Tiptoe: A Compositional Real-Time Operating System

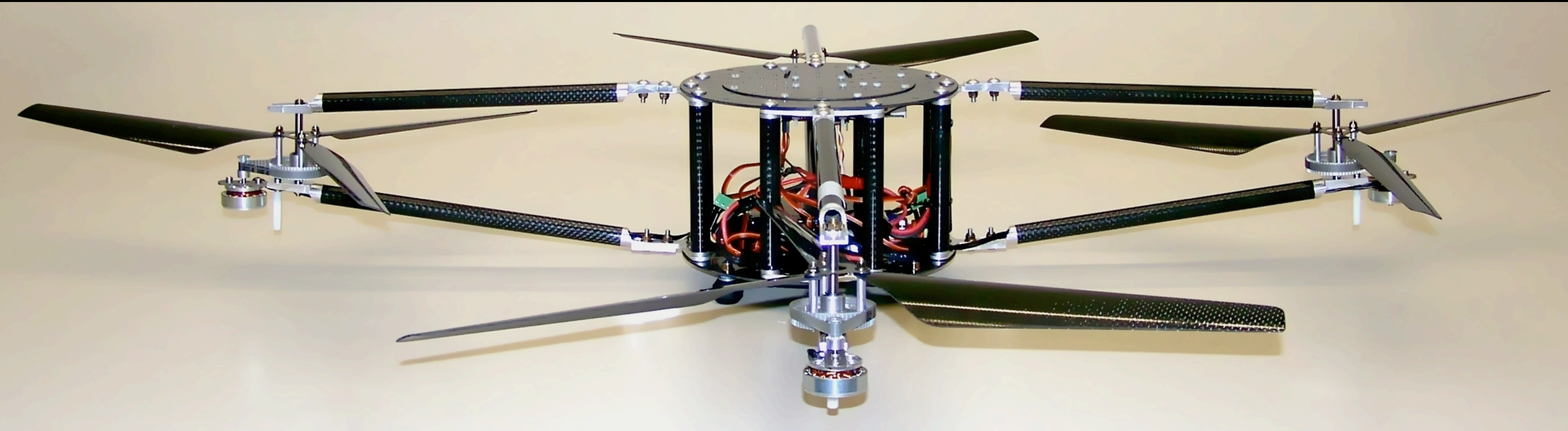
Christoph Kirsch
Universität Salzburg

UC Irvine, CS Colloquium
March 2008

tiptoe.cs.uni-salzburg.at

- Silviu Craciunas* (Programming Model)
- Hannes Payer (Memory Management)
- Harald Röck (VM, Scheduling)
- Ana Sokolova* (Theoretical Foundation)
- Horst Stadler (I/O Subsystem)

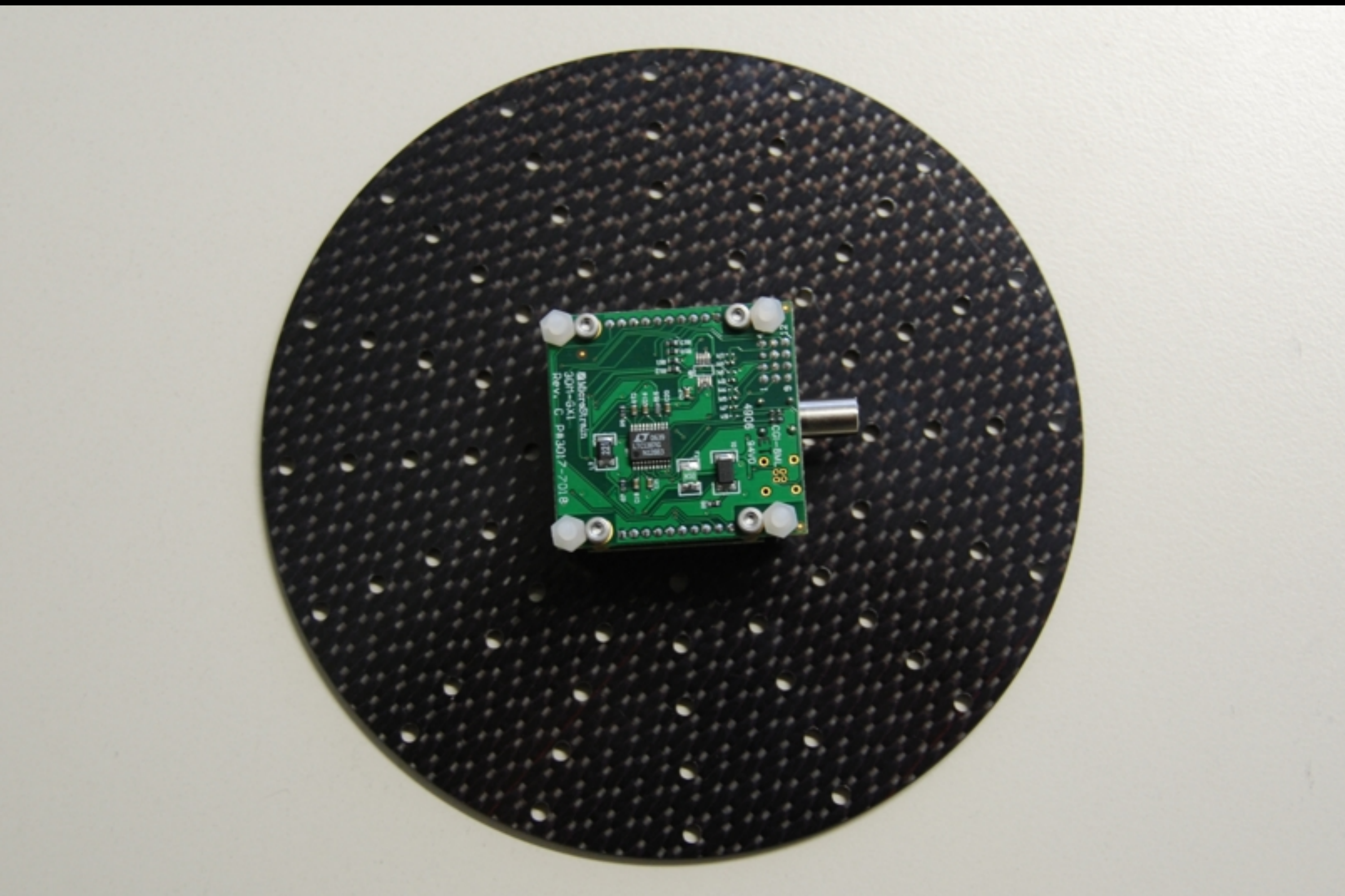
*Supported by Austrian Science Fund Project P18913-N15



