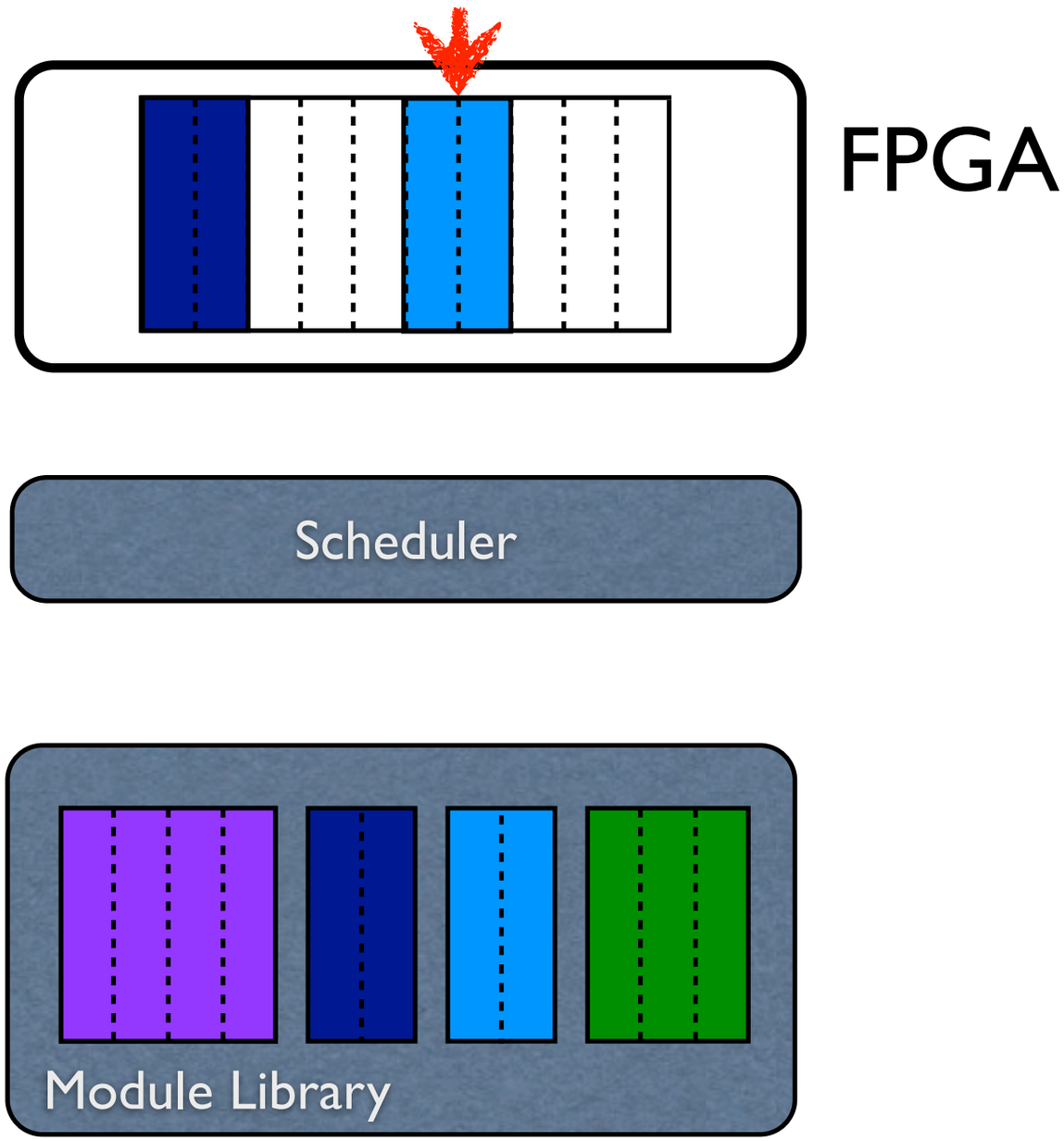
Defragmentation

Defragmentation:



