Department of Mathematical & Statistical Sciences

UNIVERSITY OF COLORADO **DENVER**

# A Greedy Reduction Algorithm

Bradley R. Lowery
University of Colorado Denver

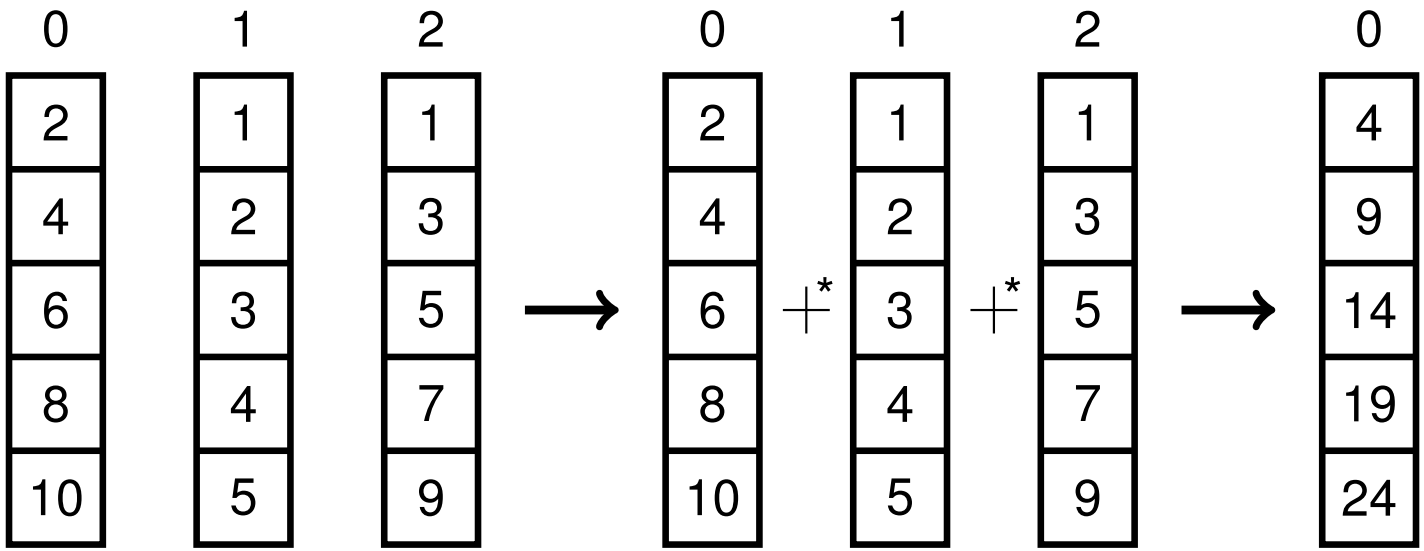
JUNE 29, 2012

## Overview

- What is the reduction operation?
- Communication Model
- Standard reduction algorithms
- The greedy algorithm
- Theoretical and numerical results
- Nonuniform segmentation

## What is a reduction?

- We will consider a set of $p$ processors with distributed memory and each processor has a message of size $m$.
- A reduction combines the messages entry-wise, and returns the value on one specified processor.
- Example: $p = 3$, $m = 5$

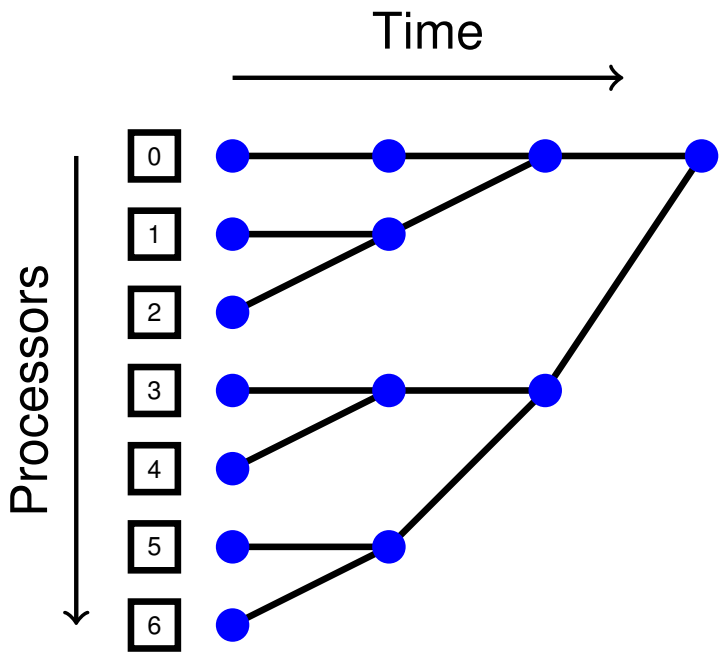



* Can be any associative operation.

## Communication Model

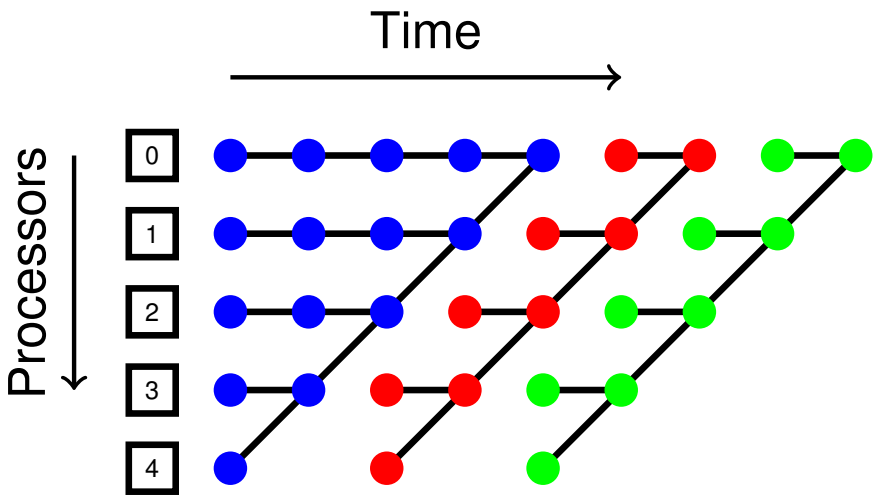
- Unidirectional system - At any given time a processor is allowed to send a message to another processor or receive a message from another processor, but not both.
- Communication time between two processors is given by the linear model, $\alpha + \beta m$, where $\alpha$ is the latency (start up time) and $\beta$ is the inverse bandwidth.
- The time for the computation is given $\gamma m$.
- In practice we have the relationship, $\alpha >> \beta > \gamma$.
- The message $m$ can be split into $q$ segments of size $s_i$.
- Uniform segmentation: $s_i = s$ for all $i$.

## Binomial Tree



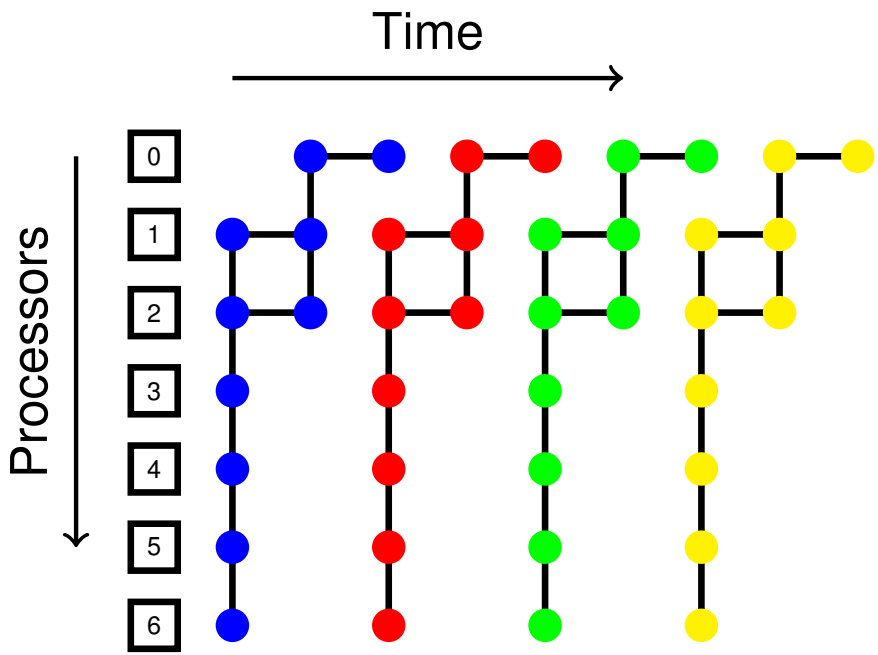

- Best for small messages, $\alpha >> \beta m$.
- Minimizes the number of communications started.
- No segmentation. Only increases latency.
- $time = \lceil \log_2 p \rceil (\alpha + \beta m + \gamma m)$

## Pipeline Tree




- Best for long messages, $\alpha << \beta m$.
- Poor startup, but optimal overhead for trailing segments.
- $time = ((p-1) + 2(q-1))(\alpha + \beta s + \gamma s)$

## Binary Tree




- At each step, two different processors send to the same receiving processor.
- An iteration is therefore twice as long as compared to the other trees.
- Good for medium sized messages, $\alpha \approx \beta m$.
- $time = (2\lceil \log_2(p+1) \rceil + 3(q-1))(\alpha + \beta s + \gamma s)$