The JAviator

javiator.cs.uni-salzburg.at

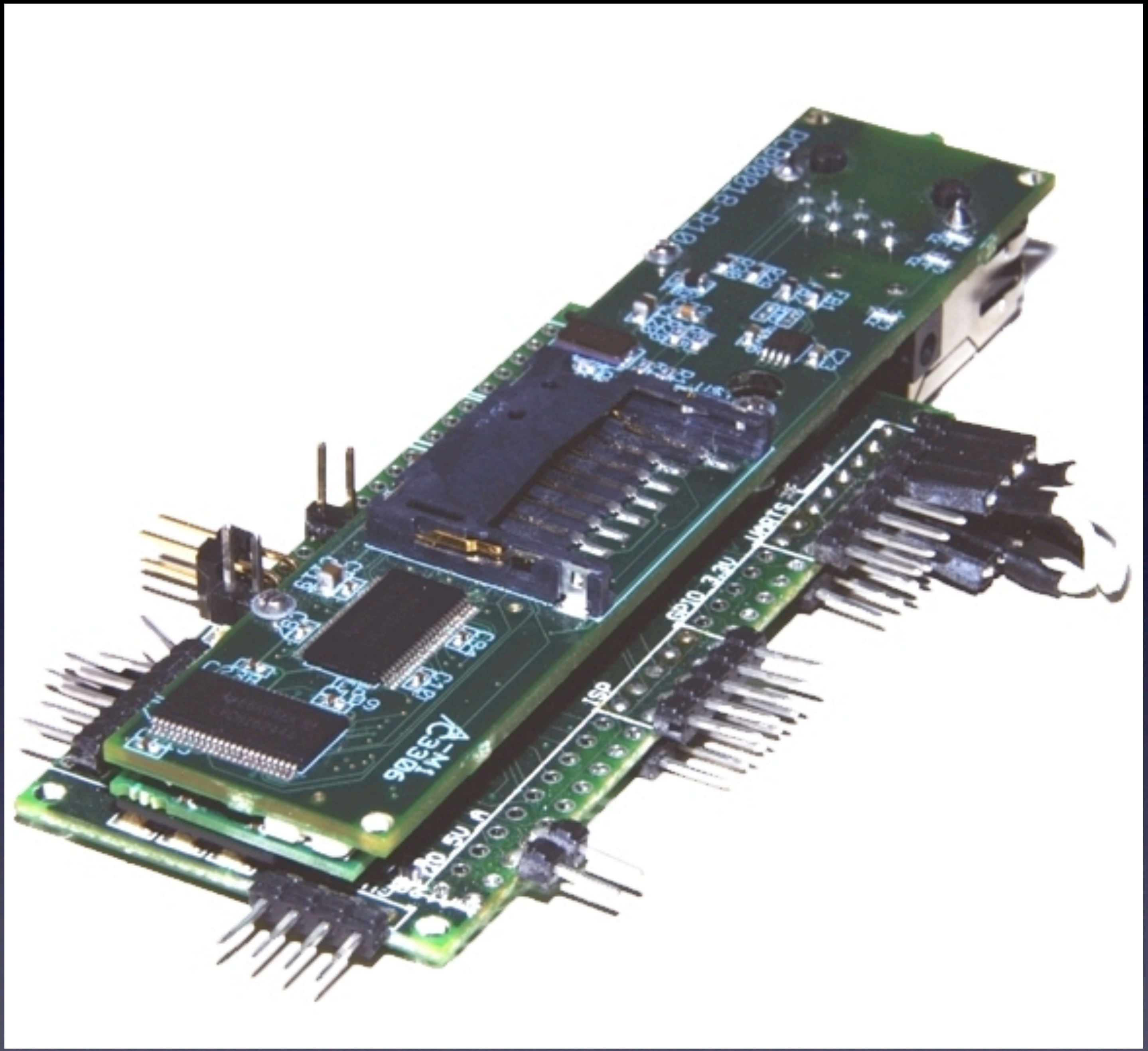
Quad-Rotor Helicopter



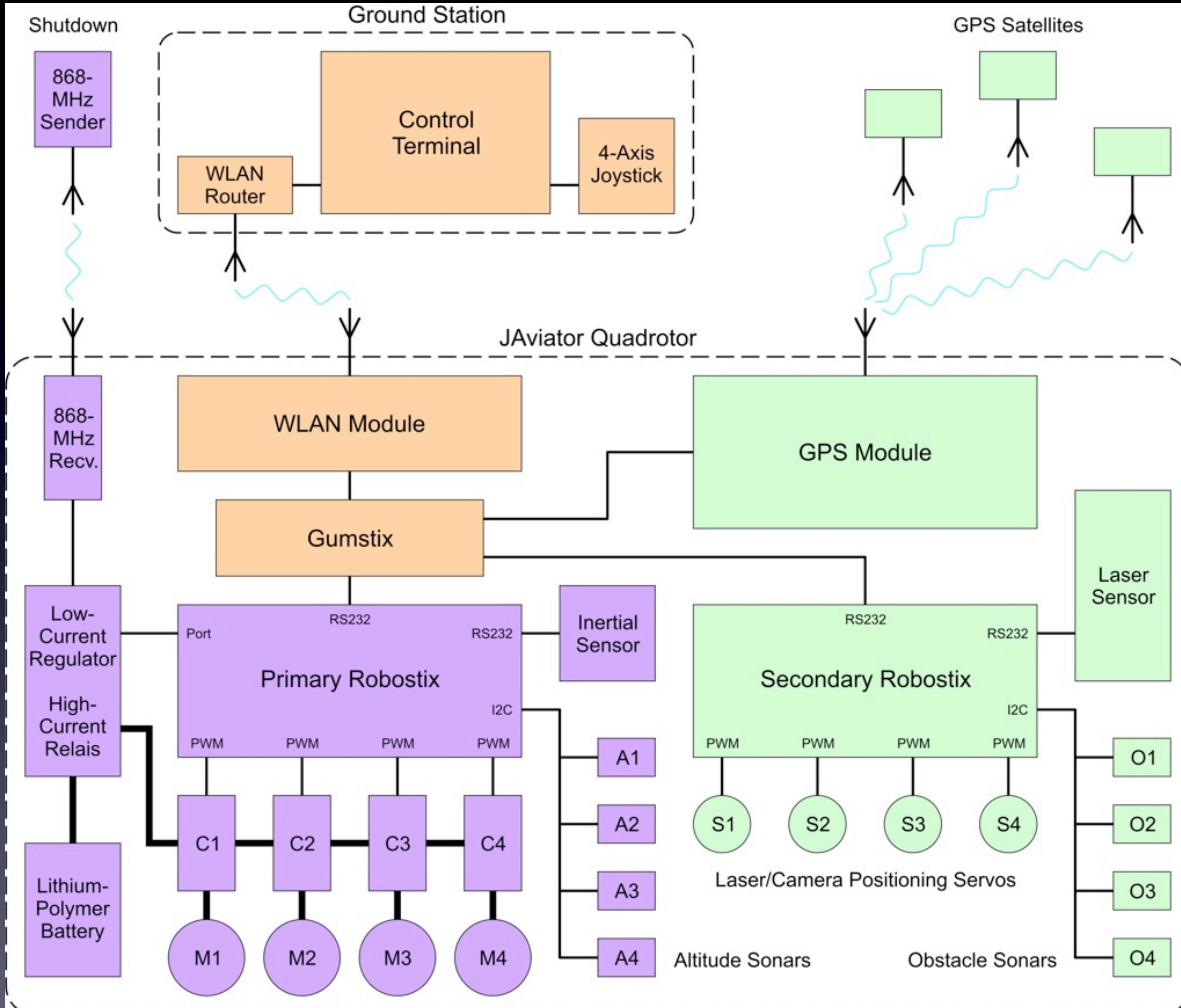
Gyro

Propulsion

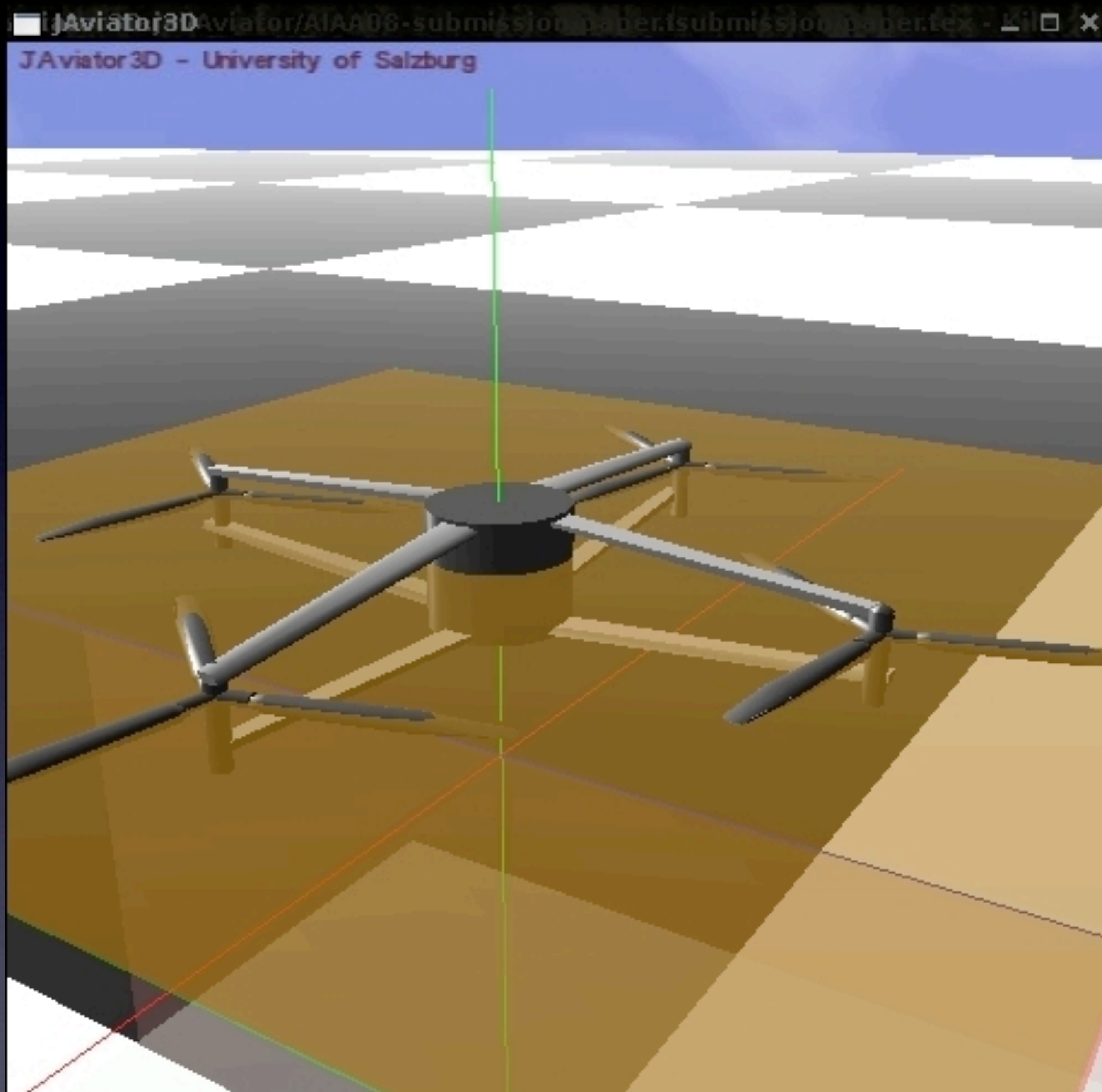
Gumstix



600MHz XScale, 128MB RAM, WLAN, Atmega uController

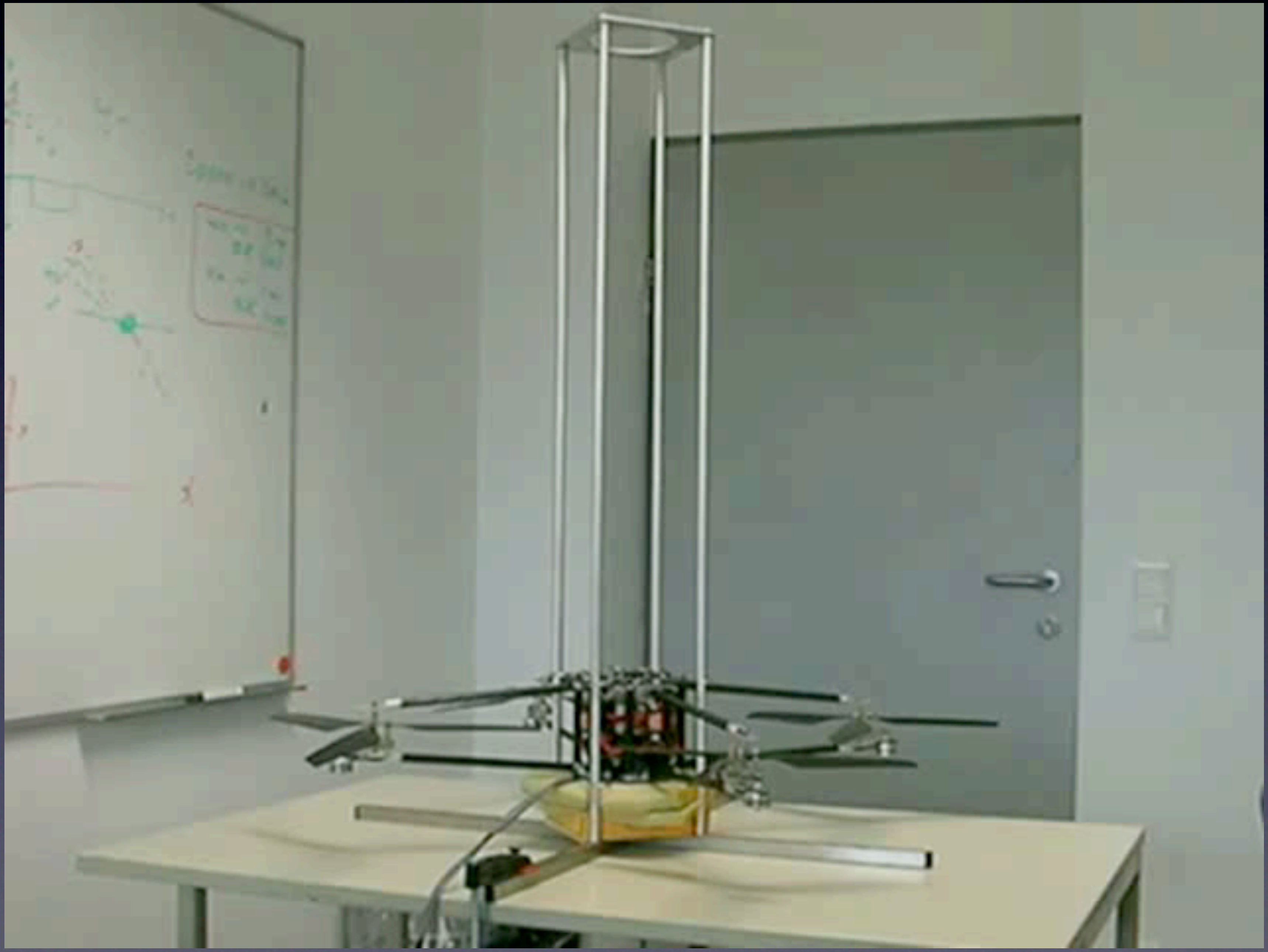

Shutdown
868-MHz Sender
Ground Station
WLAN Router
Control Terminal
4-Axis Joystick
GPS Satellites
JAviator Quadrotor
868-MHz Recv.
WLAN Module
GPS Module
Gumstix
Laser Sensor
Low-Current Regulator
High-Current Relais
Lithium-Polymer Battery
Port
RS232
RS232
Primary Robostix
I2C
PWM
PWM
PWM
PWM
Inertial Sensor
RS232
RS232
Secondary Robostix
I2C
PWM
PWM
PWM
PWM
C1
C2
C3
C4
M1
M2
M3
M4
A1
A2
A3
A4
Altitude Sonars
S1
S2
S3
S4
Laser/Camera Positioning Servos
O1
O2
O3
O4
Obstacle Sonars


JAviator Control Terminal
Yaw Limit
Joystick Mode
Direct Connection
Connect to Heli
Start Test Mode
Start Logging
Roll Limit
Yaw
deg
Front
Left
Hover
Right
Rear
Roll
deg
-1
-1
0
1
Offsets
1
33 Yaw Trim
Roll Trim 0
0
-1
0
Pitch
deg
Altitude
cm
351
353
Signals
352
351
Pitch Trim 0
Altitude Limit
Port Settings
Reset Needles
Shut down Heli
Key Assistance
About Terminal
Pitch Limit