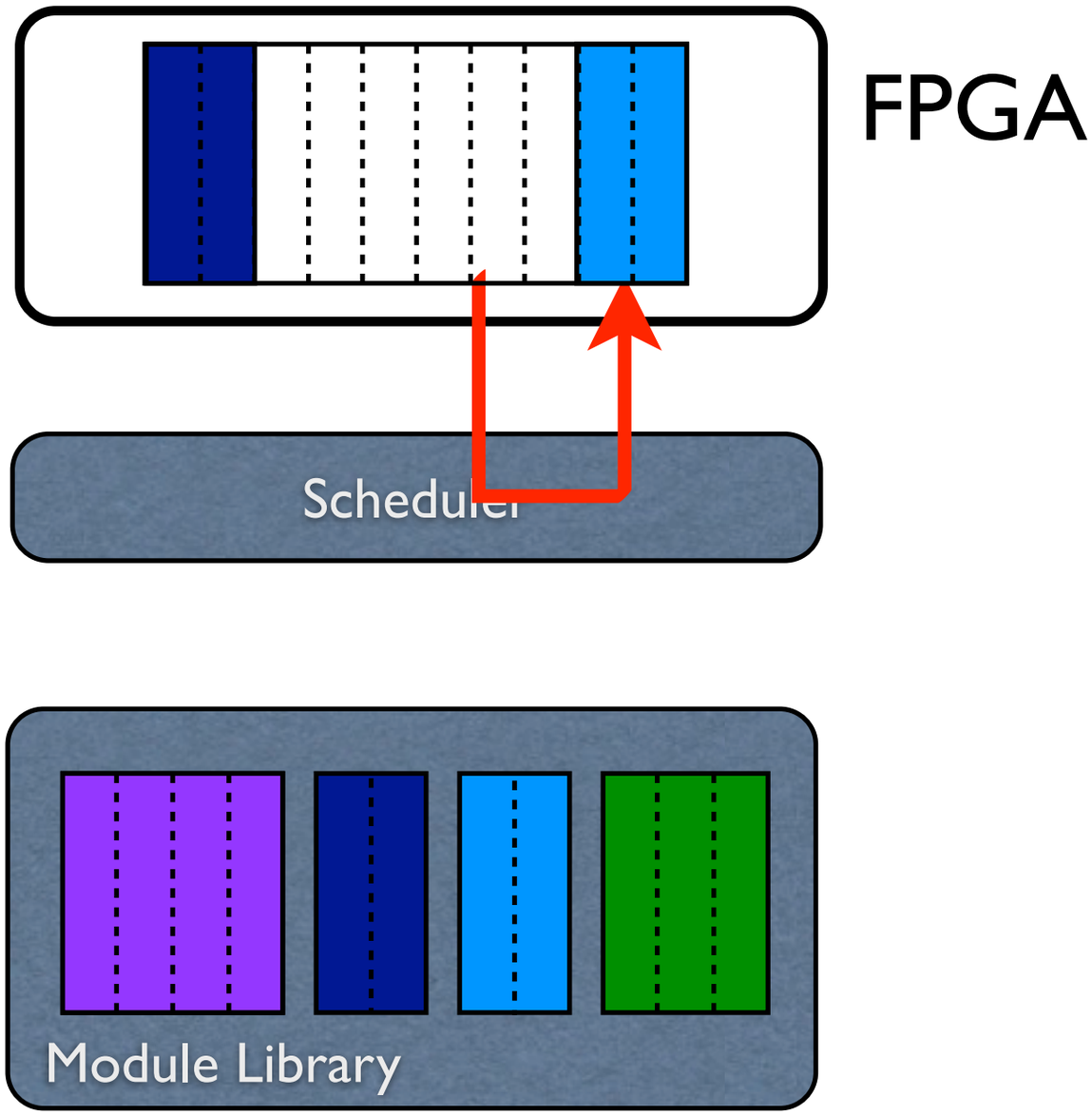
Defragmentation

Defragmentation: STOP



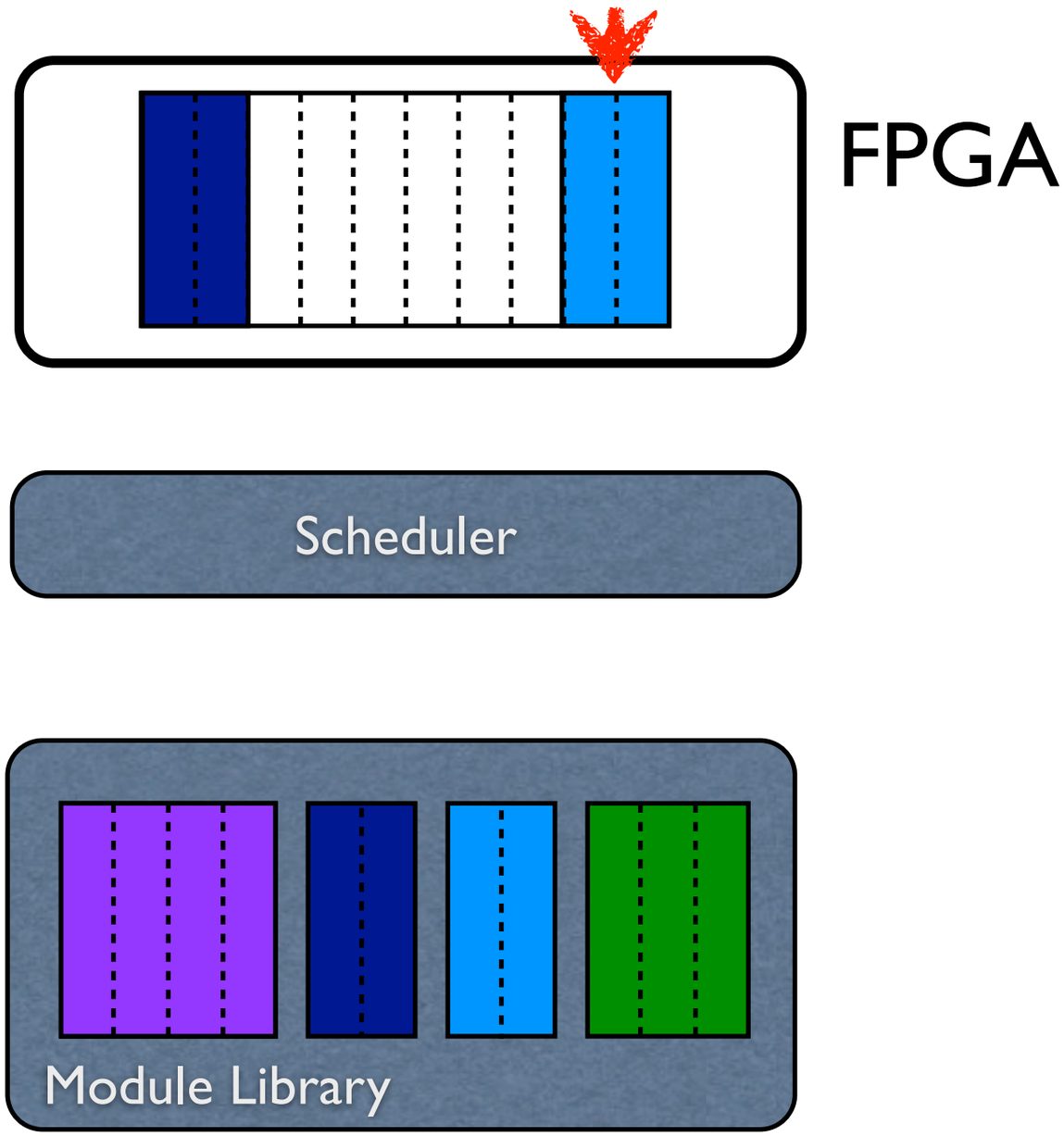
Defragmentation

Defragmentation: MOVE

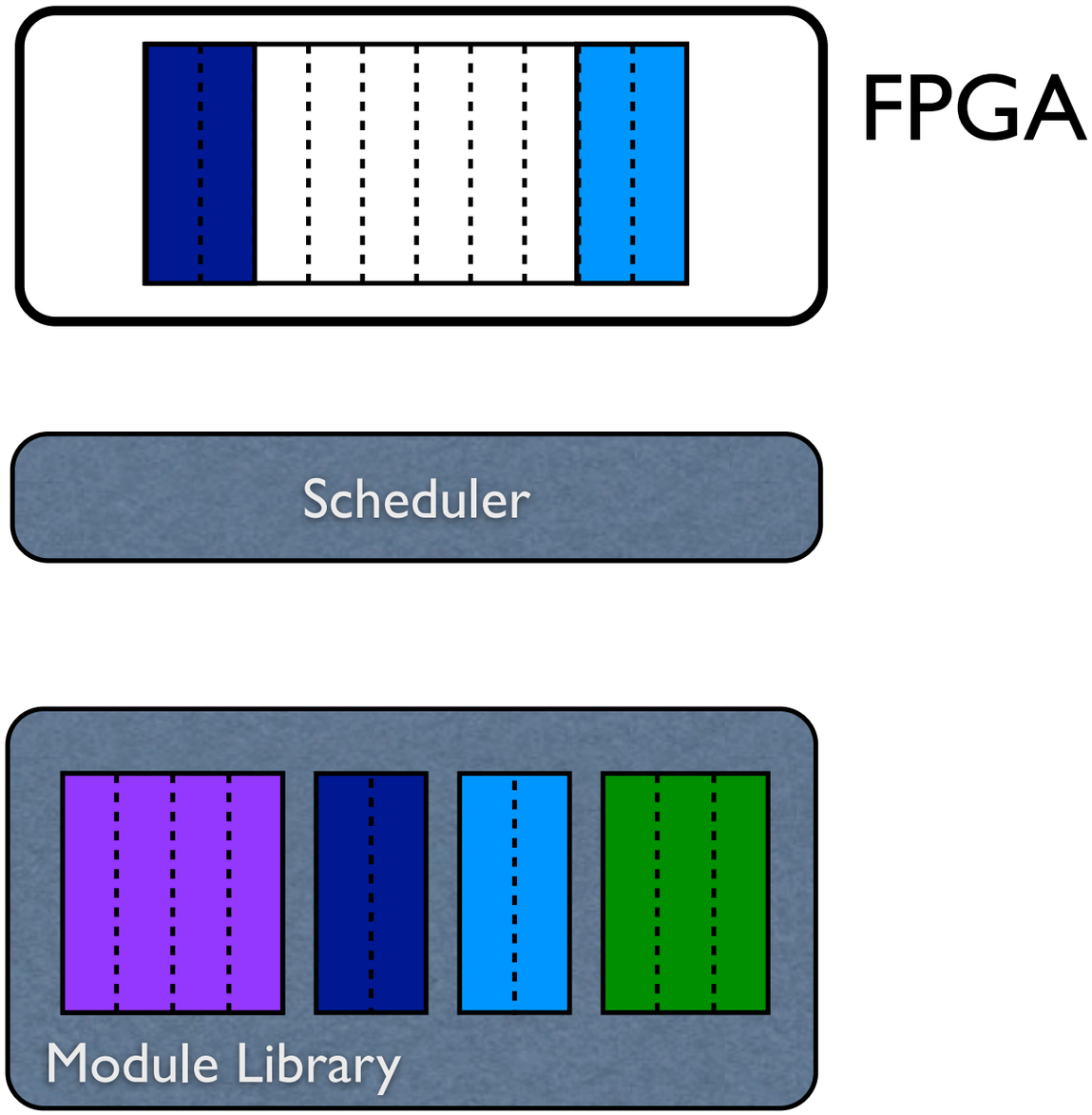


Defragmentation

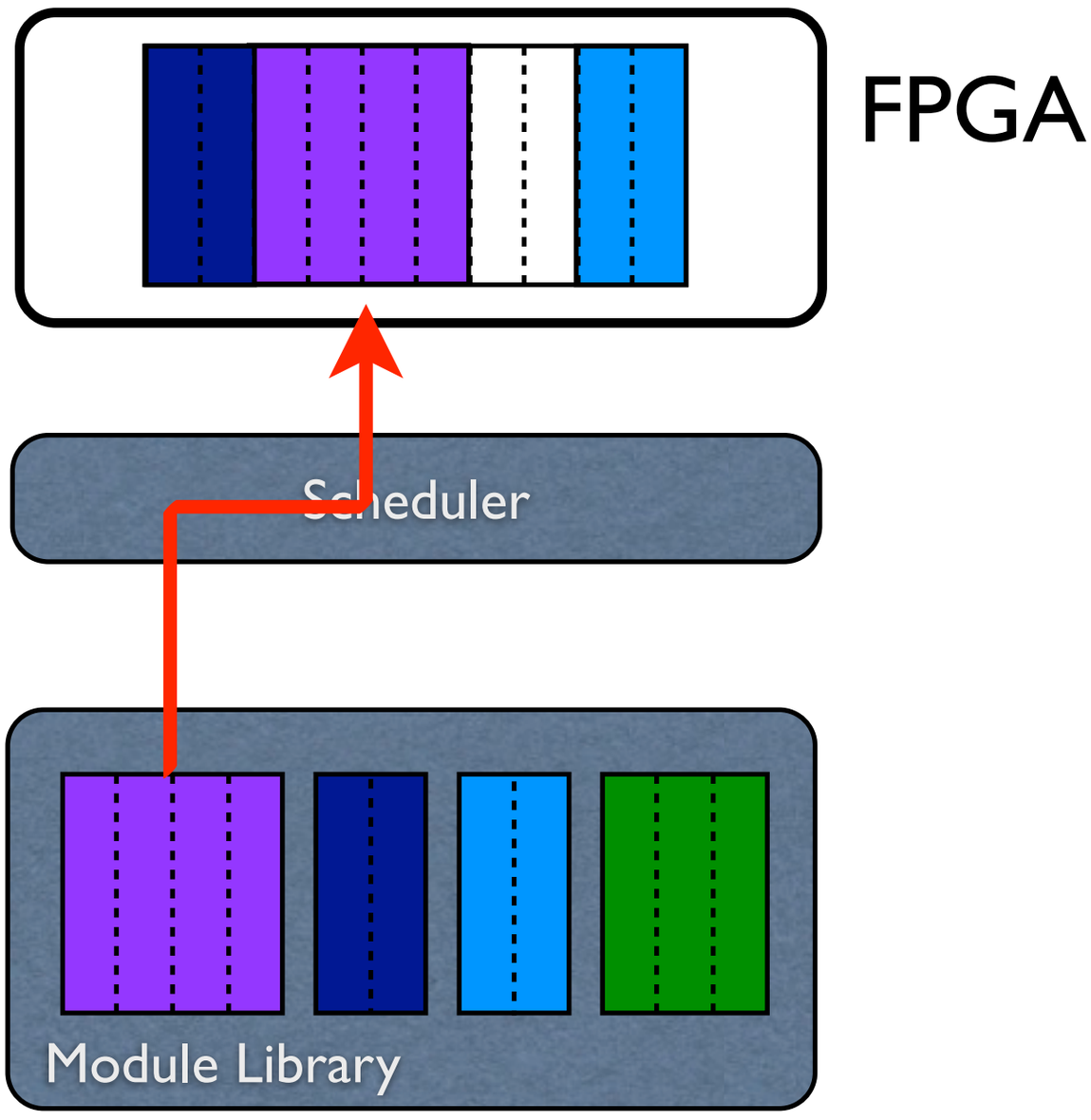
Defragmentation: RESUME



Defragmentation



Defragmentation



Defragmentation, Approach 1

- Use simple strategy for placement, defrag whole FPGA when necessary

Theorem

Rearranging an array of contiguous objects such that the free space is maximized is strongly NP-complete. Moreover, there is no PTAS within any polynomial approx. factor (unless P = NP).