# Optimal Segmentation

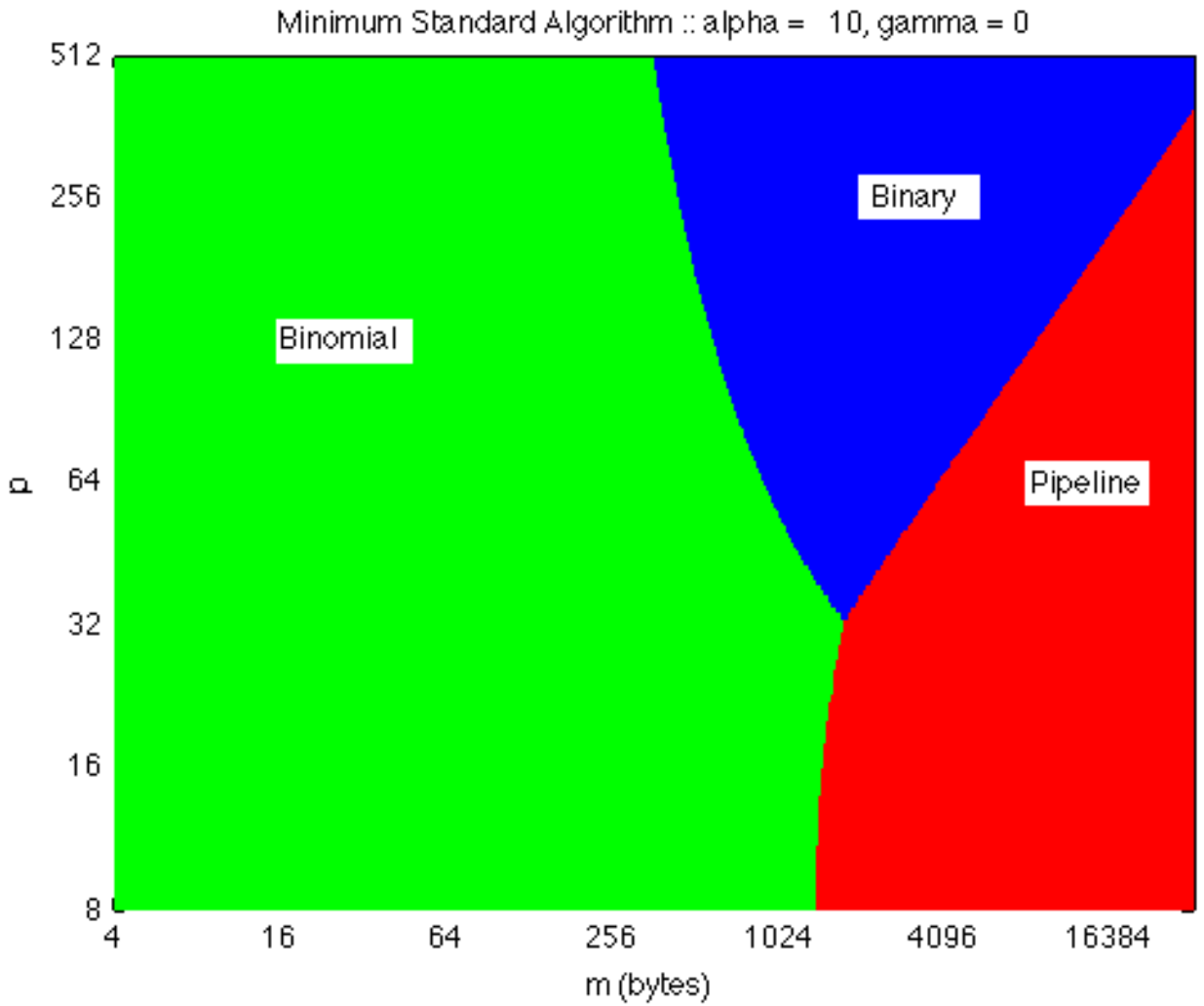
| | | |
|---|---|---|
| Binomial | Time | $\lceil log_2 p \rceil (\alpha + \beta m + \gamma m)$ |
| Pipeline | Time | $(p-1)(\alpha + \beta s + \gamma s) + 2(q-1)(\alpha + \beta s + \gamma s)$ |
| | $s_{opt}$ | $\left( \frac{2m\alpha}{(p-3)(\beta+\gamma)} \right)^{1/2}$ |
| | $T_{opt}$ | $\left[ ((p-3)\alpha)^{1/2} + (2m(\beta+\gamma))^{1/2} \right]^2$ |
| Binary | Time | $2\left( \lceil log_2(p+1) \rceil - 1 \right)(\alpha + \beta s + \gamma s) + 3(q-1)(\alpha + \beta s + \gamma s)$ |
| | $s_{opt}$ | $\left( \frac{3m\alpha}{(N-3)(\beta+\gamma)} \right)^{1/2}$ |
| | $T_{opt}$ | $2\left[ ((2N-5)\alpha)^{1/2} + (2m(\beta+\gamma))^{1/2} \right]^2$ |

- $N = \lceil \log_2(p+1) \rceil$.
- Assume uniform segmentation.

## Lower Bounds for each term in communication time

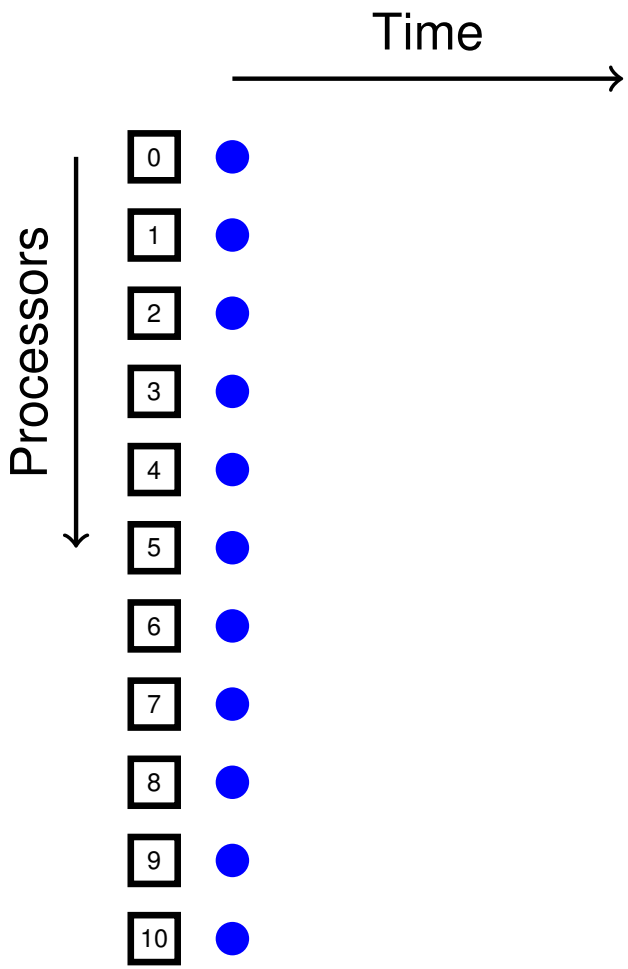
| | Latency | Bandwidth | Computation |
|---|---|---|---|
| Reduce | $\lceil log_2 p \rceil \alpha$ | $2m\beta$ | $\frac{p-1}{p} m\gamma$ |
| Binomial | $\lceil \log_2 p \rceil \alpha$ | $\lceil \log_2 p \rceil m\beta$ | $\lceil \log_2 p \rceil m\gamma$ |
| Pipeline | $(p-1)\alpha$ | $(p-3+2m)\beta$ | $(p+2m-3)\gamma$ |
| Binary | $2(N-1)\alpha$ | $(2N-5+3m)\beta$ | $2(N+2m-3)\gamma$ |

*$N = \lceil \log_2(p+1) \rceil$



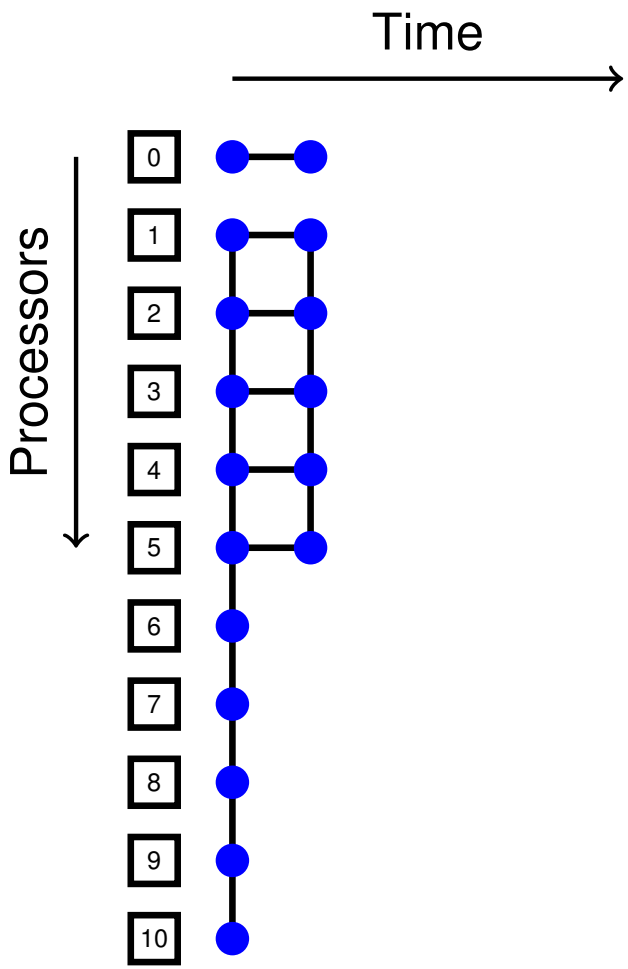

Region where binomial, pipeline, or binary is better. Each algorithm is tuned for optimal uniform segmentation. $\alpha = 10, \beta = 1, \gamma = 0$.