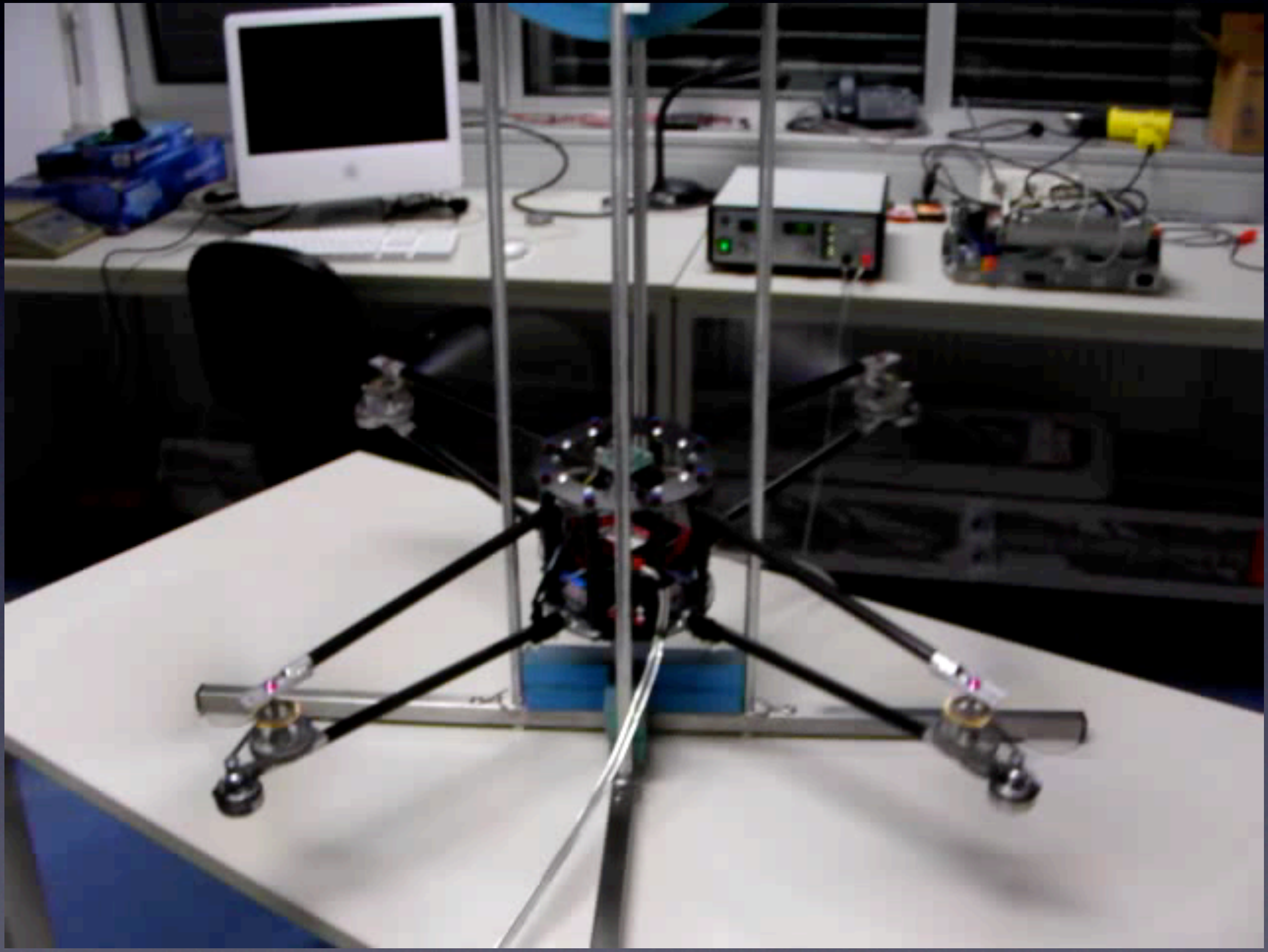

JAviator3D
JAviator3D - University of Salzburg


JAviator3D
JAviator3D - University of Salzburg

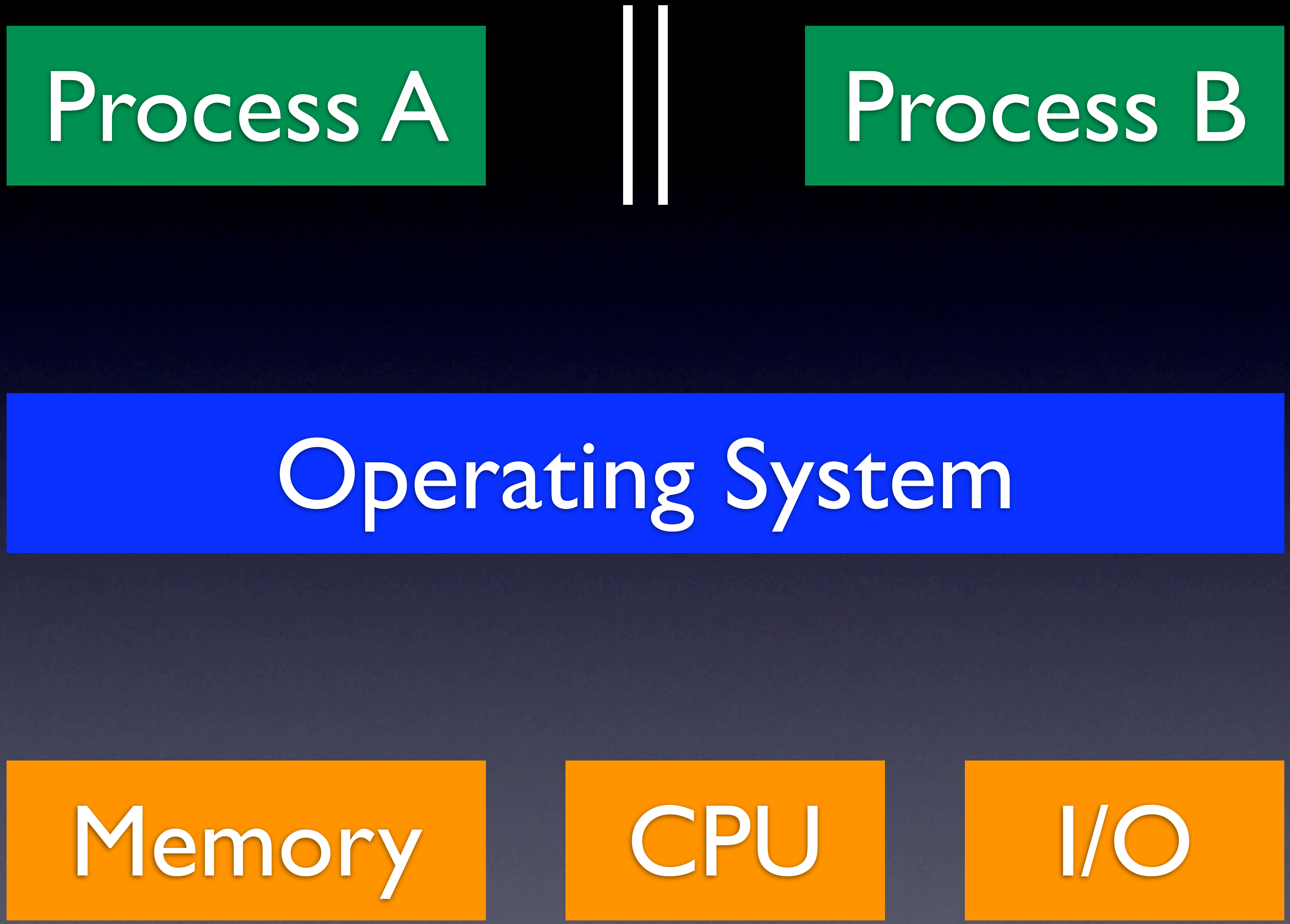
Oops



Flight Control



Free Flight


Process A
||
Process B
Operating System
Memory
CPU
I/O

“Theorem”

- (Compositionality) The time and space a software process needs to execute is determined by the process, not the system and not other software processes.
- (Predictability) The system can tell how much time and space is available without looking at any existing software processes.

“Corollary”

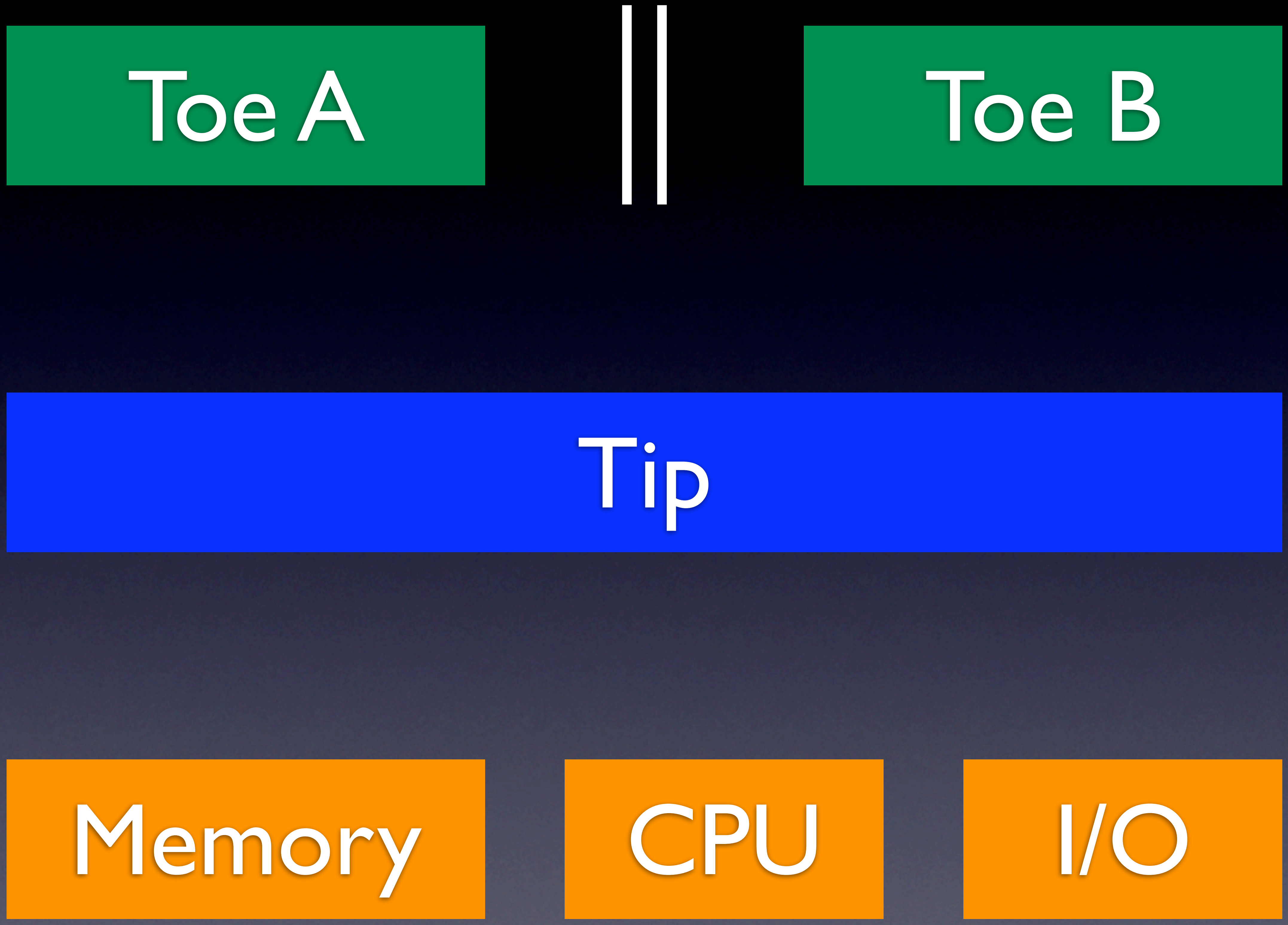
- (Memory) The time a software process takes to allocate and free a memory object is determined by the size of the object.
- (I/O) The time a software process takes to read input data and write output data is determined by the size of the data.

Programming Model

- A software process determines functional and non-functional behavior, for example:
- 1ms/100ms CPU time ($\neq$ 10ms/s)
- 4MB/2s memory allocation rate
- 1KB/10ms network bandwidth
- 10J/100ms energy consumption

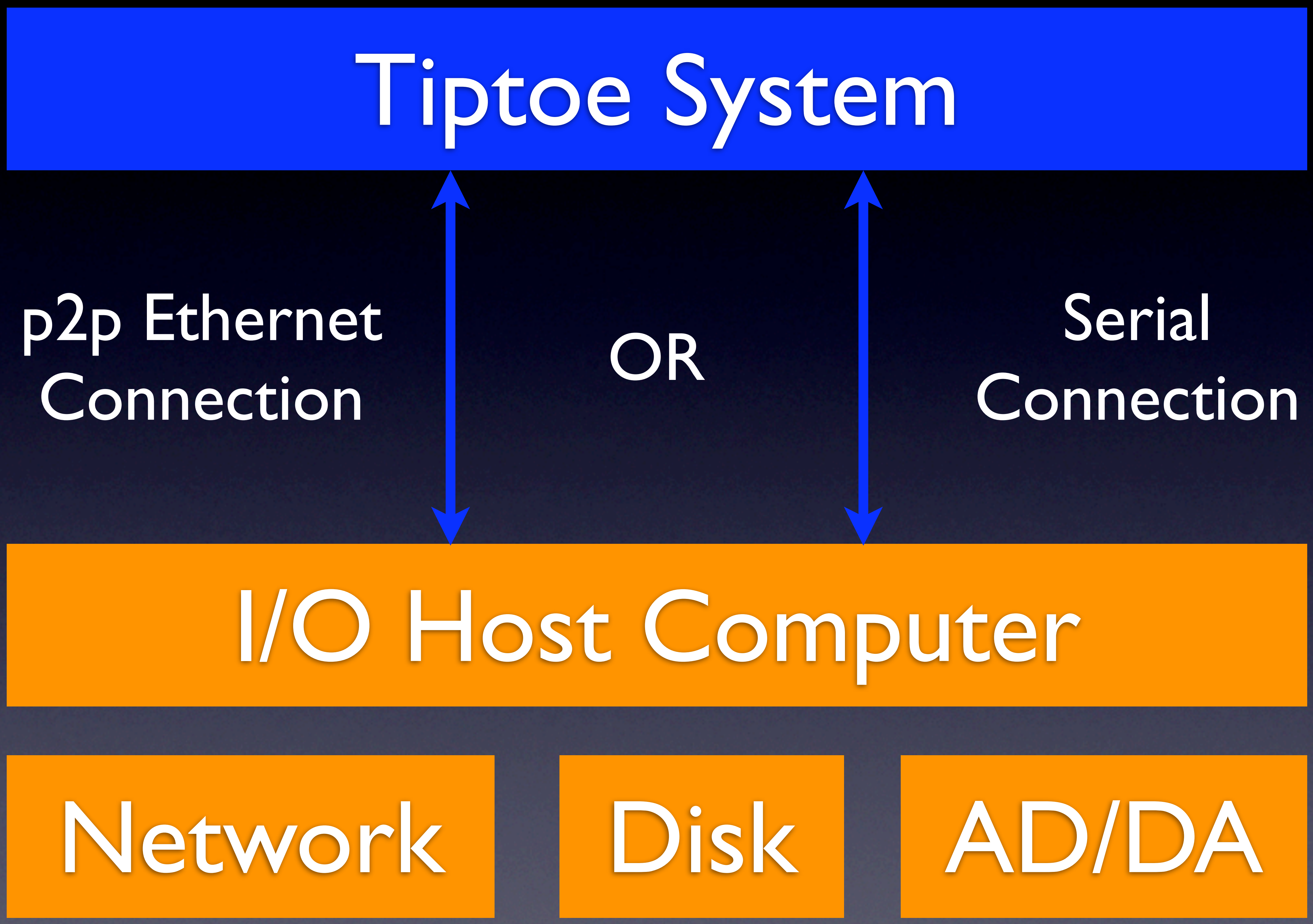
Outline

1. Memory Management
2. Concurrency Management
3. I/O Management


Toe A
||
Toe B
Tip
Memory
CPU
I/O

Outline

1. Memory Management
2. Concurrency Management
3. I/O Management


Tiptoe System
p2p Ethernet Connection
OR
Serial Connection
I/O Host Computer
Network
Disk
AD/DA