Defragmentation, Approach 1

- Easy to solve if density is low

$$\delta := \frac{1}{\ell}\sum_{i=1}^{n} m_i \leq \frac{1}{2} - \frac{1}{2\ell} \cdot \max_{i=1,\ldots,n}\{m_i\}$$

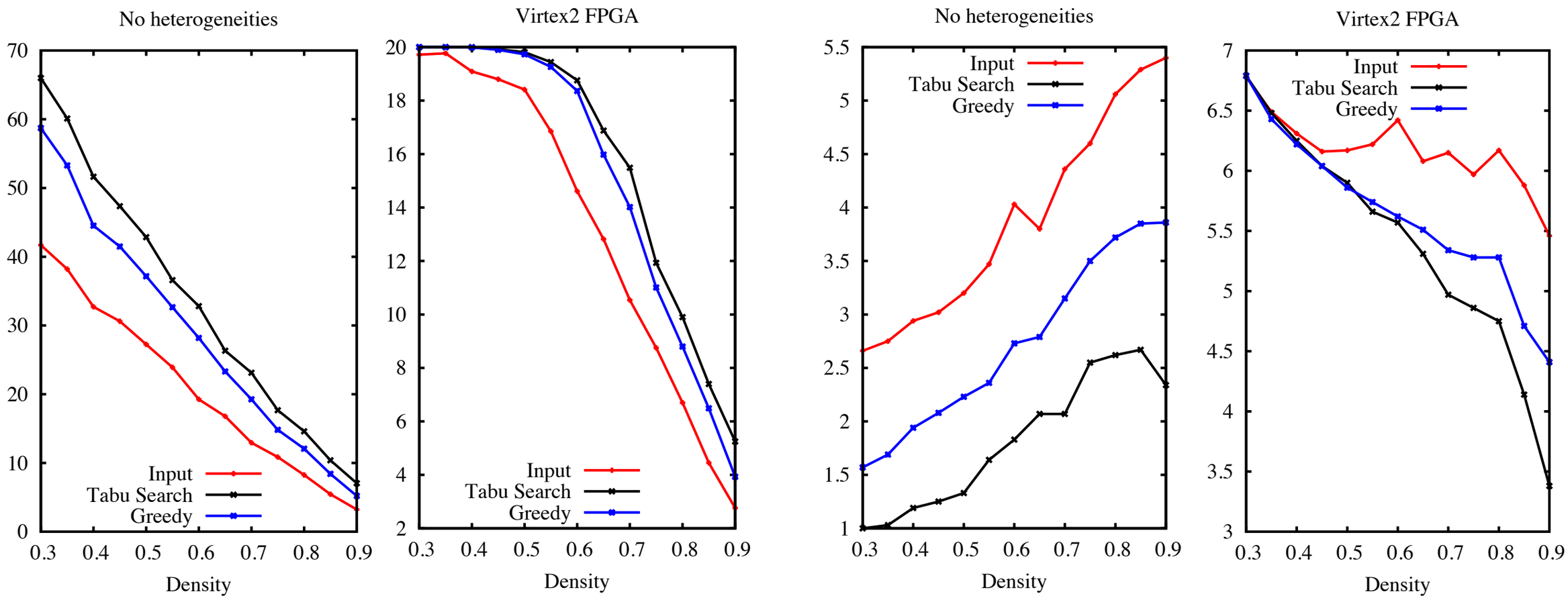
ℓ	Width of FPGA
m_i	Modules sizes

```
from right to left:
      shift every module to the right as far as possible
from left to right:
      shift every module to the left as far as possible
```

Defragmentation, Approach I

- Tabu-Search:
  - locally shift modules
  - accept shift that maximizes

$$\frac{\text{Maximal free space}}{\text{Total free space}}$$



Size of largest free space

Number of free spaces

Defragmentation, Approach II

- Maintain FPGA to avoid complete defragmentation
- Strategies:
  - *AlwaysSorted*: Keep modules sorted by their size
  - *DelayedSort*: Delay sorting until necessary
  - *ClassSort*:
    - Organize modules in size classes of power of 2
    - Maintain free blocks in each size class
  - *LocalShift*:
    - Use BestFit if possible
    - Otherwise compact FPGA in small neighborhoods of free

Defragmentation, Approach II

- Maintain FPGA to avoid complete

Theorem

AlwaysSorted achieves the optimal makespan, if there is no time penalty for moves.