## Greedy Tree



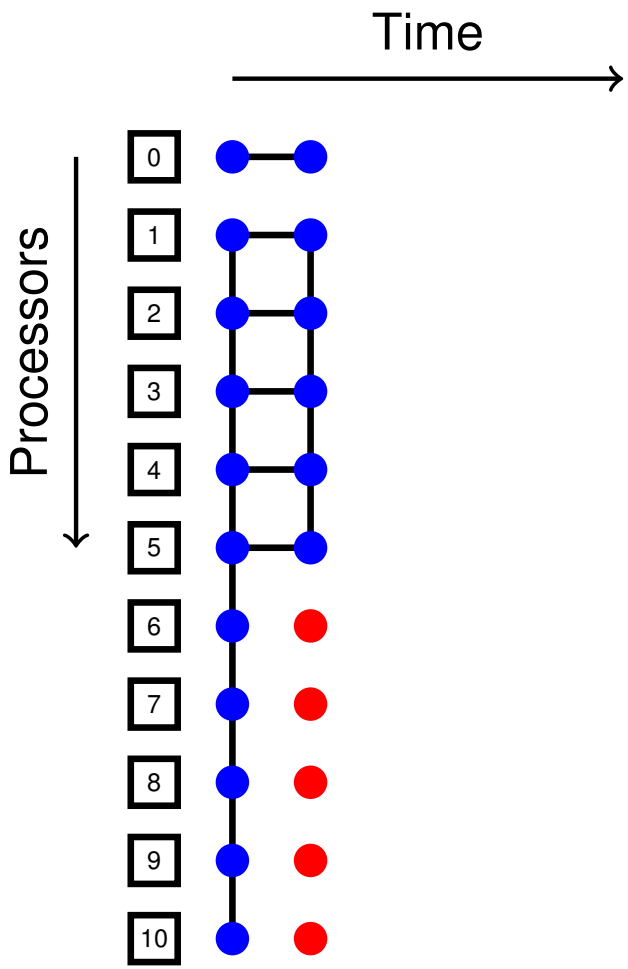

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



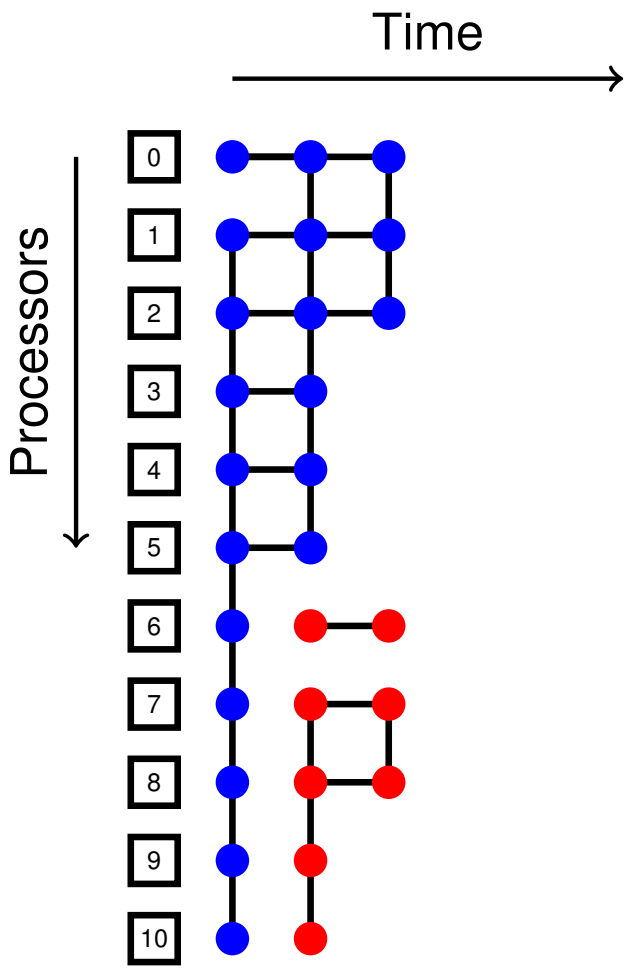

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



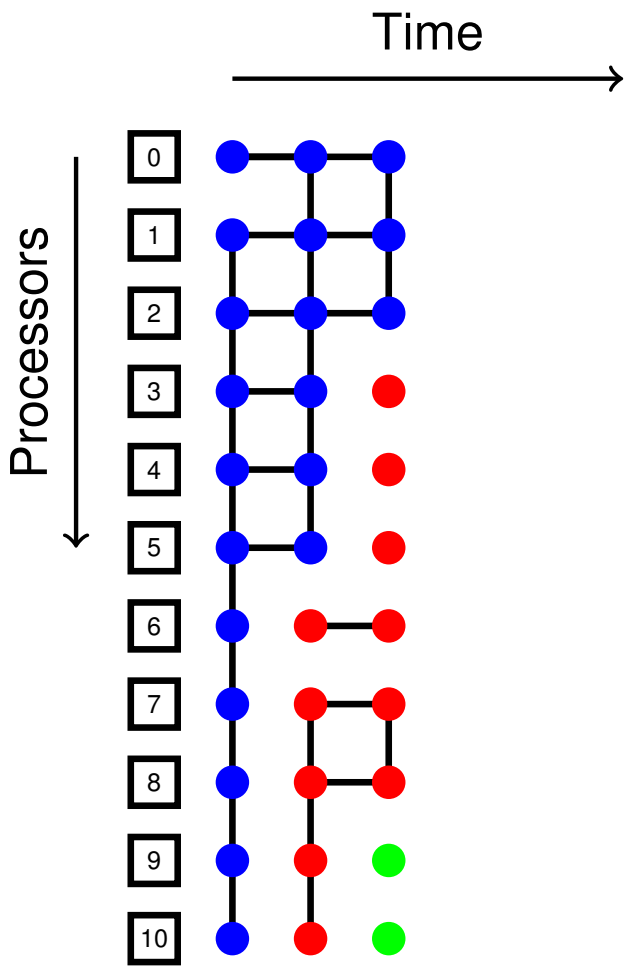

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



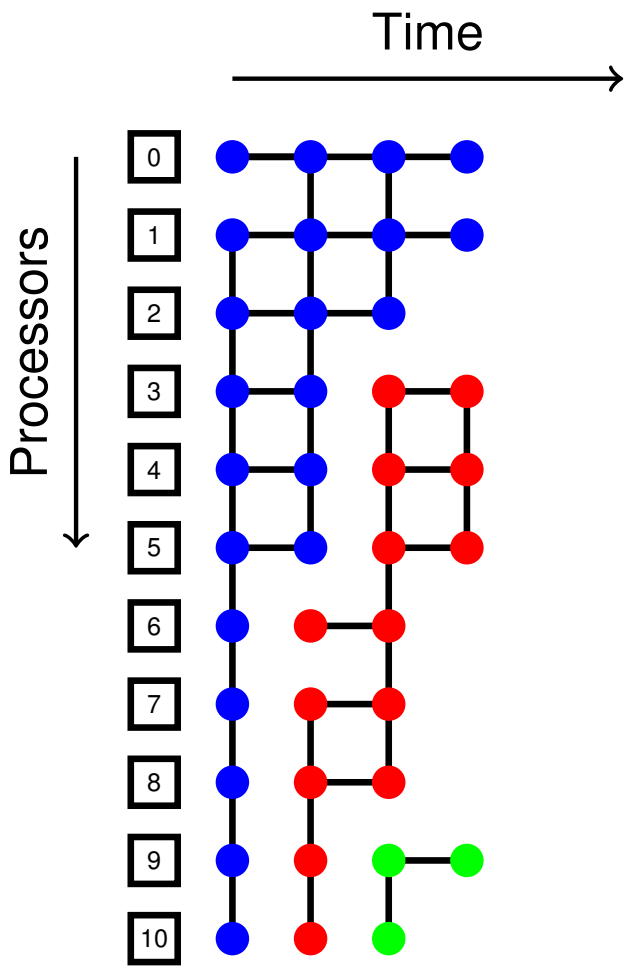

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



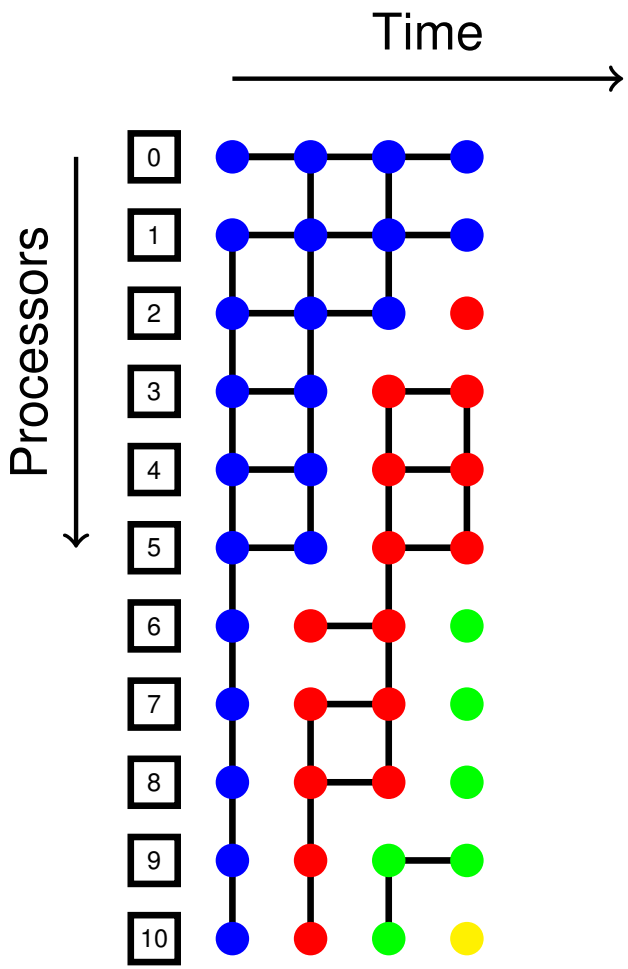