Outline

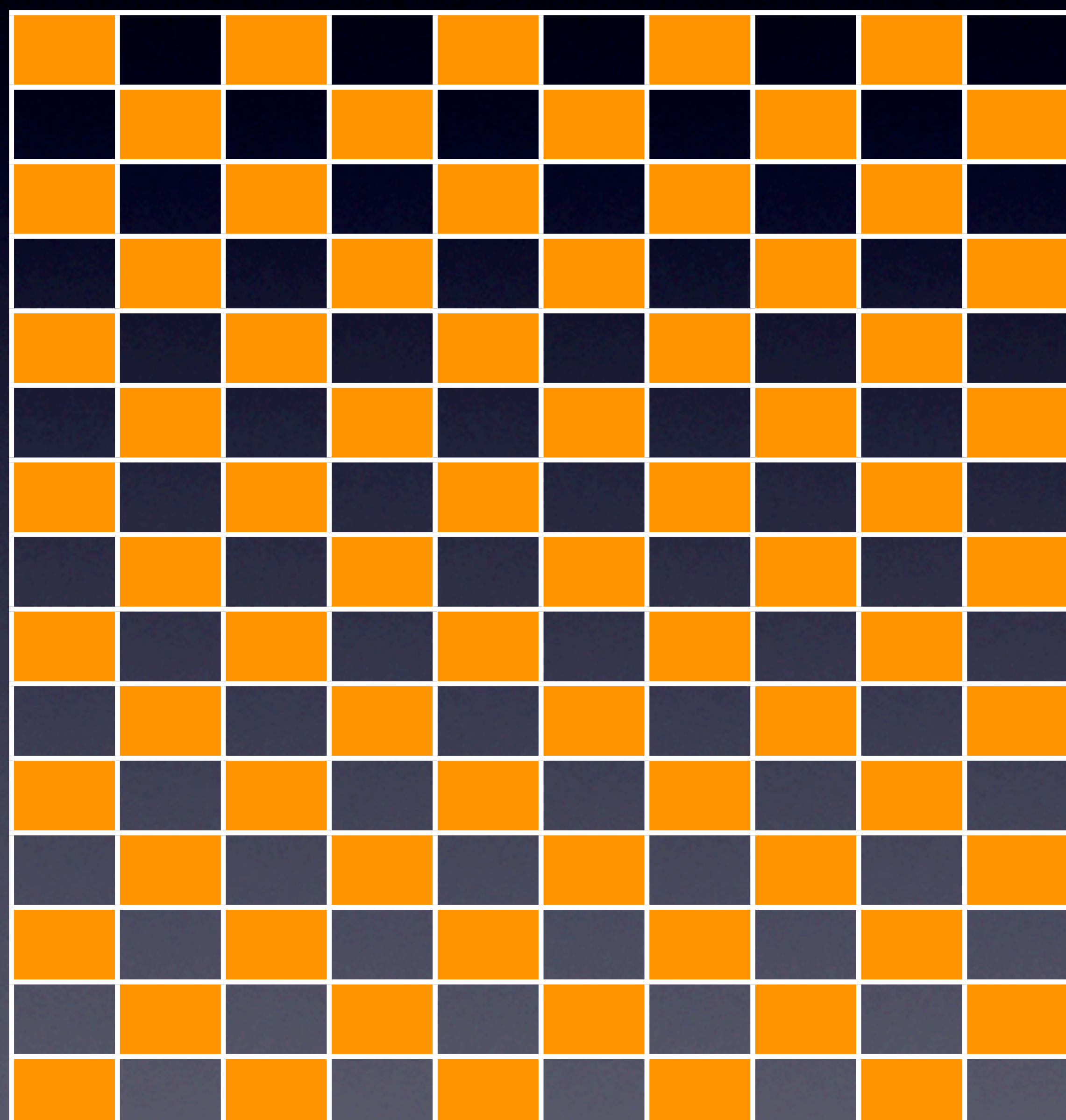
1. Memory Management
2. Concurrency Management
3. I/O Management

Goals

- malloc(n) takes at most TIME(n)
- free(n) takes at most TIME(n)
- access takes small constant time

- small and predictable memory fragmentation bound

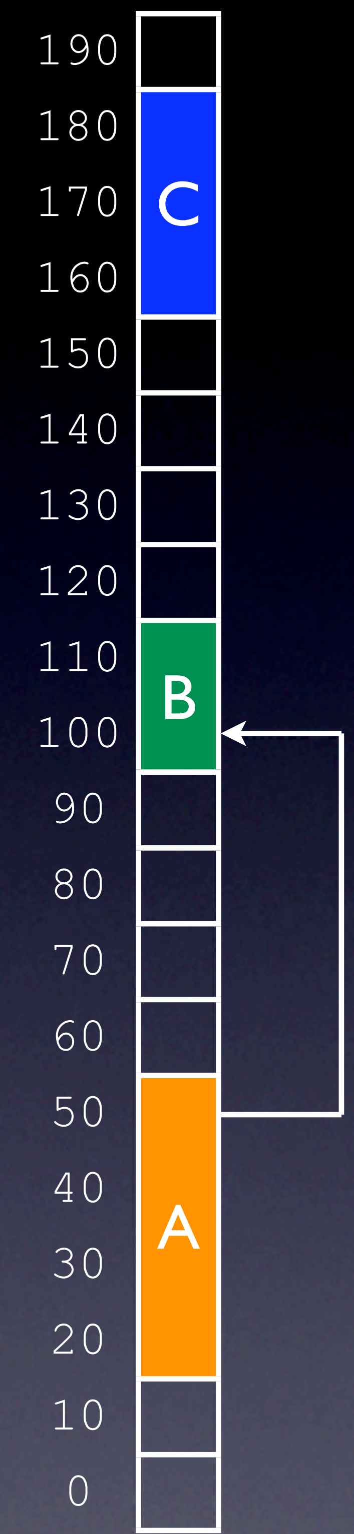
The Problem



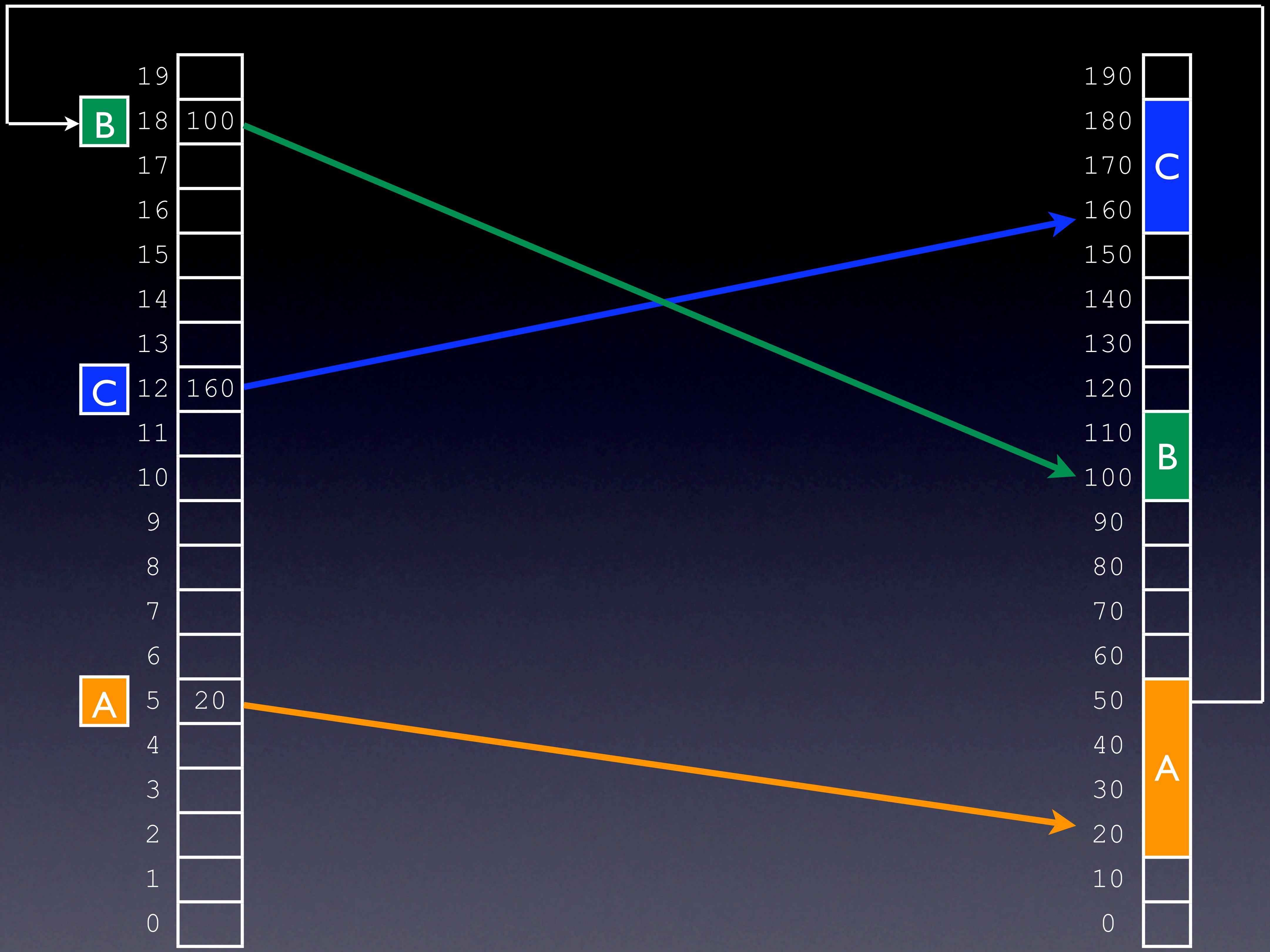
- Fragmentation
  - Compaction
- References
  - Abstract Space

Example:

- There are three objects
- Object A starts at address 20
- Object A needs 40 bytes
- B starts at 100, needs 20 bytes
- C starts at 160, needs 30 bytes
- A contains a reference to B