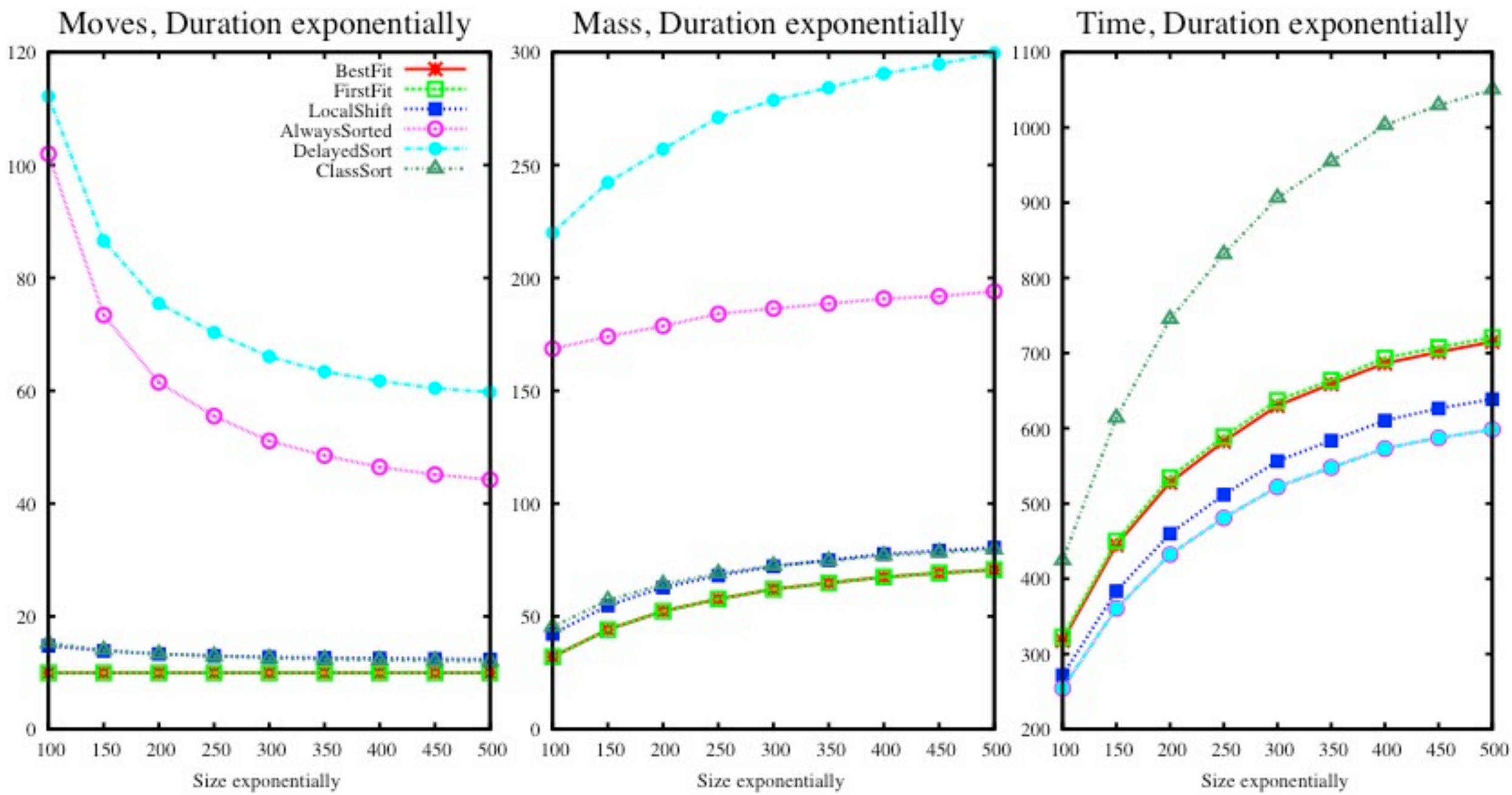
- *DelayedSort*: Delay sorting until necessary

Theorem

ClassSort performs $O(1)$ moves per operation
Amortized cost are $O(m_i \log M)$

m_i : module's size, M : largest module,

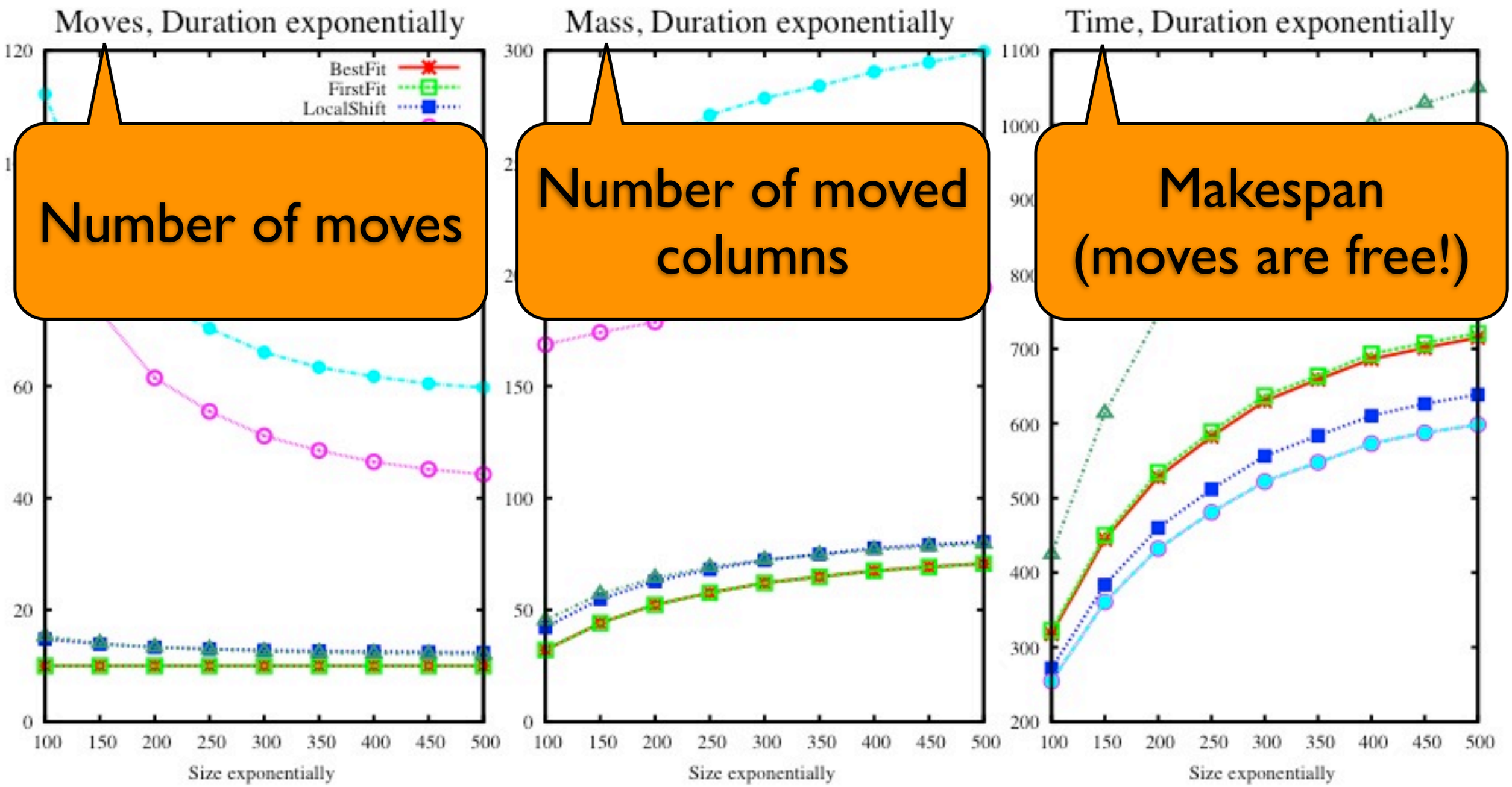
Defragmentation, Approach II



Array size: $N = 2^{10}$; $k = 8$ (LocalShift); 100000 modules per sequence; Expected duration: 300 units