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



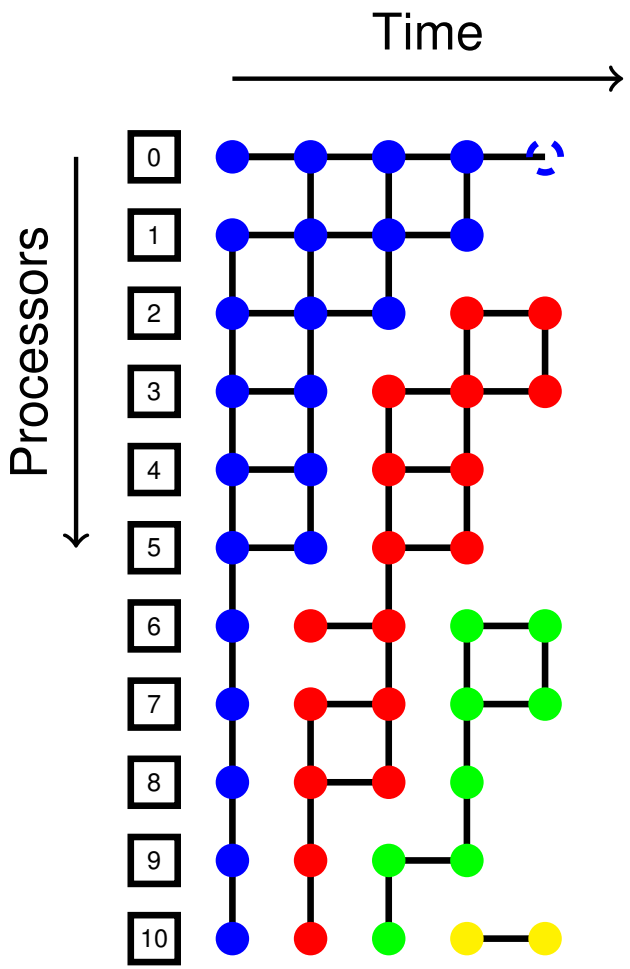

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



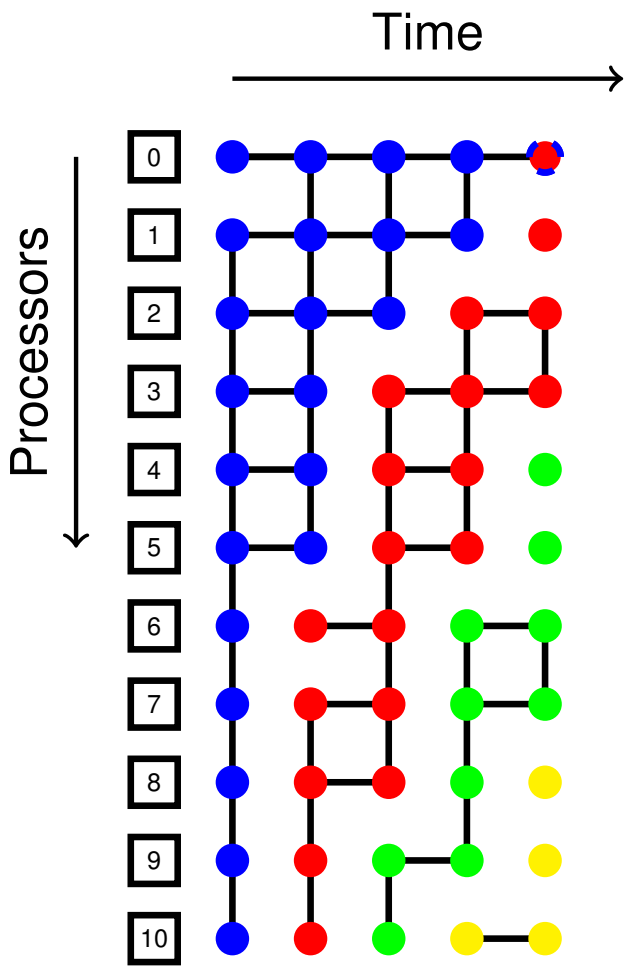

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



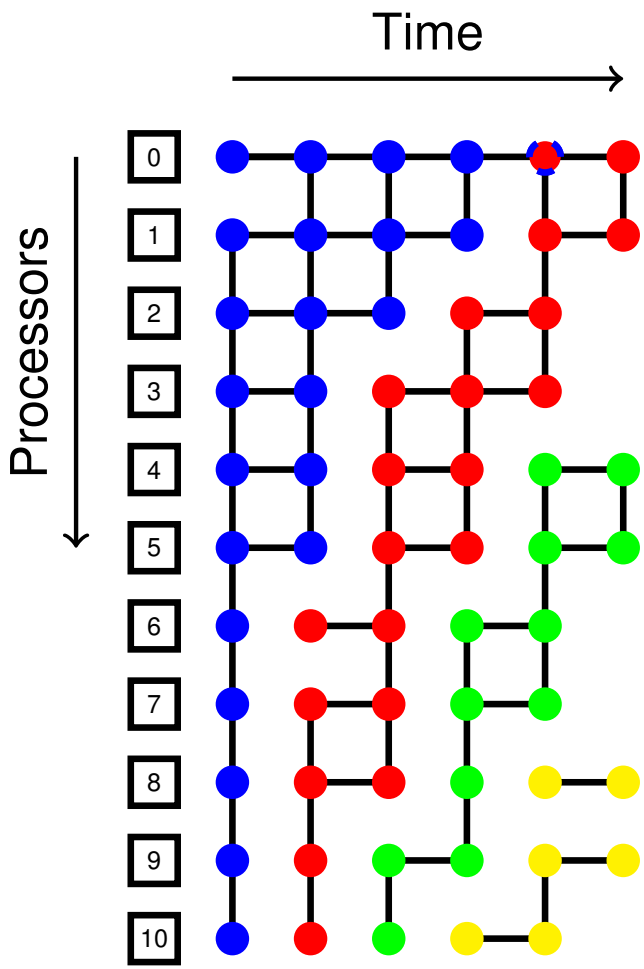

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



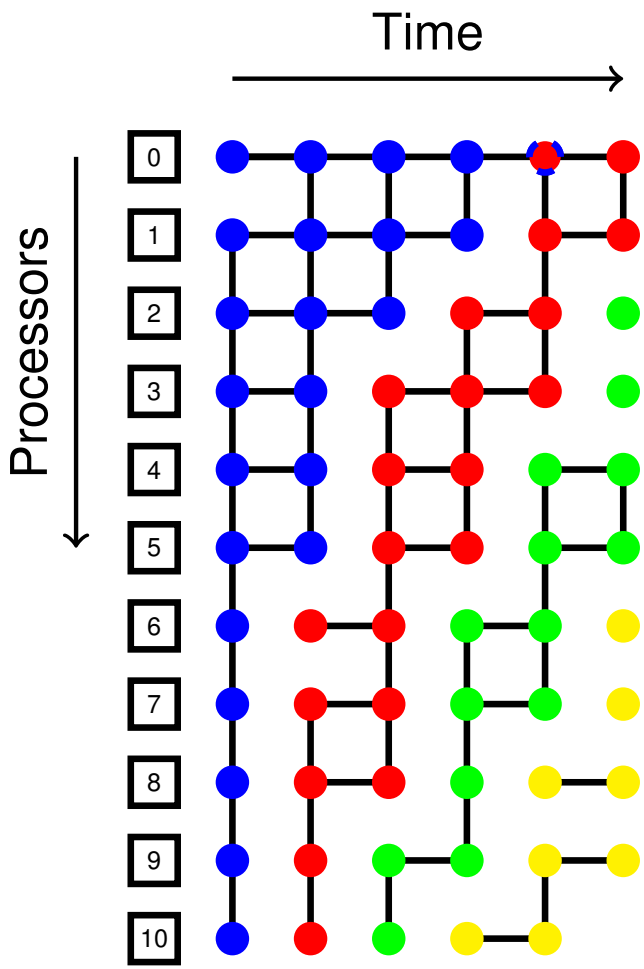

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



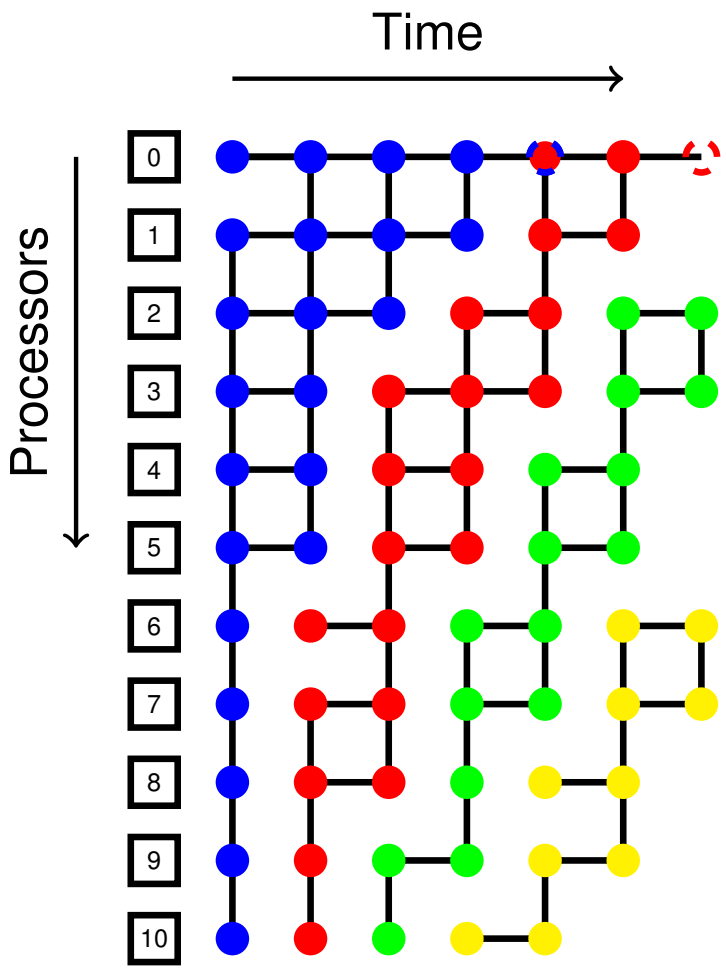