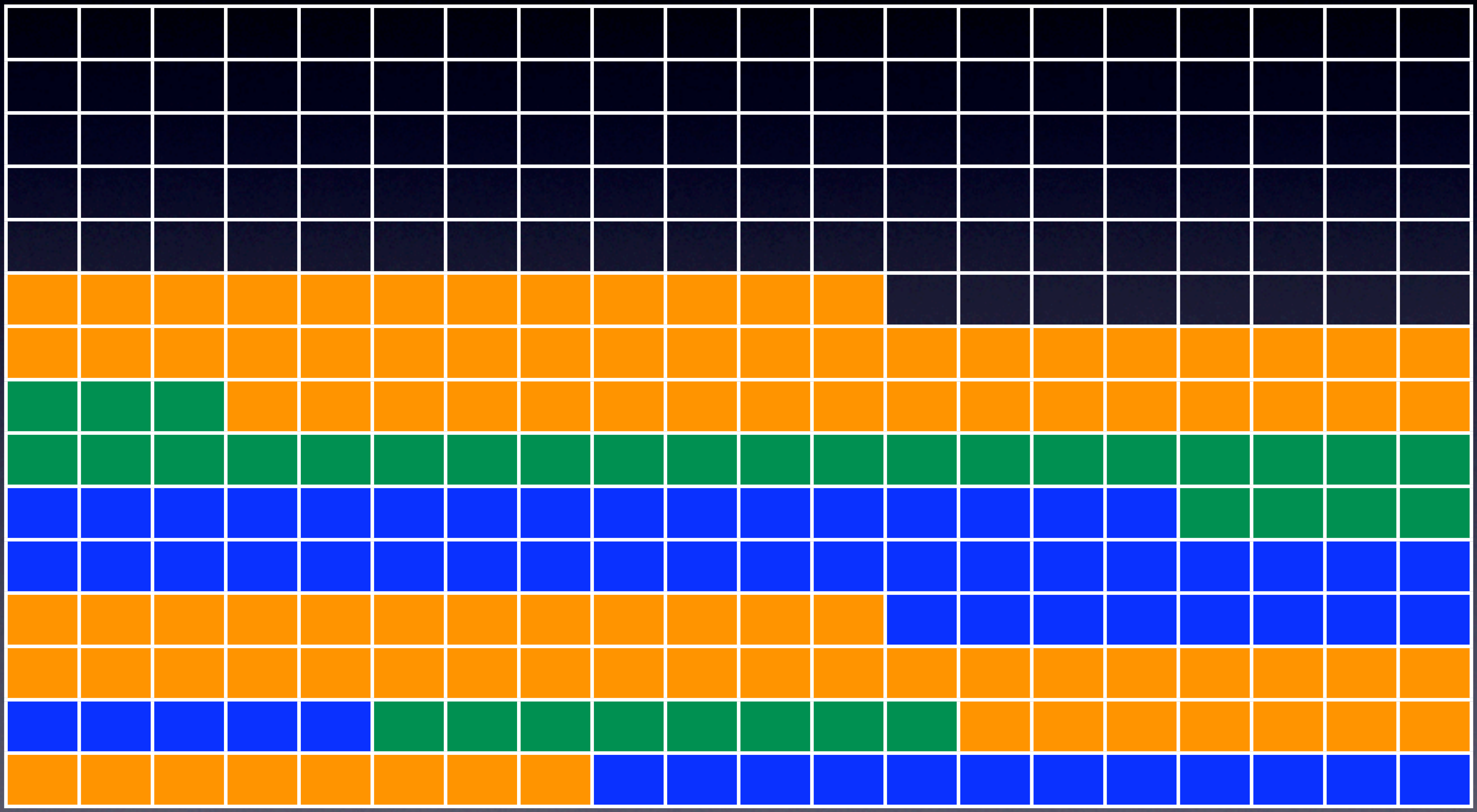
Memory



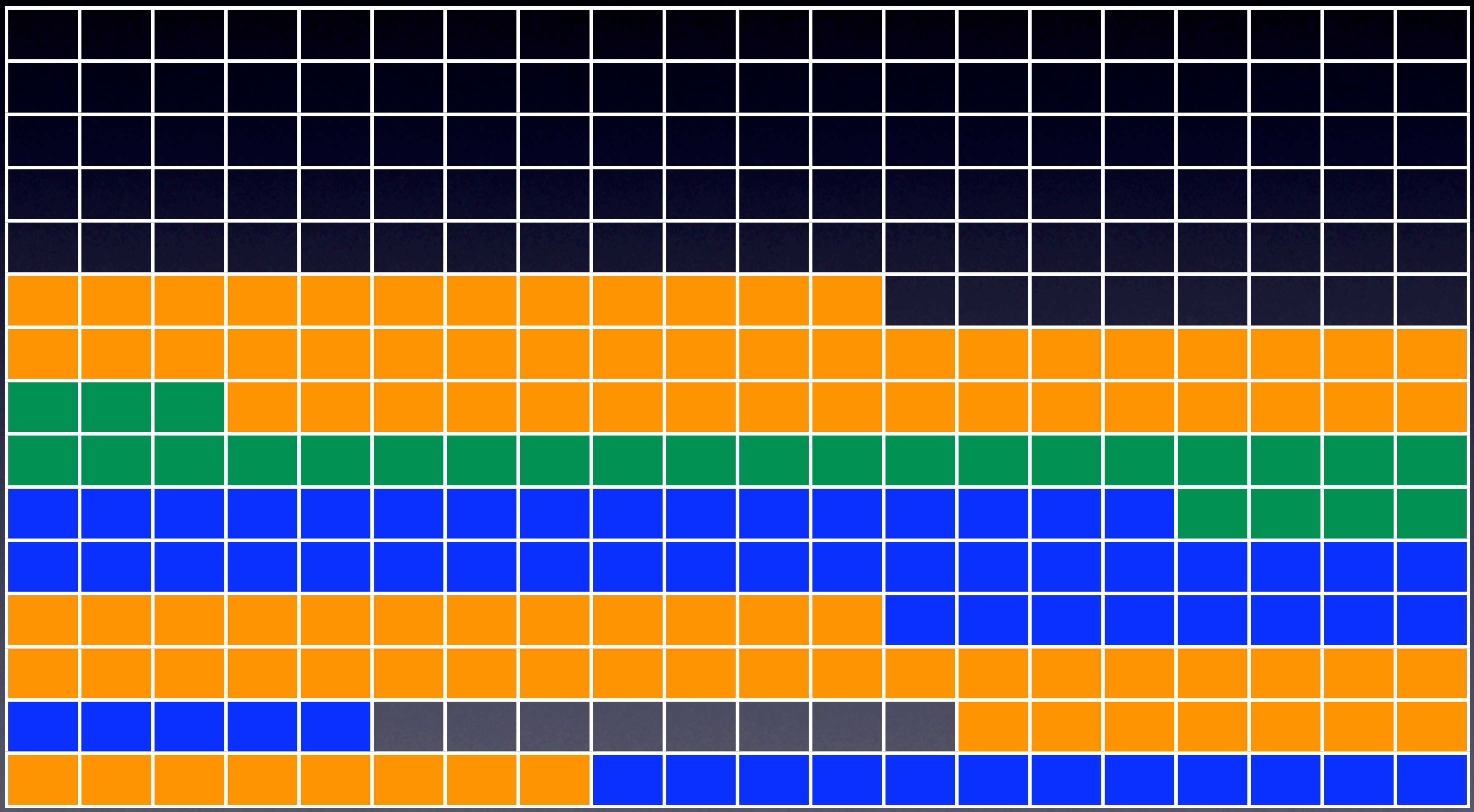
Abstract Space

Concrete Space

Keep It Compact?



Does Not Work!



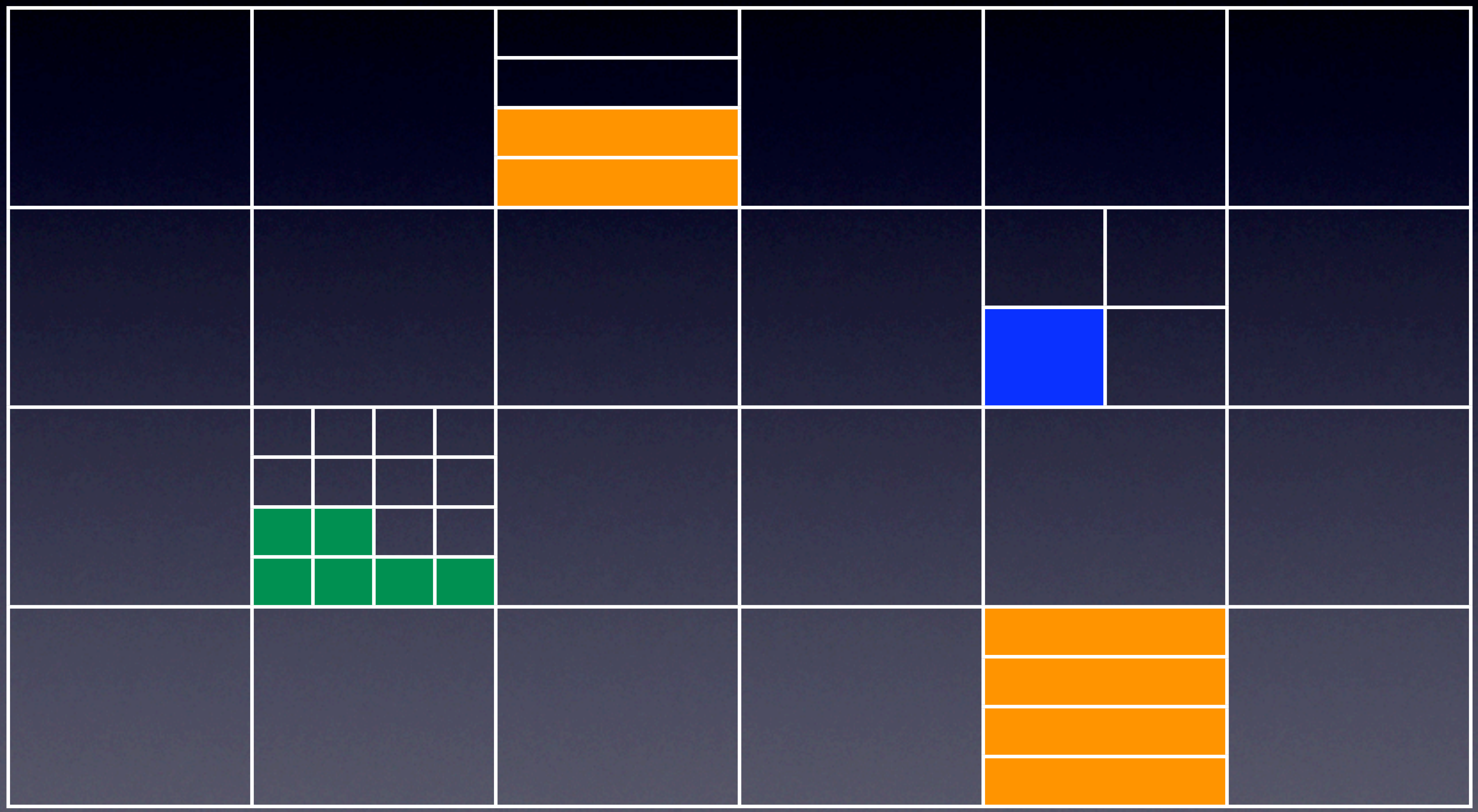
Trade-off Speed for
Memory Fragmentation