Moves/Mass $\times 10.000$, Time: $\times 300.000$

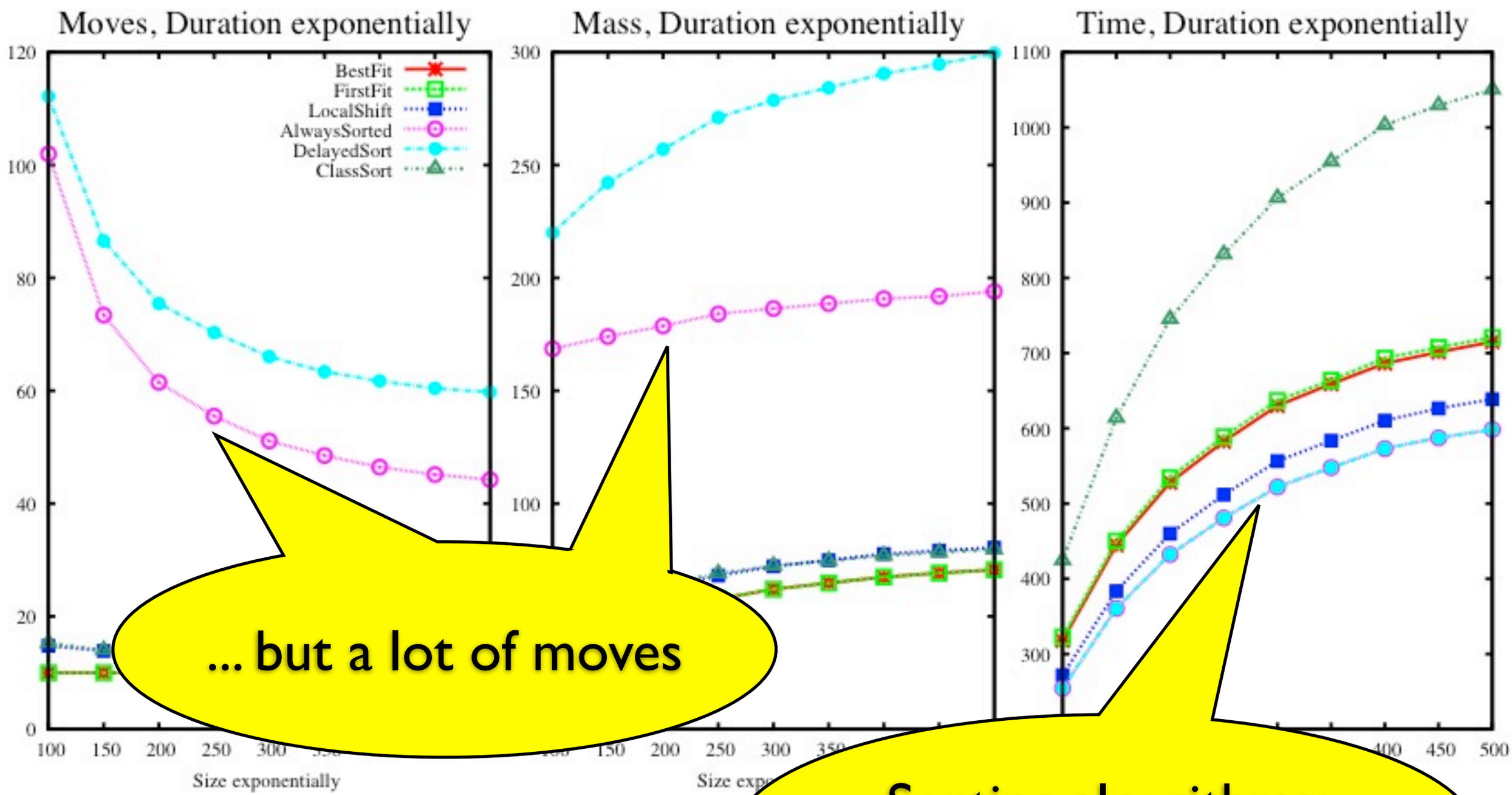
Defragmentation, Approach II



Array size: $N = 2^{10}$; $k = 8$ (LocalShift); 100000 modules per sequence; Expected duration: 300 units