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



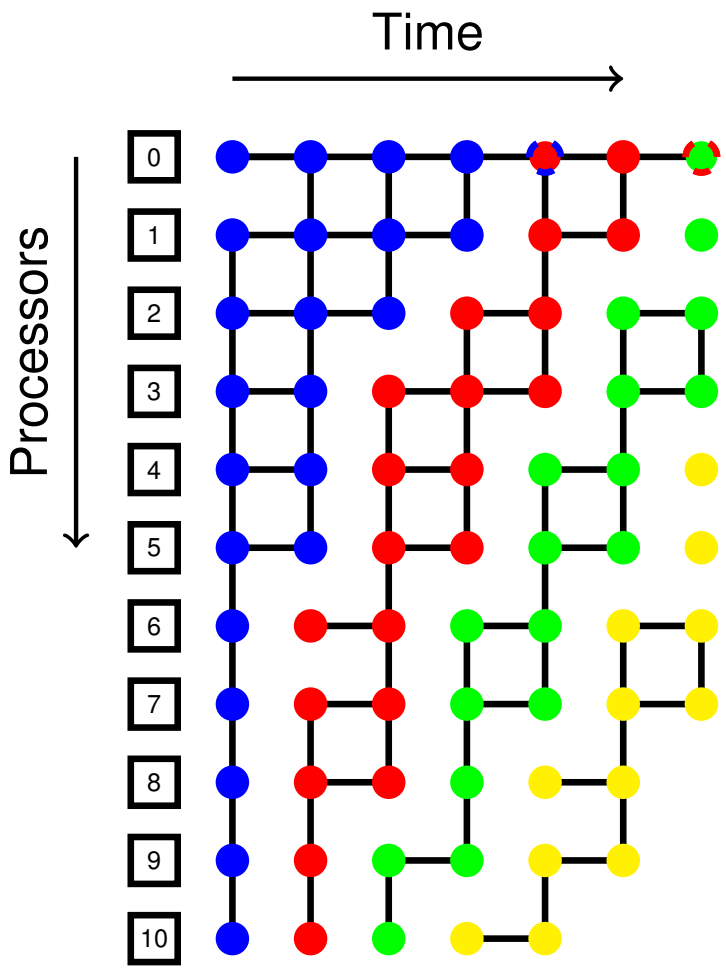

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



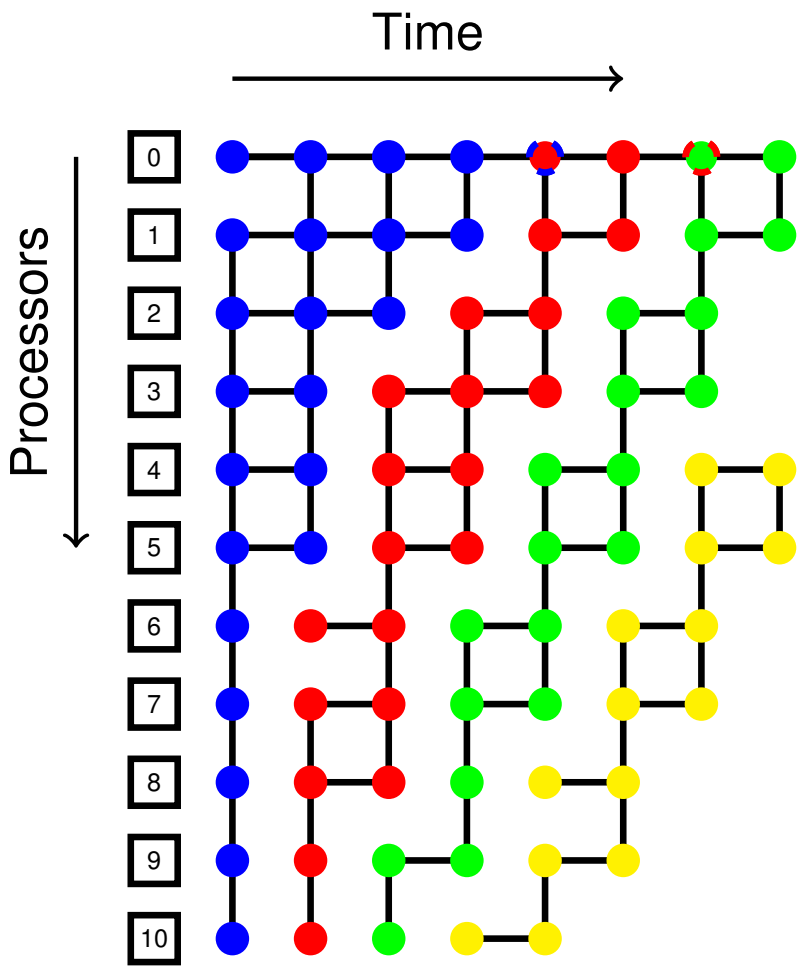

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



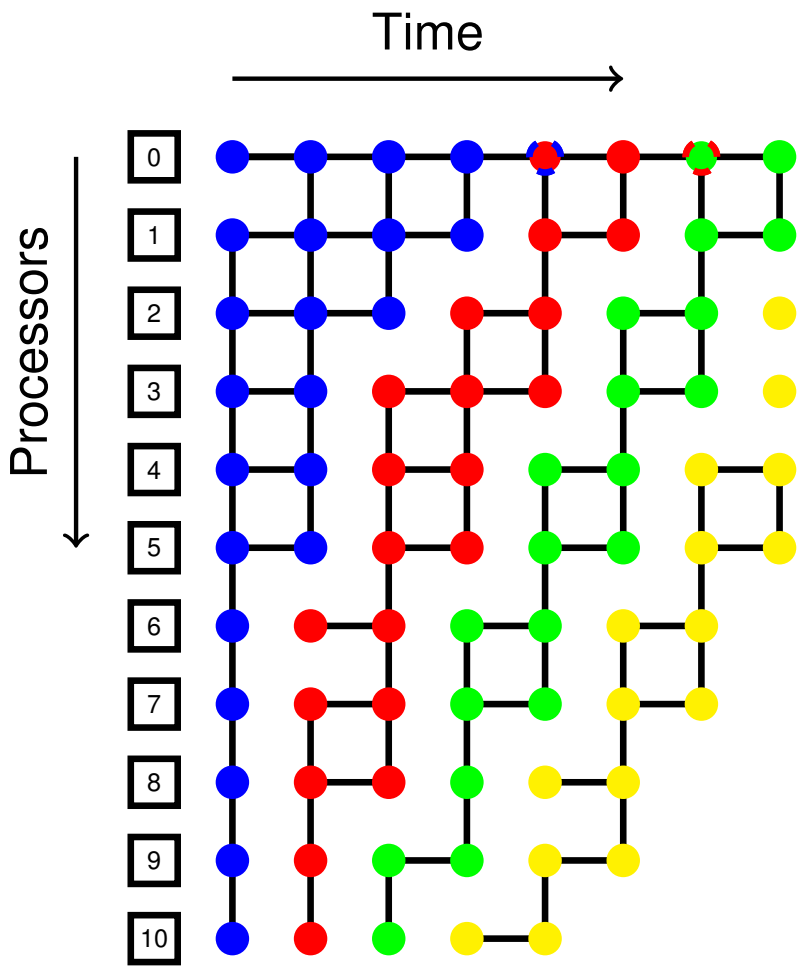

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



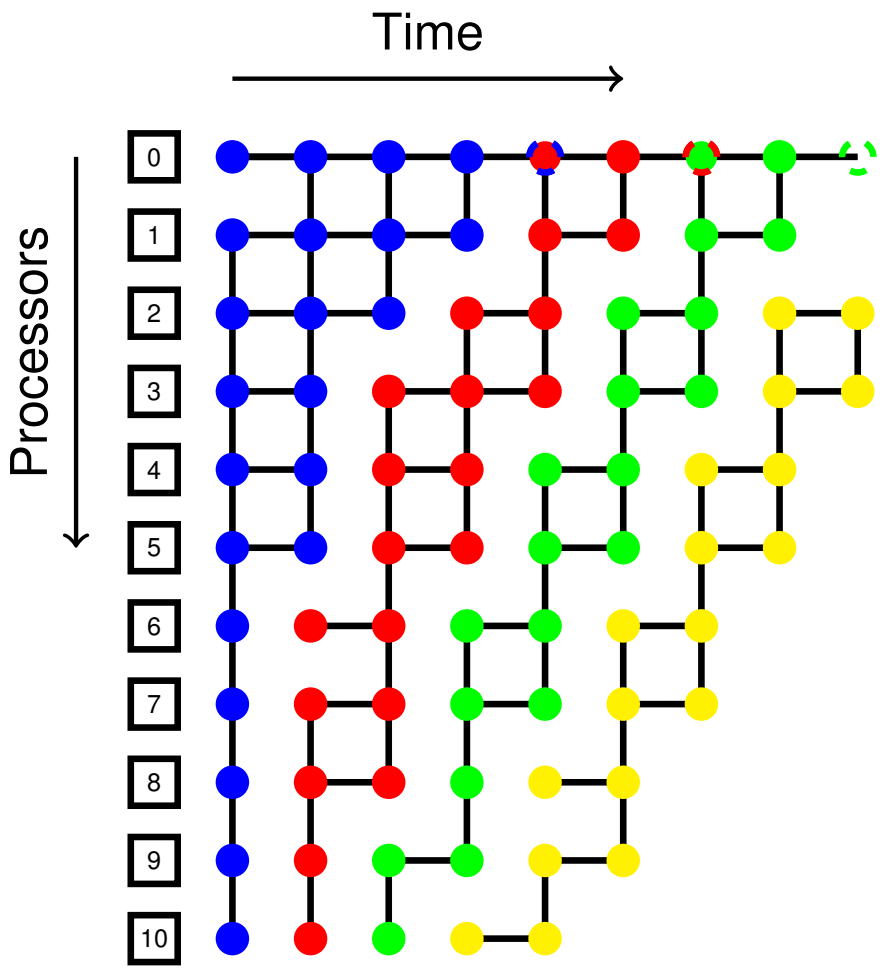

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



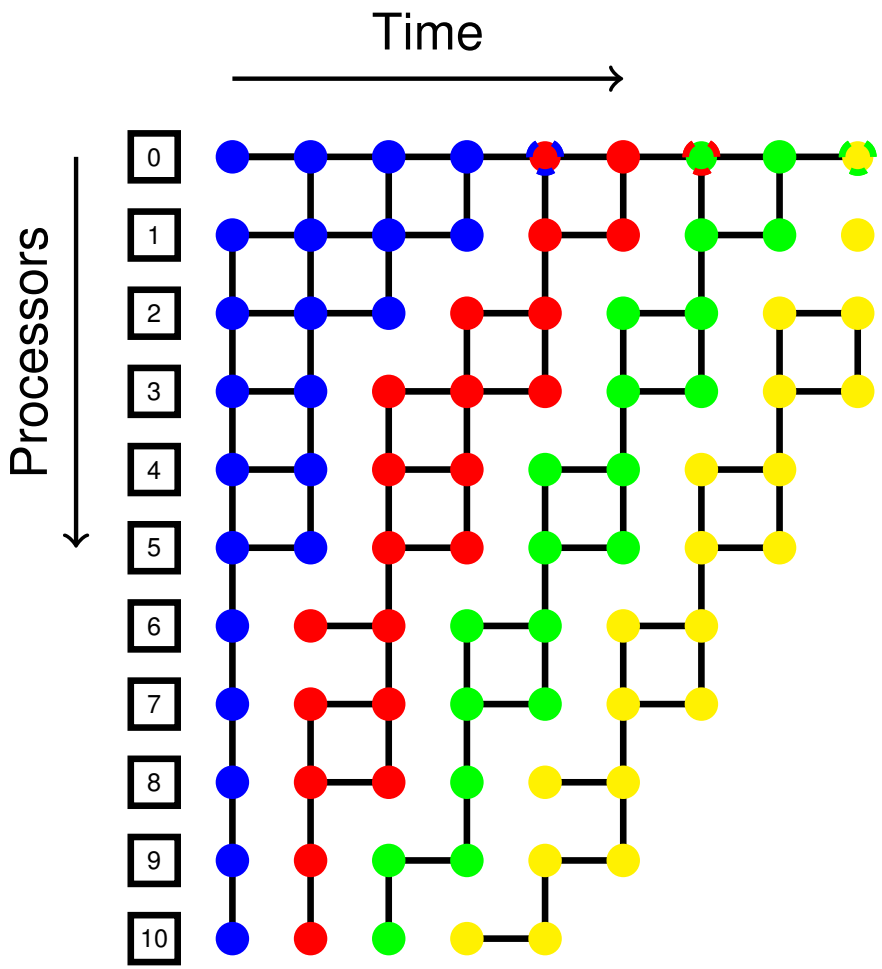