Keep Speed and
Memory Fragmentation
Bounded and Predictable

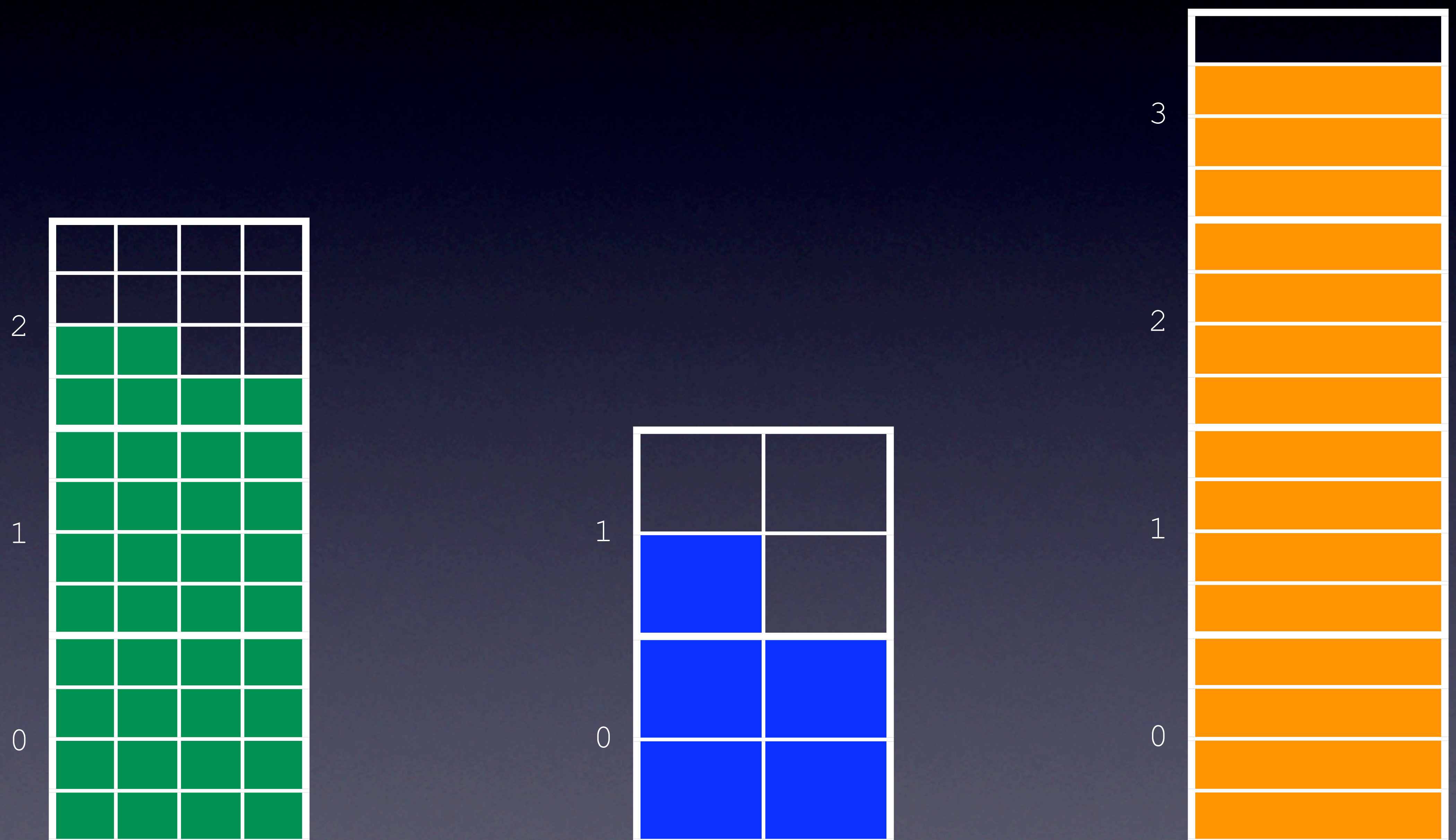
Partition Memory into Pages

16KB	16KB	16KB	16KB	16KB	16KB
16KB	16KB	16KB	16KB	16KB	16KB
16KB	16KB	16KB	16KB	16KB	16KB
16KB	16KB	16KB	16KB	16KB	16KB

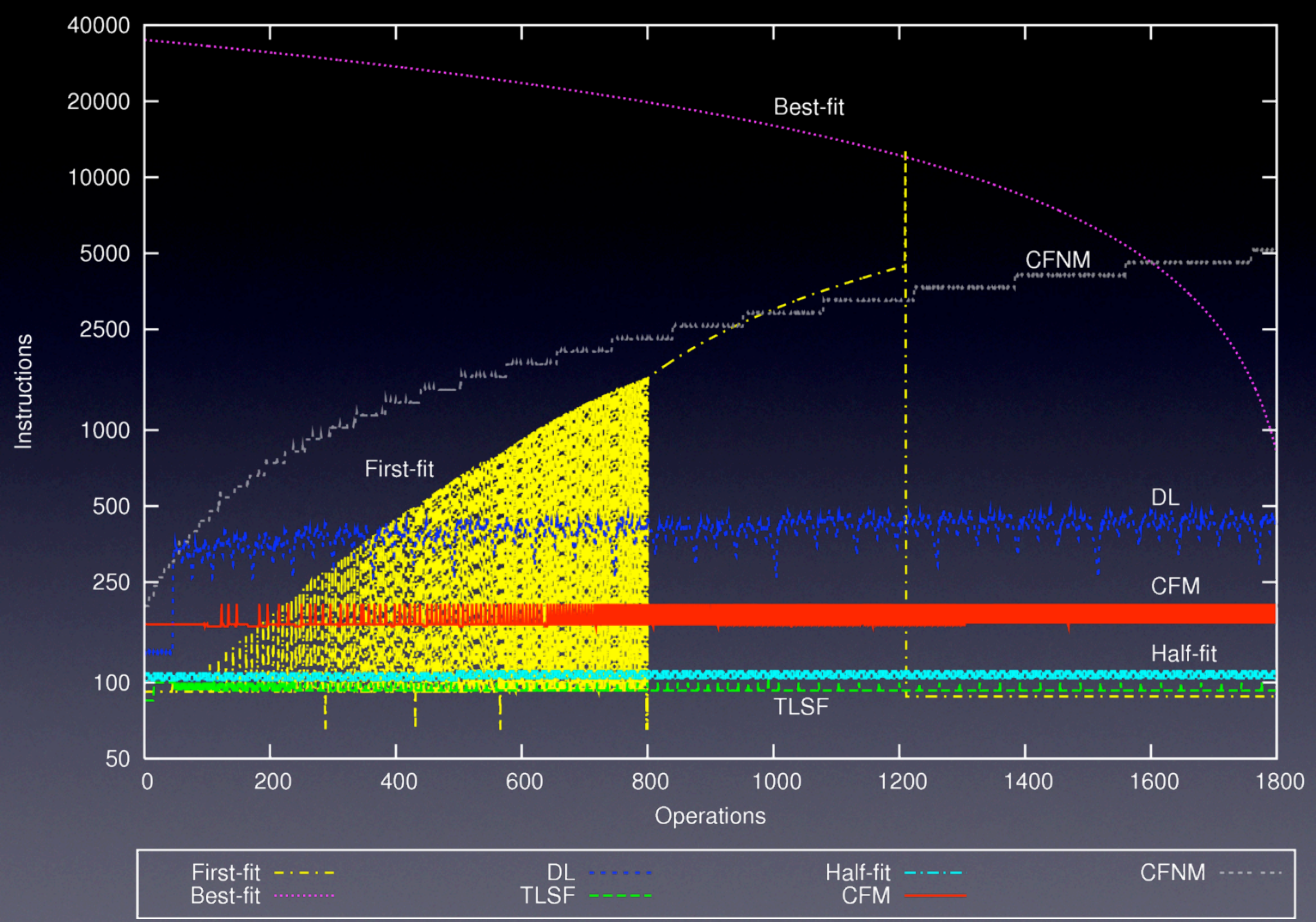
Partition Pages into Blocks



Size-Class Compact

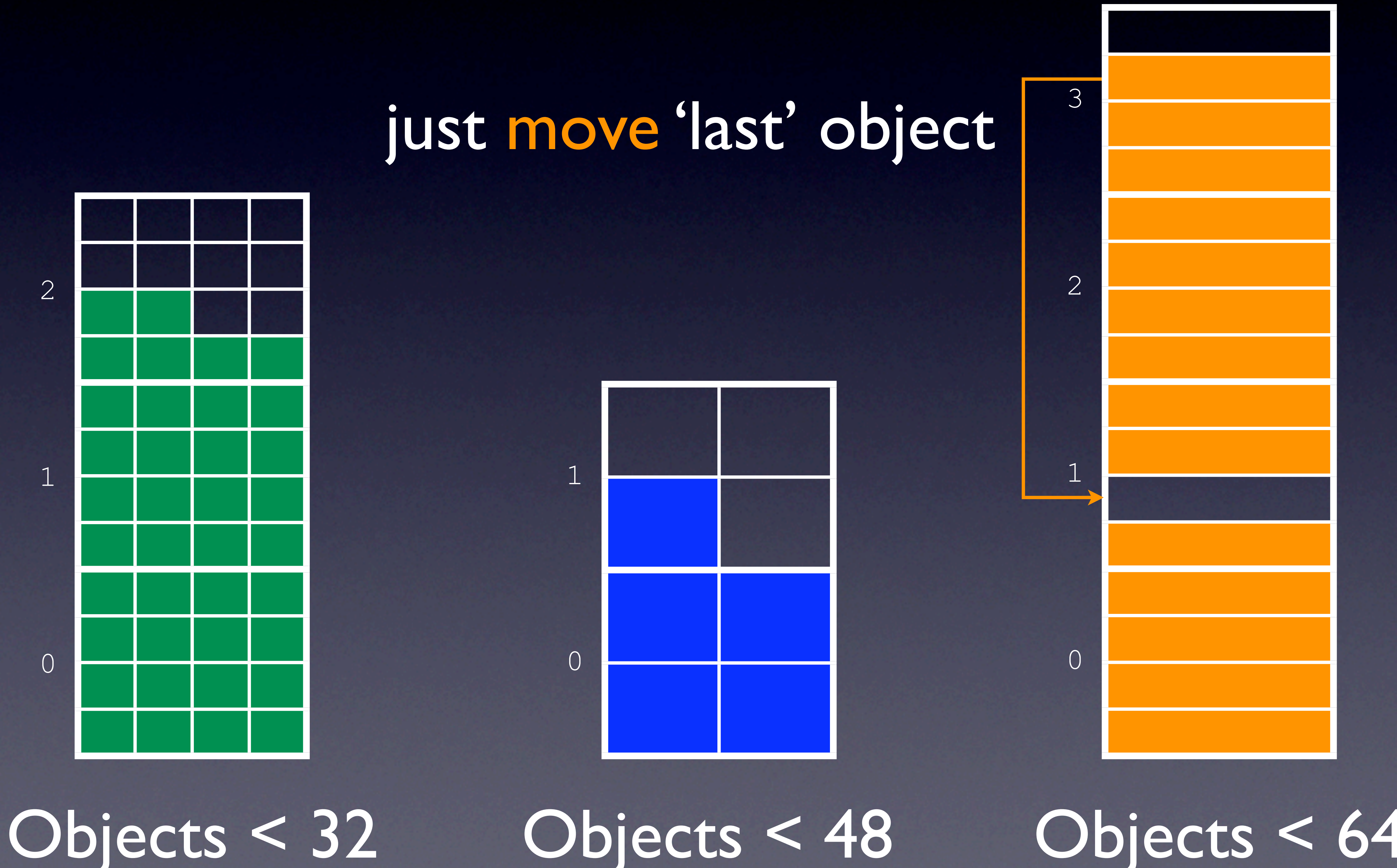


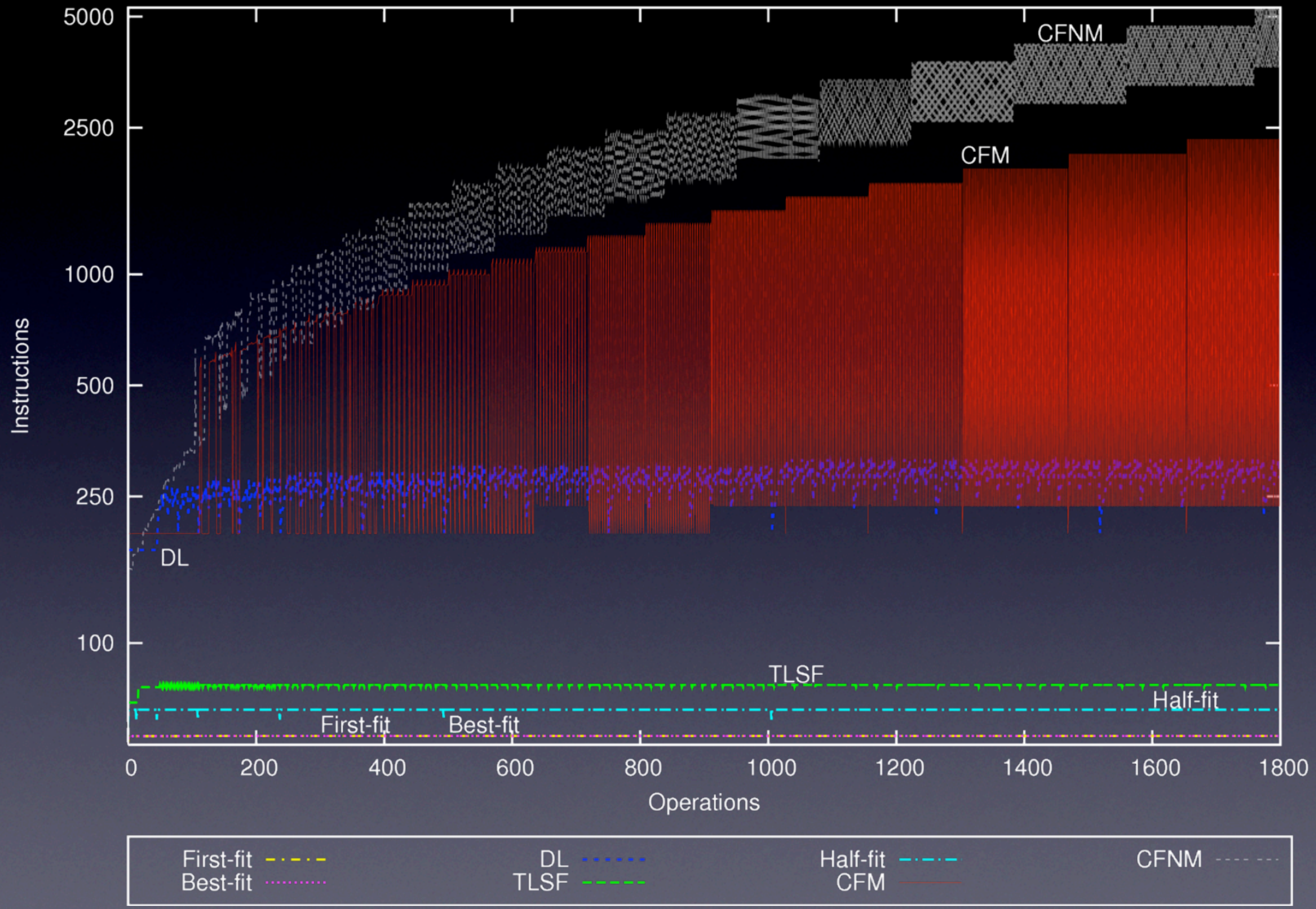
Objects < 32   Objects < 48   Objects < 64



40000
20000
10000
5000
2500
1000
500
250
100
50
Instructions
0
200
400
600
800
1000
1200
1400
1600
1800
Operations
Best-fit
CFNM
First-fit
DL
CFM
Half-fit
TLSF
First-fit
Best-fit
DL
TLSF
Half-fit
CFM
CFNM

"Compact-Fit" (Bounded Compaction)