Moves/Mass $\times 10.000$, Time: $\times 300.000$

Defragmentation, Approach II



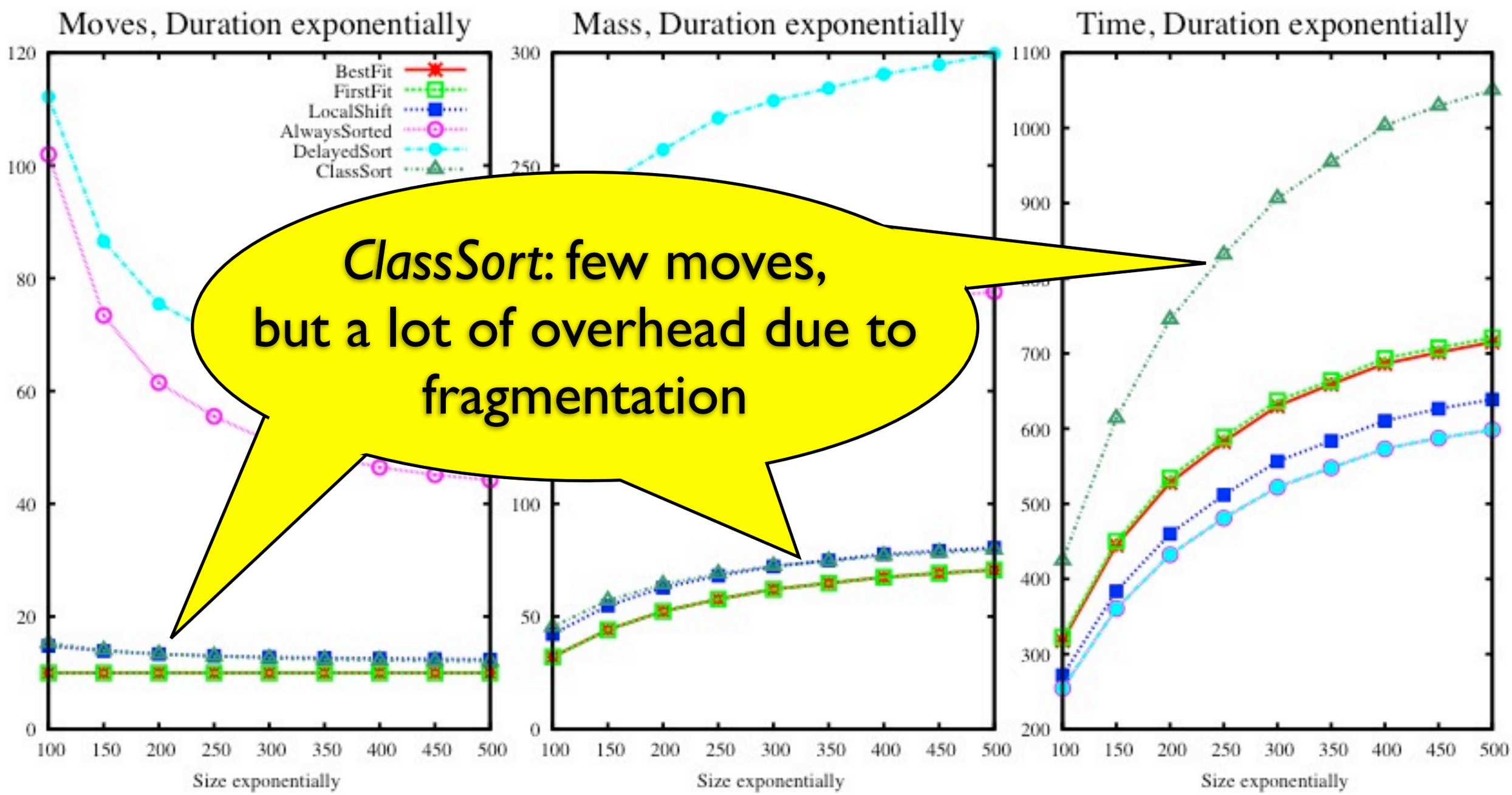
... but a lot of moves

Array size: $N = 2^{10}$; $k = 8$ (LocalShift); 100000 modules pe

Moves/Mass $\times 10.000$, Time: $\times 300.000$

Sorting algorithms: optimal makespan

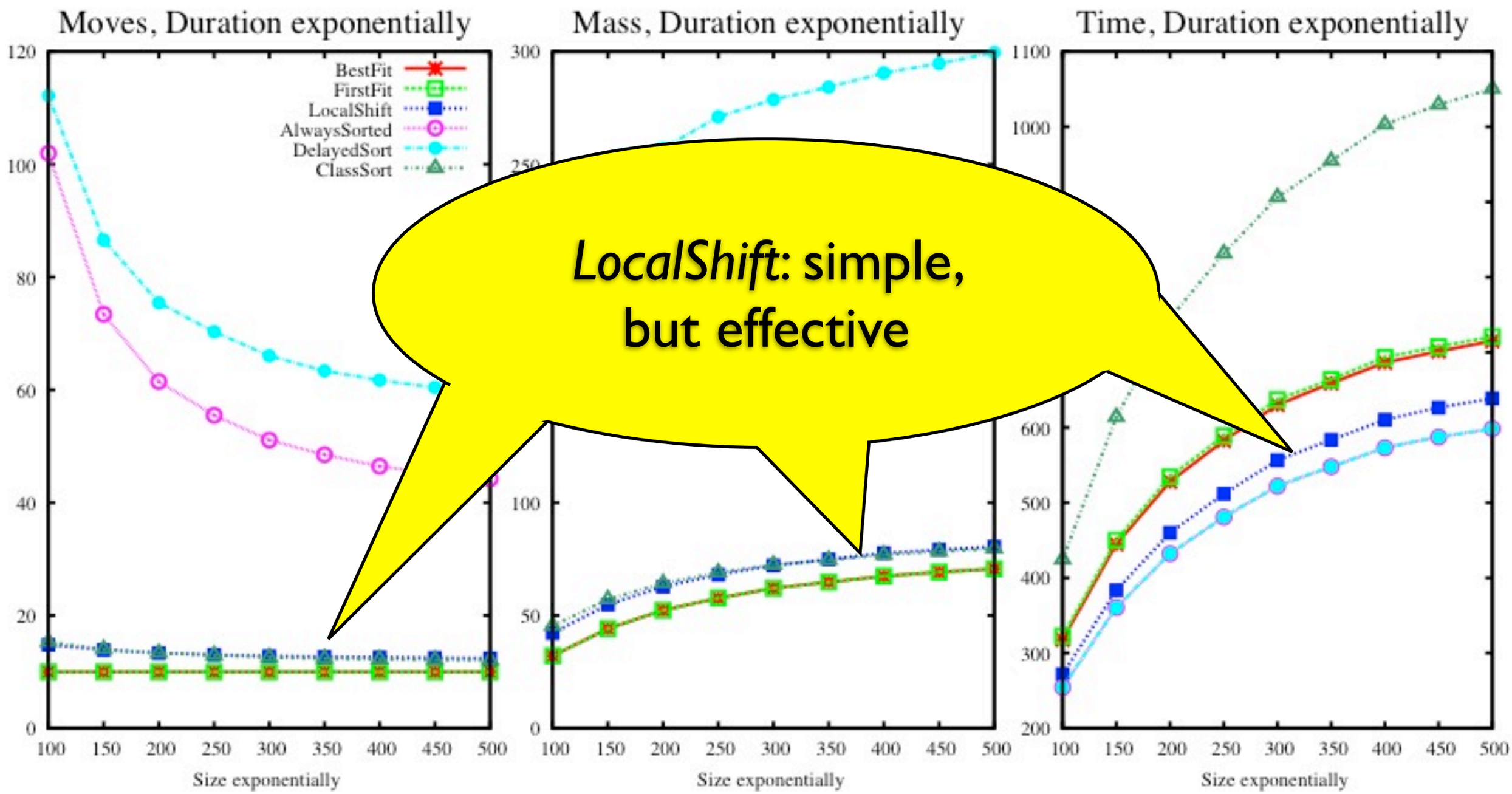
Defragmentation, Approach II



Array size: $N = 2^{10}$; $k = 8$ (LocalShift); 100000 modules per sequence; Expected duration: 300 units