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



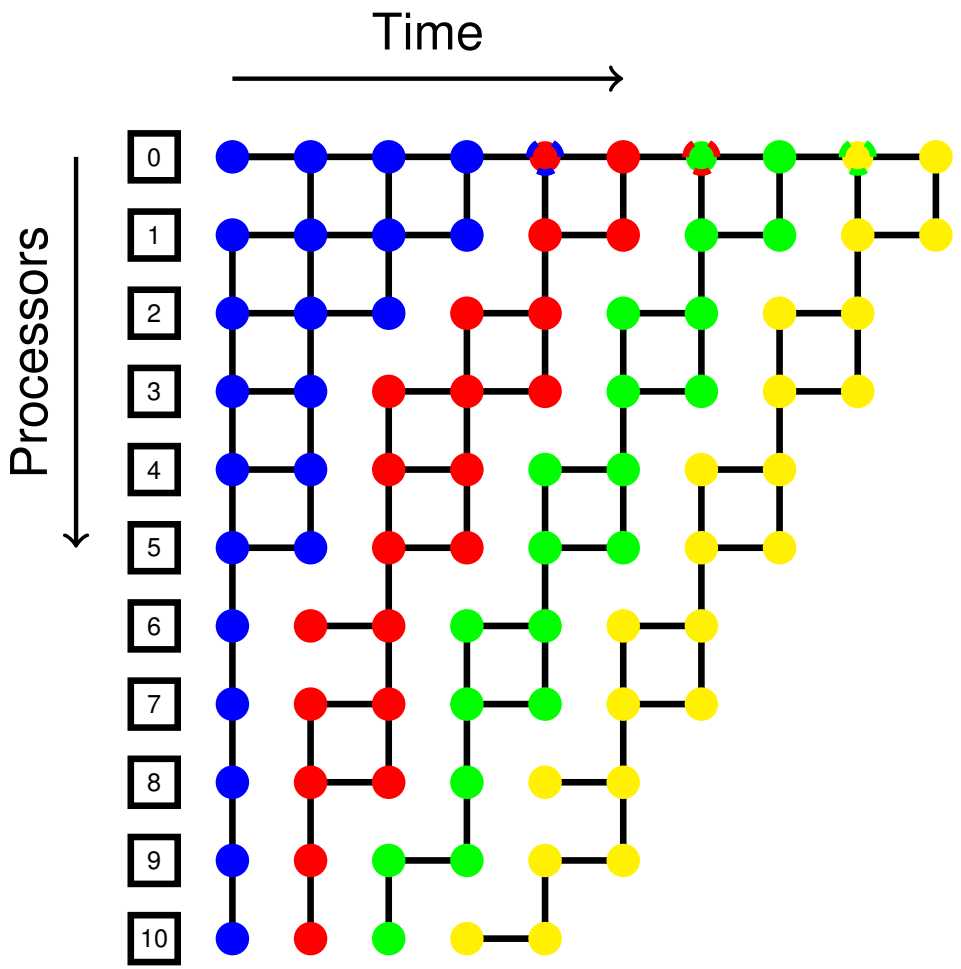

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree



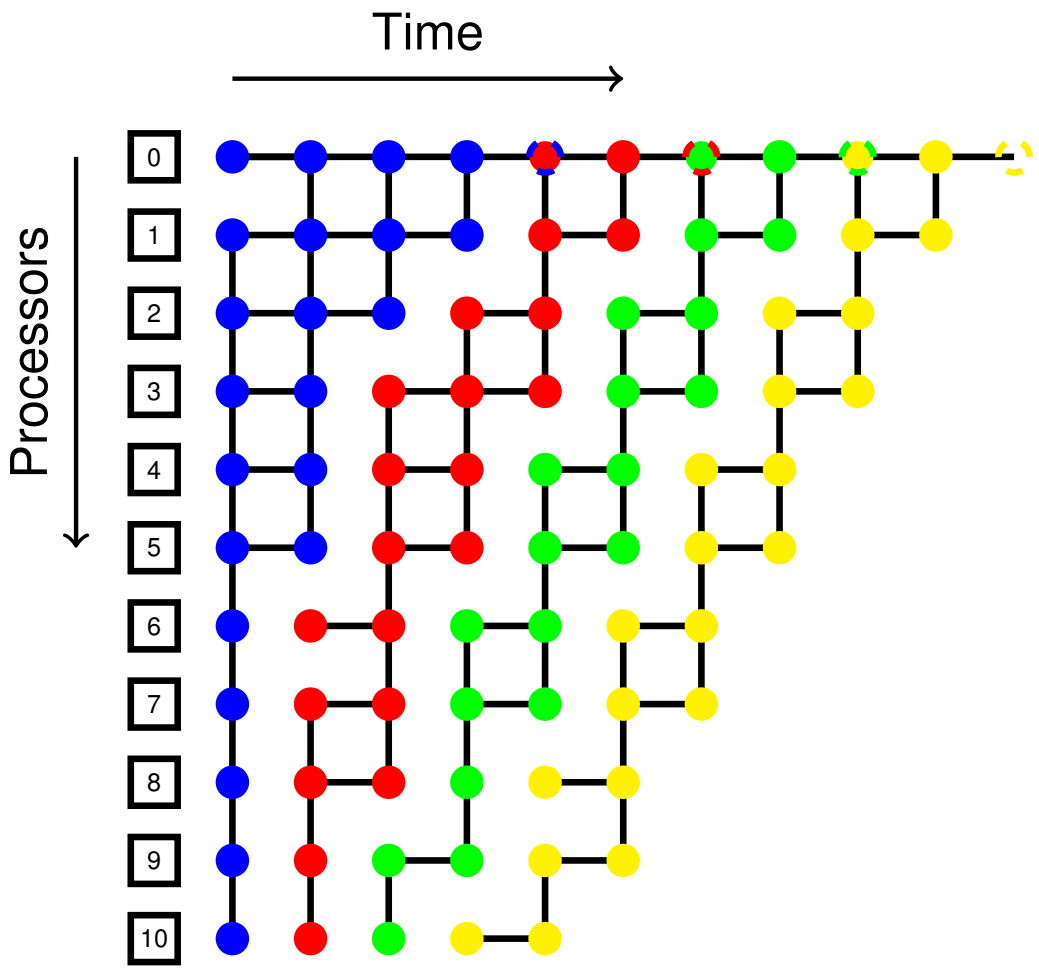

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Greedy Tree




- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

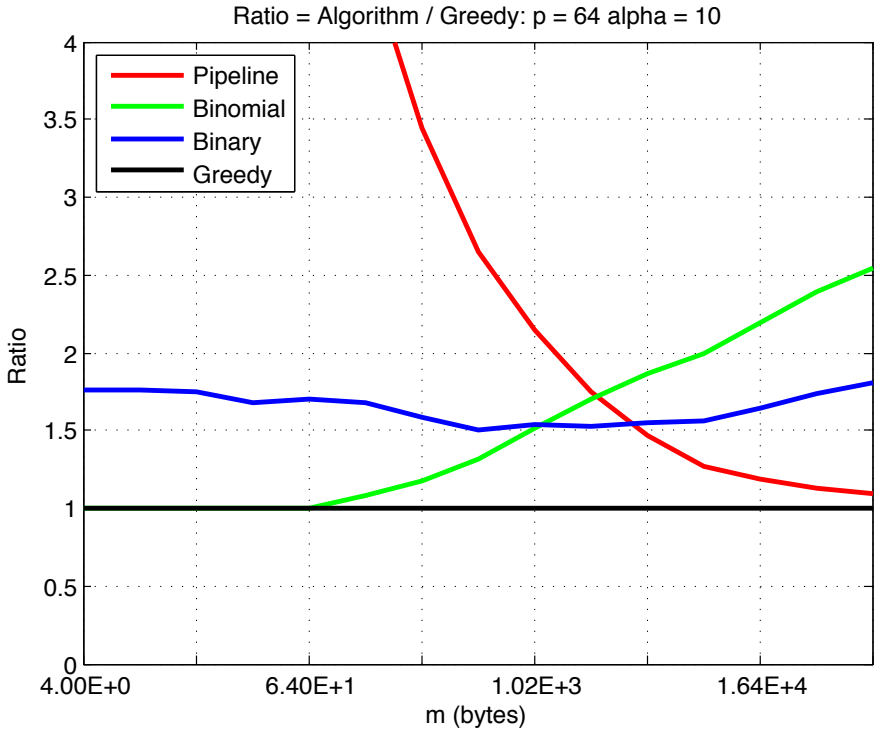
## Greedy Tree

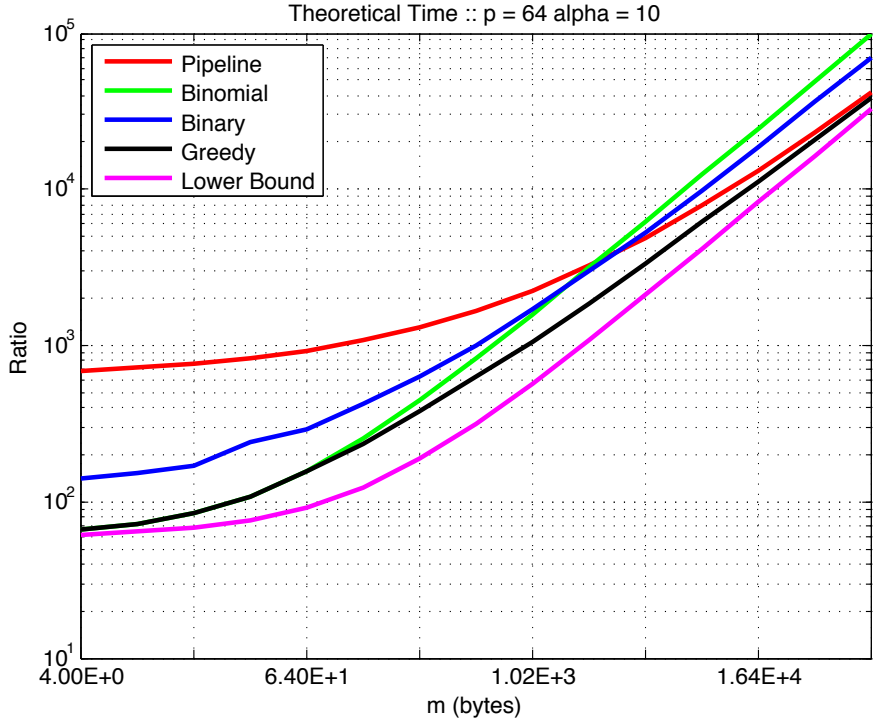


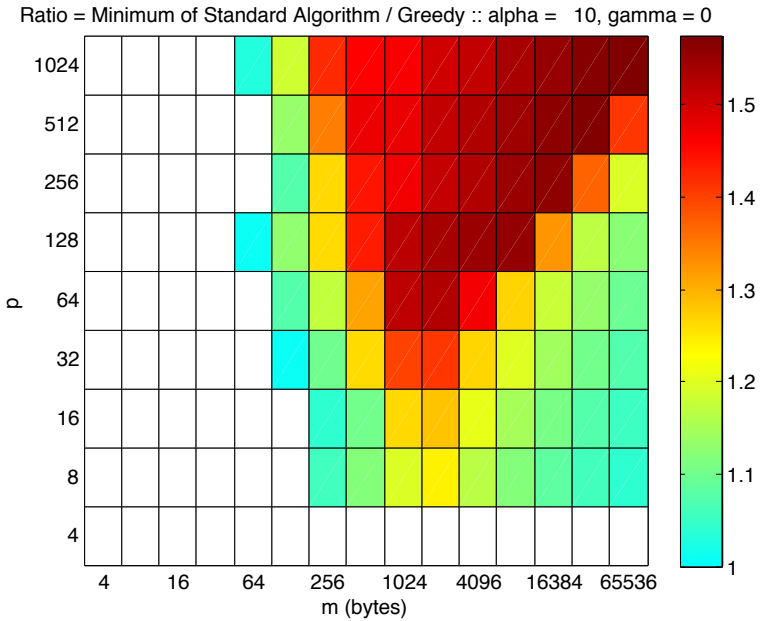