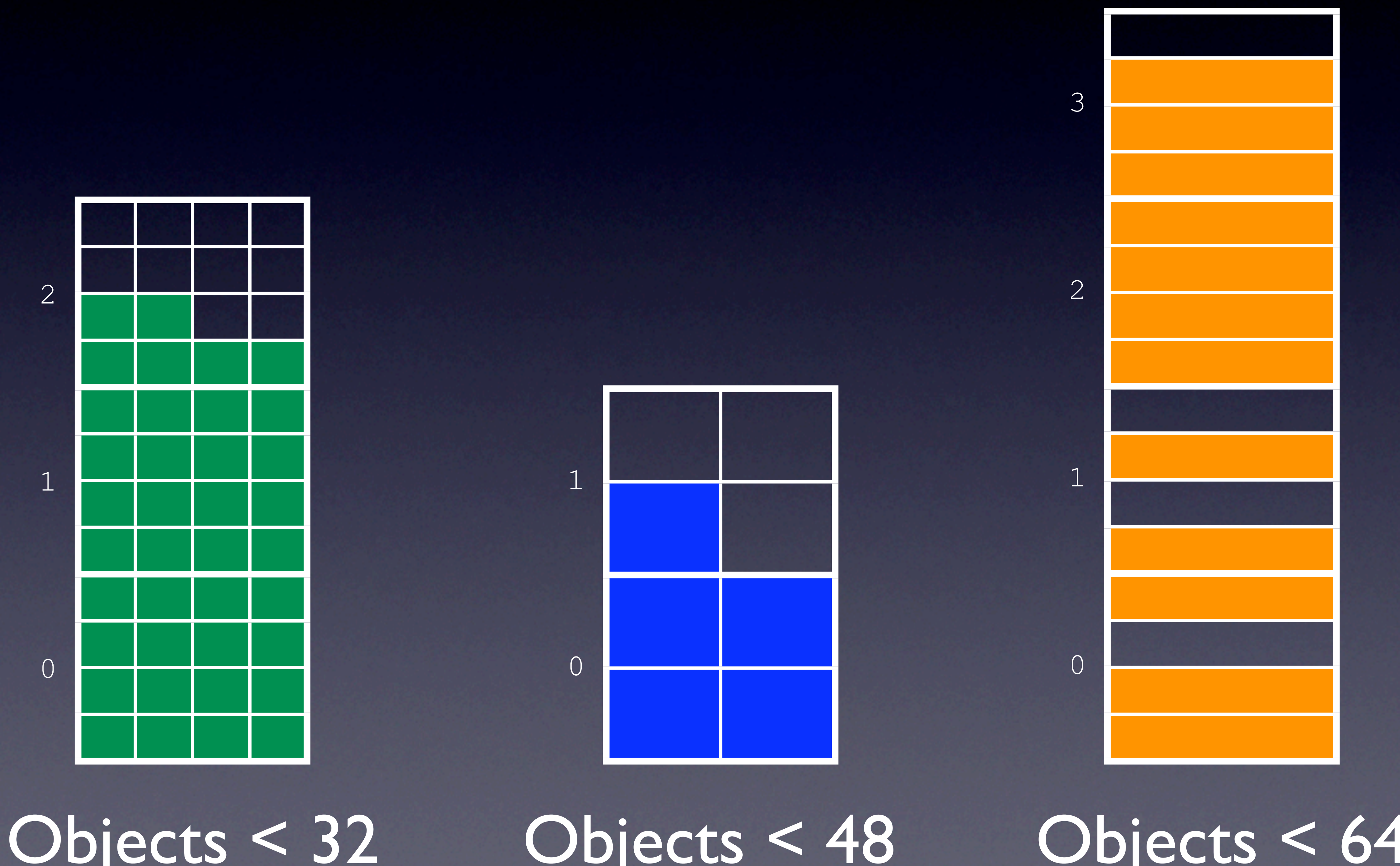


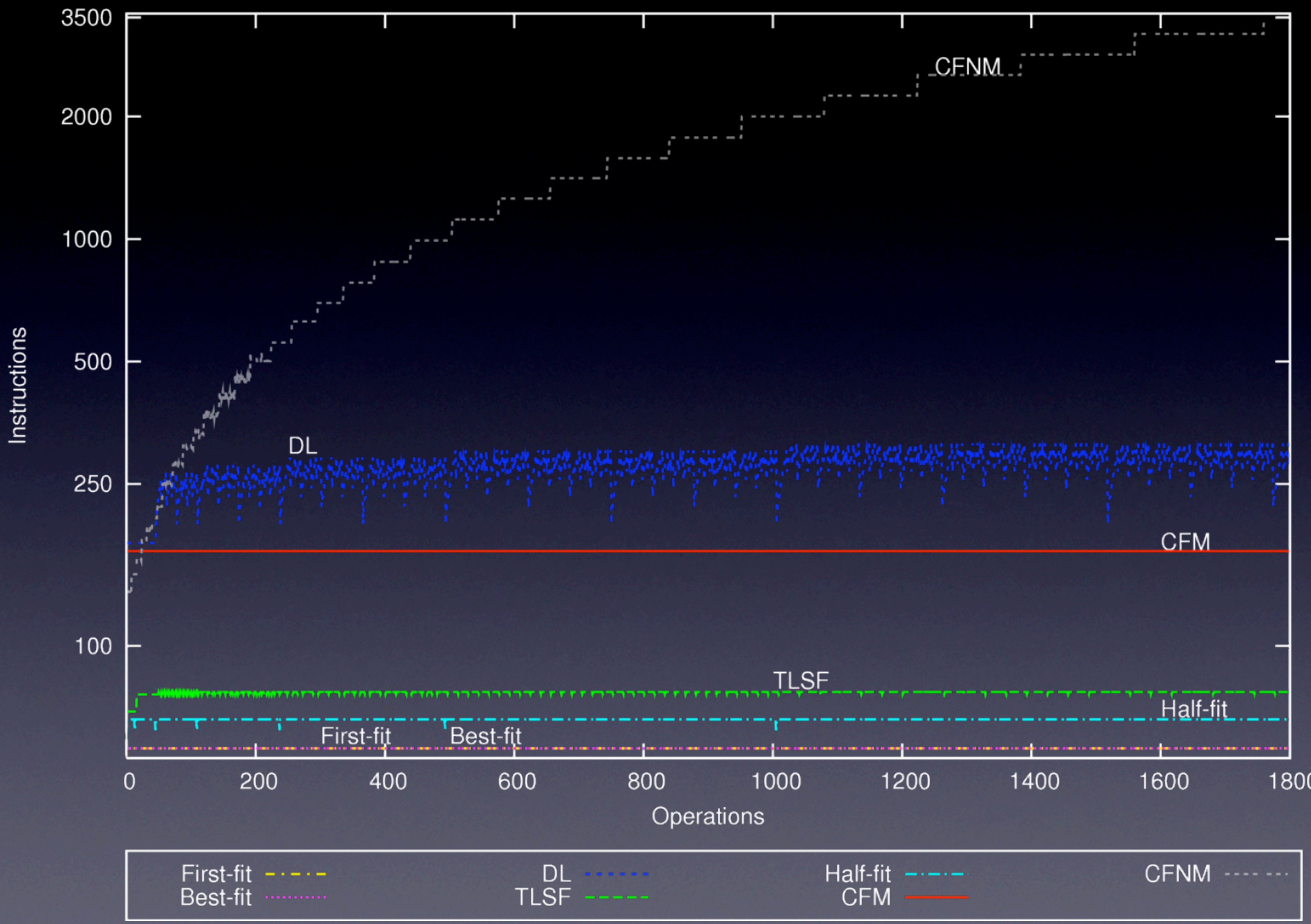
CFNM
CFM
DL
TLSF
Half-fit
First-fit
Best-fit
5000
2500
1000
500
250
100
Instructions
0
200
400
600
800
1000
1200
1400
1600
1800
Operations
First-fit
Best-fit
DL
TLSF
Half-fit
CFM
CFNM

Results I

- malloc(n) takes O(1)
- free(n) takes O(n)
- access takes one indirection
- memory fragmentation is bounded and predictable in constant time

Partial Compaction





Instructions
3500
2000
1000
500
250
100
0
200
400
600
800
1000
1200
1400
1600
1800
Operations
CFNM
DL
CFM
TLSF
Half-fit
First-fit
Best-fit
First-fit
Best-fit
DL
TLSF
Half-fit
CFM
CFNM

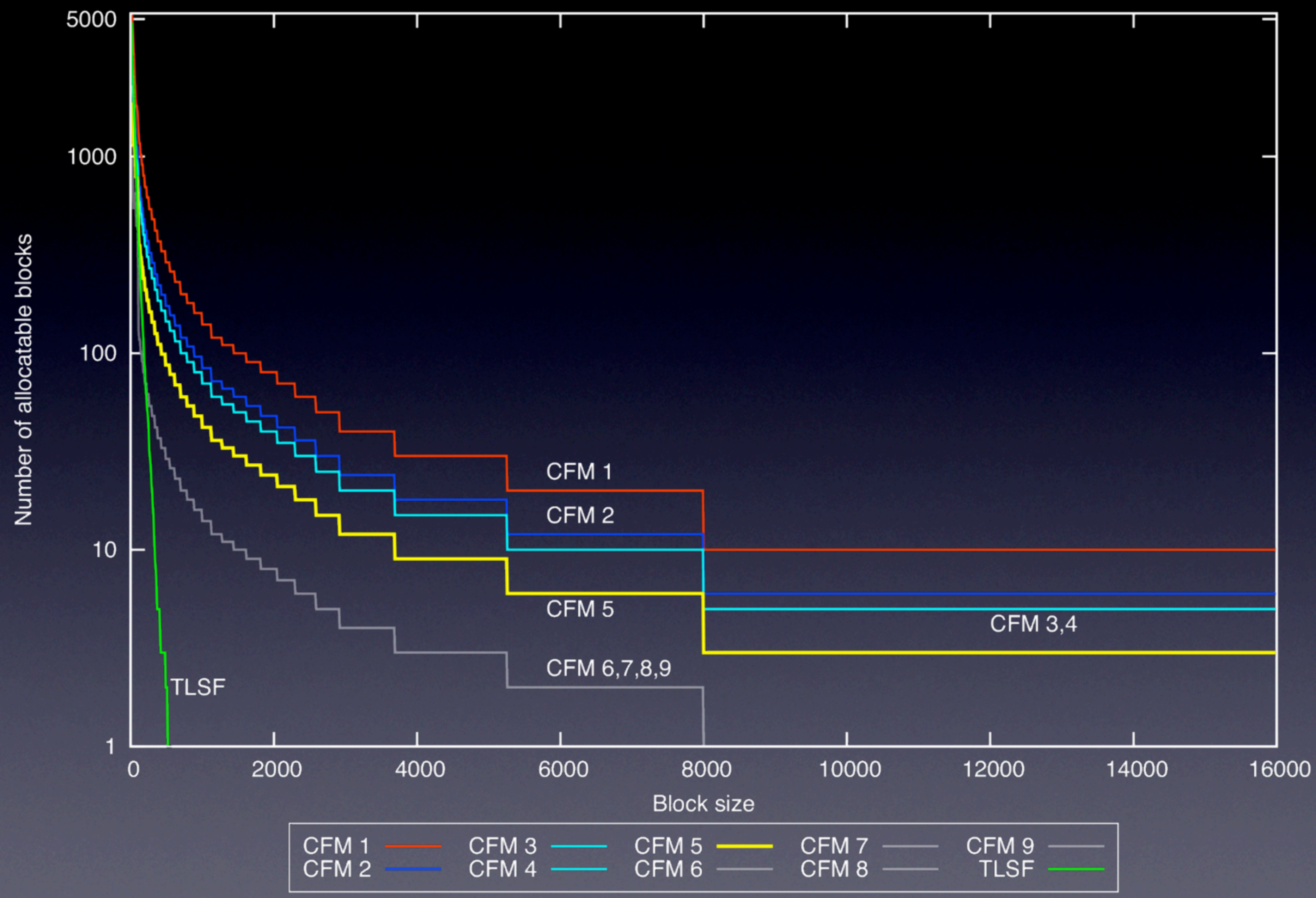
Program Analysis

Definition:
Let k count deallocations in a given size-class for which no subsequent allocation was done ("k-band mutator").

Proposition:
Each deallocation that happens when k < max_number_of_non_full_pages takes constant time.

Results II

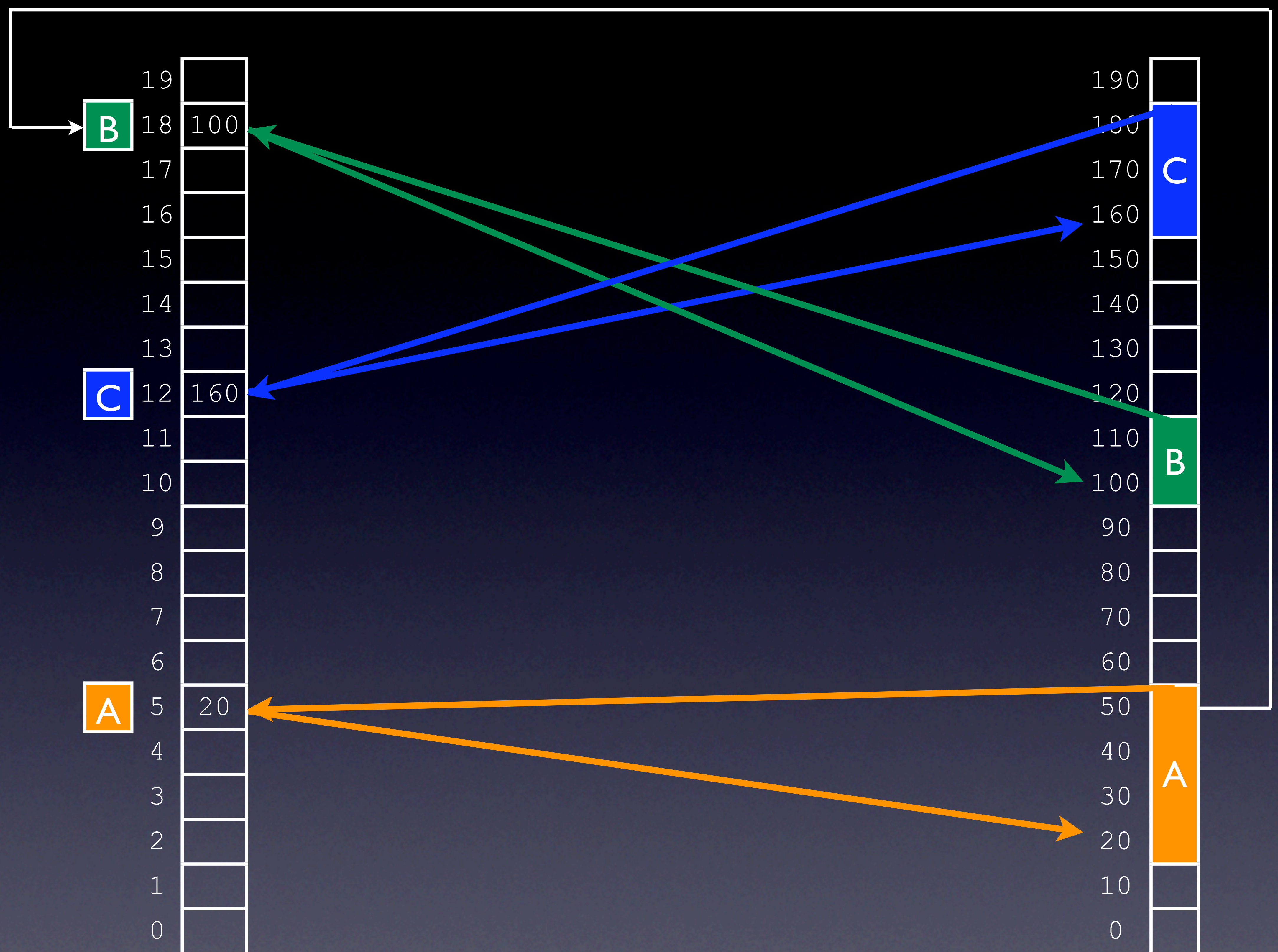
- if mutator stays within k-bands:
  - malloc(n) takes O(1)
  - free(n) takes O(1)
  - access takes one indirection
- memory fragmentation is bounded in k and predictable in constant time



Number of allocatable blocks
5000
1000
100
10
1
0
2000
4000
6000
8000
10000
12000
14000
16000
Block size
CFM 1
CFM 2
CFM 5
CFM 3,4
CFM 6,7,8,9
TLSF
CFM 1
CFM 2
CFM 3
CFM 4
CFM 5
CFM 6
CFM 7
CFM 8
CFM 9
TLSF

Two Implementations!