Moves/Mass $\times 10.000$, Time: $\times 300.000$

Defragmentation, Approach II



Array size: $N = 2^{10}$; $k = 8$ (LocalShift); 100000 modules per sequence; Expected duration: 300 units

Moves/Mass $\times 10.000$, Time: $\times 300.000$

ReCoNodes Publications

> 40 papers

[42] Maintaining Arrays of Contiguous Objects.
M. A. Bender, S. P. Fekete, T. Kamphans, N. Schweer.
Proceedings 17th Internat. Sympos. Fund. Comput. Theory.

[41] Optimal Placement-aware Trace-based Scheduling of Hardware Reconfigurations for FPGA Accelerators.
J. Sim, W. Wong and J. Teich.
Proceedings 17th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2009), Napa, California, April, 2009

[40] General Methodology for Mapping Iterative Approximation Algorithms to Dynamically Partially Reconfigurable Systems.
J. Angermeier, A. Amouri, and J. Teich.
Proceedings 19th International Conference on Field-Programmable Logic and Applications (FPL 2009), Prague, August, 2009.

[39] Online Square Packing.
S. P. Fekete, T. Kamphans, N. Schweer.
Proceedings 20th Algorithms and Data Structure Symposium (WADS).

[38] No-Break Dynamic Defragmentation of Reconfigurable Devices.
S. P. Fekete, T. Kamphans, N. Schweer, J. van der Veen, J. Angermeier, D. Koch, and J. Teich.
Proceedings 18th International Conference on Field-Programmable Logic and Applications (FPL 2008), Heidelberg, September, 2008.

[37] A comparison of embedded reconfigurable video-processing architectures.
C. Claus, W. Stechele, M. Kovatsch, J. Angermeier and J. Teich.
Proceedings of 18th International Conference on Field Programmable Logic and Applications (FPL 2008), Heidelberg, Germany, September 8 - 10, 2008, pp. 587-590..

ReCoNodes Publications

[36] **Heuristics for Scheduling Reconfigurable Devices with Consideration of Reconfiguration Overheads.**
J. Angermeier and J. Teich.
Proceedings 15th Reconfigurable Architectures Workshop (RAW 2008), Miami, Florida, April 2008.

[35] **Reconfigurable HW/SW Architecture of a Reconfigurable HW/SW Architecture of a Real-Time Driver Assistance System.**
J. Angermeier, U. Batzer, M. Majer, J. Teich, C. Claus, and W. Stechele.
Proceedings of the Fourth International Workshop on Applied Reconfigurable Computing (ARC), Lecture Notes in Computer Science (LNCS), Springer, London, United Kingdom, March 26-28, 2008.

[34] **Offline and Online Aspects of Defragmenting the Module Layout of a Partially Reconfigurable Device.**
S. Fekete, J. van der Veen, A. Ahmadinia, D. Göhringer, M. Majer, and J. Teich.
IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2008.

[33] **The Erlangen Slot Machine: A flexible FPGA-platform for partially reconfigurable applications at run-time.**
J. Angermeier, D. Göhringer, M. Majer, and J. Teich.
Tutorial, 20th International Conference on Architecture of Computing Systems (ARCS 2007), Springer LNCS series, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, March 12-15, 2007.

[32] **Scheduling and communication-aware mapping of HW-SW modules for dynamically and partially reconfigurable SoC architectures.**
S. Fekete, J. van der Veen, J. Angermeier, D. Göhringer, M. Majer, and J. Teich.
In Proceedings of the Dynamically Reconfigurable Systems Workshop (DRS 2007), Zürich, Switzerland, pages 151-160, March 15, 2007.

ReCoNodes Publications

[31] The Erlangen Slot Machine: A Dynamically Reconfigurable FPGA-Based Computer.
M. Majer, J. Teich, A. Ahmadinia, and C. Bobda.
Journal of VLSI Signal Processing Systems, Springer, vol. 46(2), March 2007.

[30] The Erlangen Slot Machine - A Platform for Interdisciplinary Research in Reconfigurable Computing.
J. Angermeier, D. Göhringer, M. Majer, J. Teich, S. Fekete, and J. van der Veen.
it - Information Technology, Heft 3/2007, Oldenbourg, München, 2007.

[29] Optimal free-space management and routing-conscious dynamic placement for reconfigurable computing.
A. Ahmadinia, C. Bobda, S. Fekete, J. Teich, and J. van der Veen.
IEEE Transactions on Computers, volume 56, number 3, 2007.

[28] An FPGA-Based Dynamically Reconfigurable Platform: from Concept to Realization.
M. Majer., J. Teich
In Proceedings of 16th International Conference on Field Programmable Logic and Applications, pp. 963-964, Madrid, Spain, August 28-30, 2006.

[27] Minimizing communication cost for reconfigurable slot modules.
S. Fekete, J. van der Veen, M. Majer and J. Teich.
In Proceedings 16th International Conference on Field-Programmable Logic and Applications (FPL 2006), pp. 535-540, Madrid, Spain, August 28-30, 2006.

[26] Higher-dimensional packing with order constraints.
S. Fekete, E. Köhler and J. Teich. SIAM Journal on Discrete Mathematics.

[25] Optimal Simultaneous Scheduling, Binding and Routing for Processor-Like Reconfigurable Architectures.
J. A. Brenner, J. van der Veen, S. Fekete, J. Oliveira Filho, W. Rosenstiel .
In Proceedings 16th International Conference on Field-Programmable Logic and Applications (FPL 2006), pp. 527-534, Madrid, Spain, August 28-30, 2006.

Thank you