- Optimal for uniform segmentation.
- Motivated by greedy QR factorization scheme. [Cosnard and Robert ’86 [11]]

## Theoretical Results

- Each algorithm is tuned for optimal uniform segmentation.
- For given parameters, $p$, $m$, $\alpha$, $\beta$, $\gamma$, which algorithm is the best?



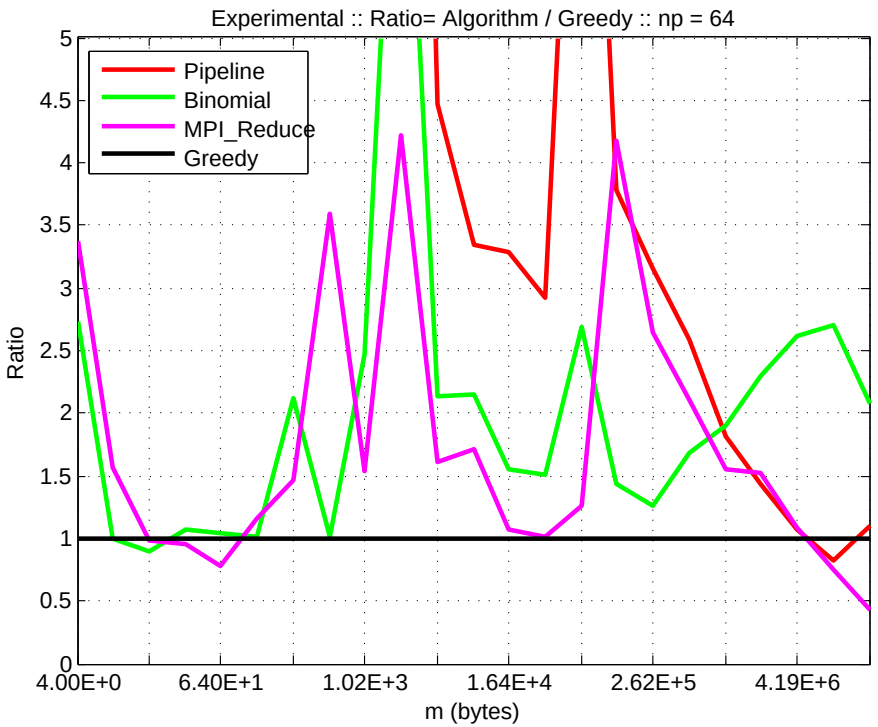



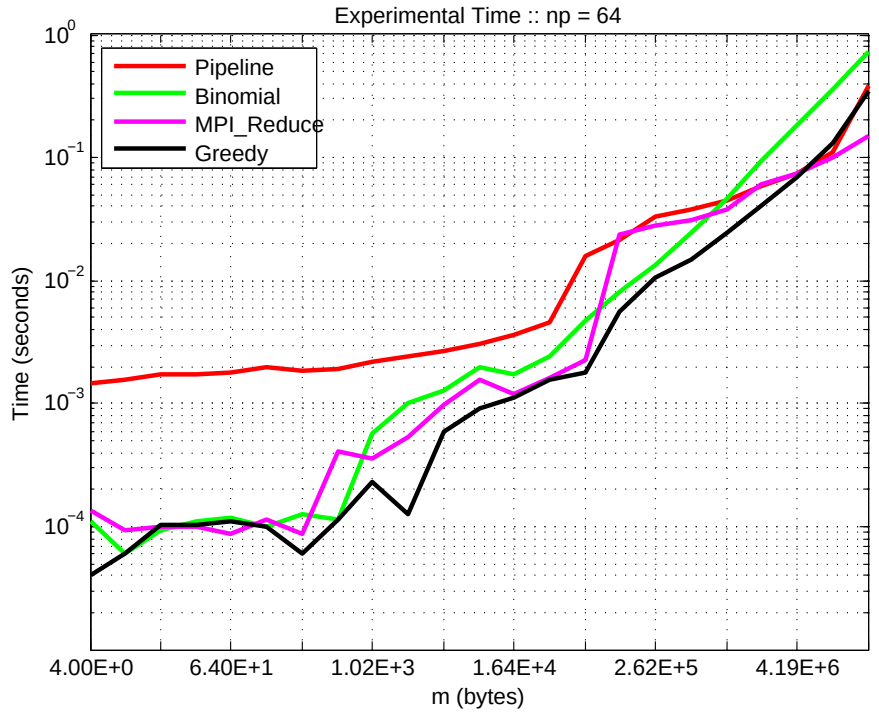




- $\alpha = 10$, $\beta = 1$, and $\gamma = 0$.
- For all parameters checked greedy was the best algorithm.

# Numerical Results

- Janus supercomputer.
- Each algorithm was implemented with OpenMPI v1.4.3 point-to-point functions MPI_Send and MPI_Recv.



## Nonuniform Segmentation

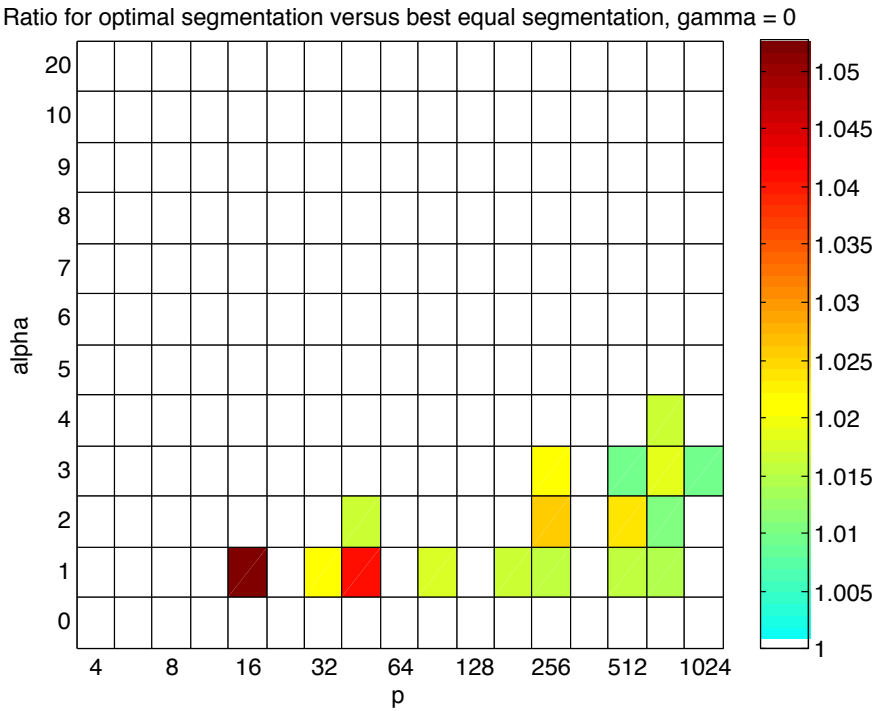
Why does segmentation have to be uniform?

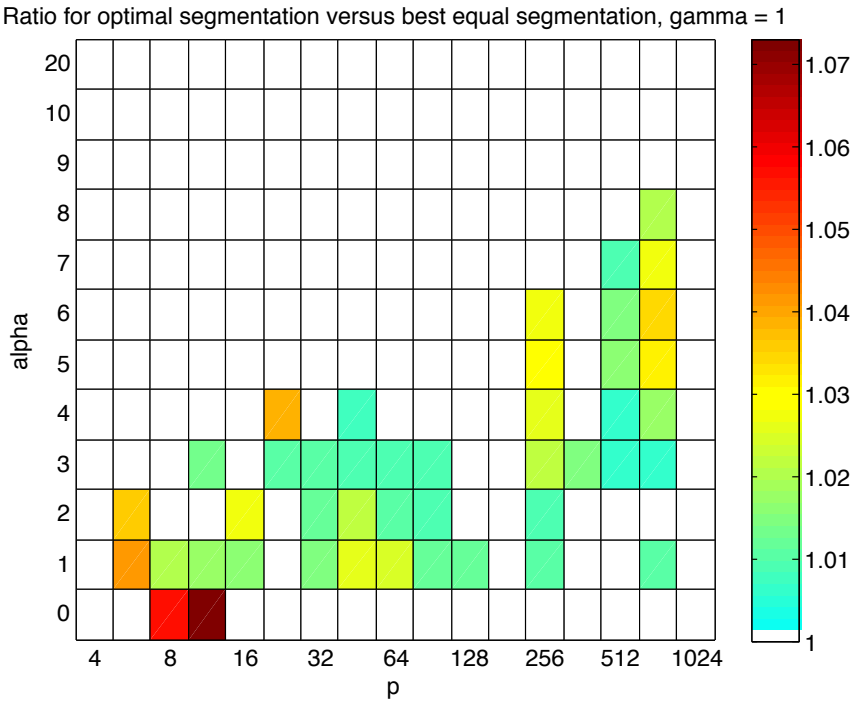
- Experiment: Fix the message size to $m = 10$ and check all possible segmentations.
- Results for greedy
  - 61 of the 986 total trials where optimized by a nonuniform segmentation.
  - The maximum improvement of nonuniform versus uniform segmentation was 7.3%.
  - Of the 61 trials optimized by a nonuniform segmentation the average improvement was 2%.
- For pipeline, all trials where optimized by uniform segmentation.

# Nonuniform Segmentation

Sample Segmentations:

| Parameters | Percent | Best Uniform | Optimal |
|---|---|---|---|
| $p = 12, \alpha = 0, \beta = 1, \gamma = 1$ | 7.3% | (1,1,1,1,1,1,1,1,1,1,1,1) | (2,2,1,1,1,1,1,1,1,1) |
| $p = 48, \alpha = 1, \beta = 1, \gamma = 1$ | 2.6% | (2,2,2,2,2) | (3,2,2,2,1)<br>(2,2,1,2,2,1) |
| $p = 256, \alpha = 5, \beta = 1, \gamma = 1$ | 3.0% | (4,4,2) | (5,3,2) |



## Conclusion

- Compared the greedy reduction with three standard algorithms.
- Greedy was the best theoretically.
- Most improvement is for medium sized messages (1Kb - 1Mb).
- Nonuniform segmentation is consider.
- Greedy is optimized by nonuniform segmentation for some machine parameters.

Questions?

# References


Amotz Bar-Noy and Shlomo Kipnis.

Broadcasting multiple messages in simultaneous send/receive systems.
*Discrete Applied Mathematics*, 55(2):95 – 105, 1994.

Amotz Bar-Noy, Shlomo Kipnis, and Baruch Schieber.

Optimal multiple message broadcasting in telephone-like communication systems.
*Discrete Appl. Math.*, 100(1-2):1–15, March 2000.

O. Beaumont, A. Legrand, L. Marchal, and Y. Robert.

Pipelining broadcasts on heterogeneous platforms.
*Parallel and Distributed Systems, IEEE Transactions on*, 16(4):300 – 313, April 2005.

O. Beaumont, L. Marchal, and Y. Robert.

Broadcast trees for heterogeneous platforms.
In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, page 80b, April 2005.

Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert.

Pipelining broadcasts on heterogeneous platforms.
In *International Parallel and Distributed Processing Symposium IPDPS'2004. IEEE Computer.* Society Press, 2004.

Olivier Beaumont, Arnaud Legrand, Loris Marchal, and Yves Robert.

Pipelining broadcasts on heterogeneous platforms.
*IEEE Trans. Parallel Distrib. Syst.*, 16(4):300–313, April 2005.

# References


Olivier Beaumont, Loris Marchal, and Yves Robert.

Broadcast trees for heterogeneous platforms.
In *19th International Parallel and Distributed Processing Symposium IPDPS'2005*, pages 3–8. Society Press, 2005.

E. Chan, M. Heimlich, A. Purkayastha, and R. van de Geijn.

On optimizing collective communication.
In *CLUSTER '04 Proceedings of the 2004 IEEE International Conference on Cluster Computing*, pages 145–156, 2004.

Ernie Chan, Robert A. van de Geijn, William Gropp, and Rajeev Thakur.

Collective communication on architectures that support simultaneous communication over multiple links.
In *PPOPP*, pages 2–11, 2006.

Michel Cosnard, Jean-Michel Muller, and Yves Robert.

Parallel QR decomposition of a rectangular matrix.
*Numerische Mathematik*, 48:239–249, 1986.

Michel Cosnard and Yves Robert.

Complexity of parallel QR factorization.
*Journal of the A.C.M.*, 33(4):712–723, 1986.

William Gropp and Ewing Lusk.

Reproducible measurements of mpi performance characteristics.
In *Proceedings of the 6th European PVM/MPI Users' Group Meeting on Recent Advances in PVM and MPI*, pages 11–18. Springer-Verlag, 1999.

R. Hockney.

The communication challenge for MPP: Intel Paragon and Meiko CS-2.
*Parallel Computing*, 20(3):389–398, 1994.

# References


A. Legrand, L. Marchal, and Y. Robert.

Optimizing the steady-state throughput of scatter and reduce operations on heterogeneous platforms.
*J. Parallel Distrib. Comput.*, 65(12):1497–1514, December 2005.

J.J. Modi and M.R.B. Clarke.

An alternative Givens ordering.
*Numerische Mathematik*, 43:83–90, 1984.

J. Pjesivac-Grbović, T. Angskun, G. Bosilca, G.E. Fagg, E. Gabriel, and J.J. Dongarra.

Performance analysis of MPI collective operations.
In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, page 8 pp., april 2005.

Rolf Rabenseifner.

Optimization of collective reduction operations.
In *International Conference on Computational Science*, pages 1–9, 2004.

Rajeev Thakur and William Gropp.

Improving the performance of collective operations in mpich.
In *PVM/MPI*, pages 257–267, 2003.

Rajeev Thakur, Rolf Rabenseifner, and William Gropp.

Optimization of collective communication operations in mpich.
*IJHPCA*, 19(1):49–66, 2005.

Jesper Larsson Träff and Andreas Ripke.

Optimal broadcast for fully connected processor-node networks.
*J. Parallel Distrib. Comput.*, 68(7):887–901, 2008.

# References

Sathish S. Vadhiyar, Graham E. Fagg, and Jack Dongarra.

Automatically tuned collective communications.

In *Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)*, Supercomputing '00, Washington, DC, USA, 2000. IEEE Computer Society.