1. Concrete Space = Physical Memory
2. Concrete Space = Virtual Memory

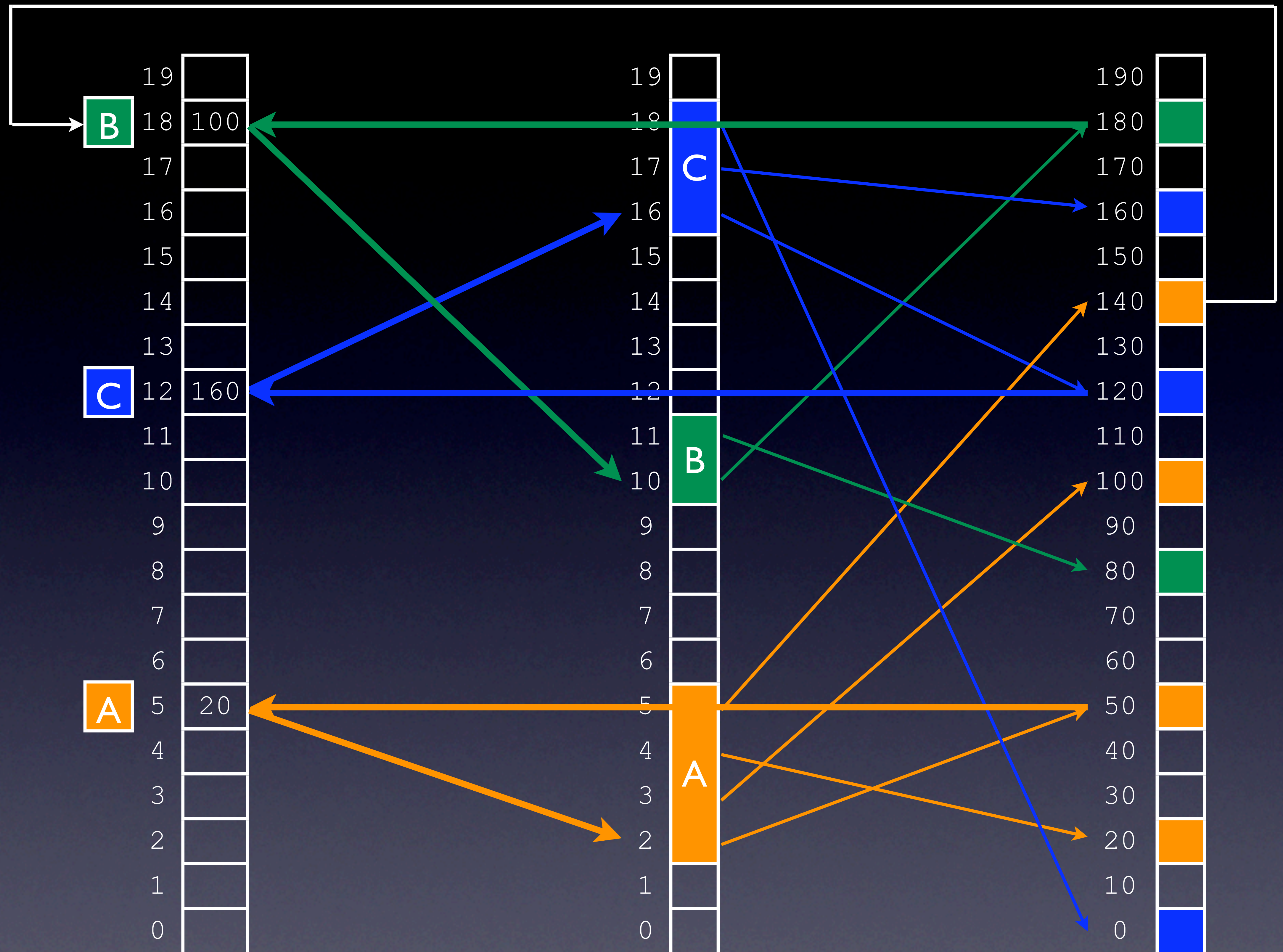


Abstract Space

Physical Memory

Two Implementations!

1. Concrete Space = Physical Memory
2. Concrete Space = Virtual Memory

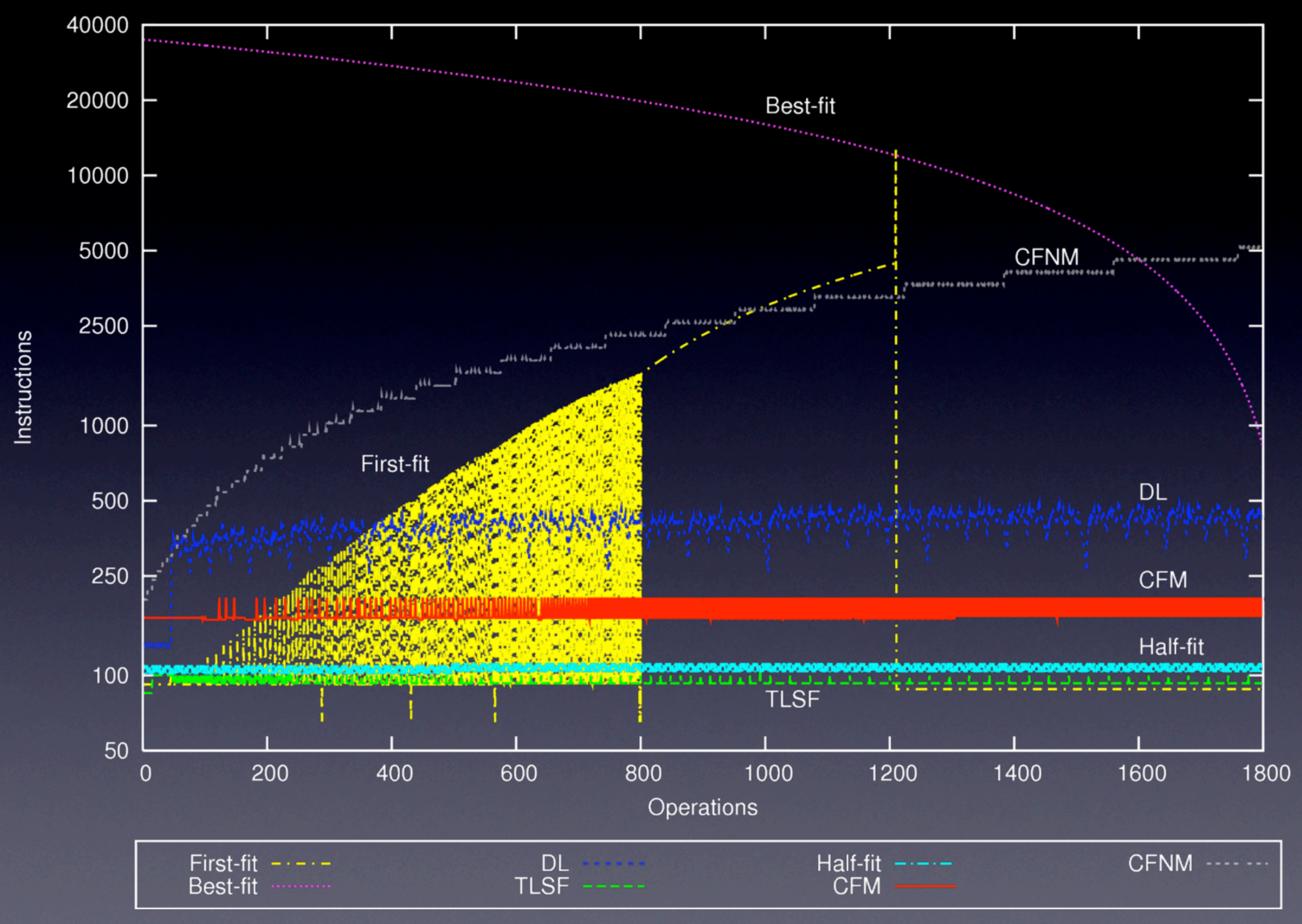


Abstract Space Virtual Space Physical Memory

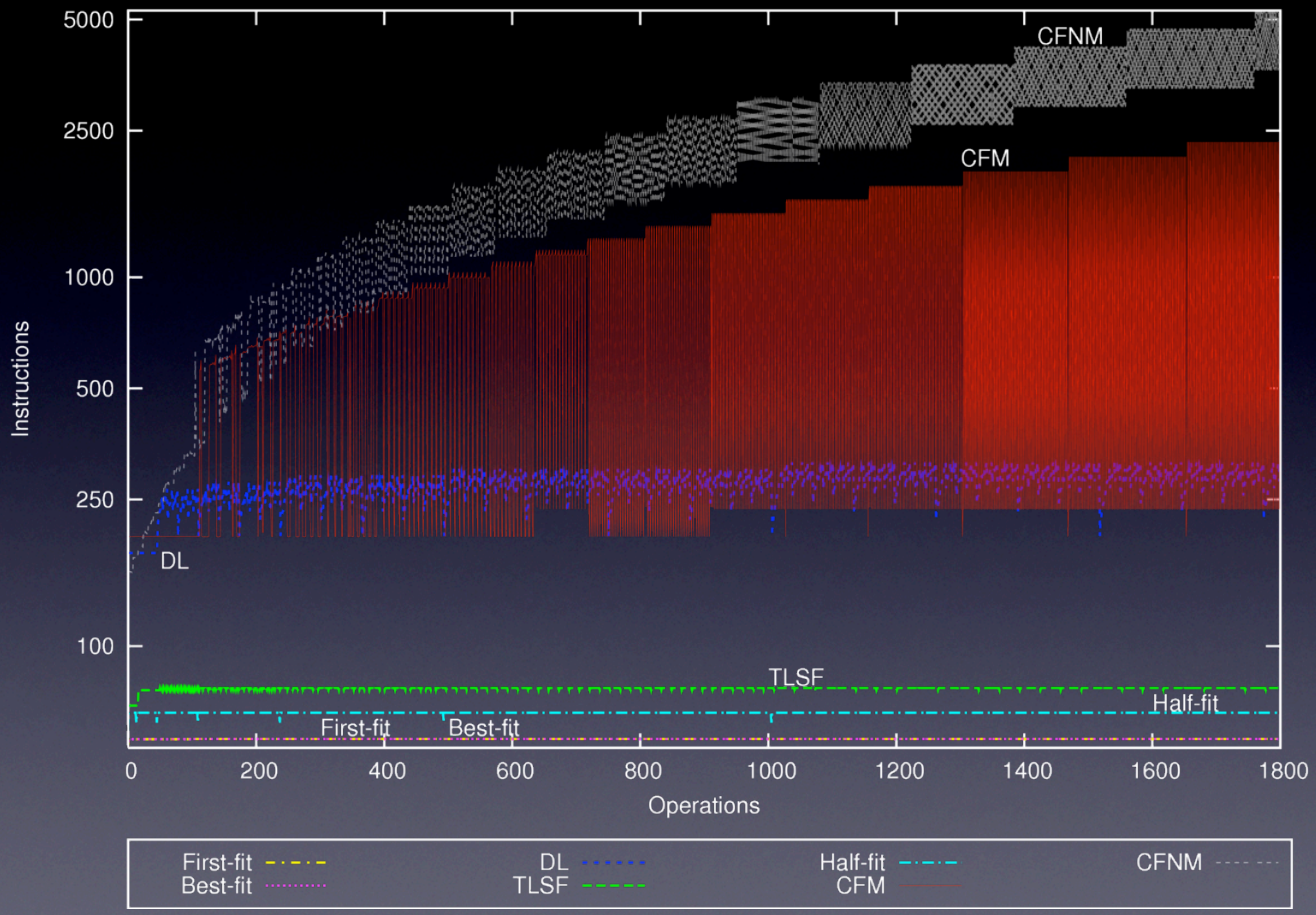
Results III

- malloc(n) takes O(n)
- free(n) takes O(n)
- access takes two indirections

- memory fragmentation is bounded in k and predictable in constant time



40000
20000
10000
5000
2500
1000
500
250
100
50
Instructions
0
200
400
600
800
1000
1200
1400
1600
1800
Operations
Best-fit
CFNM
First-fit
DL
CFM
Half-fit
TLSF
First-fit
Best-fit
DL
TLSF
Half-fit
CFM
CFNM



5000
2500
1000
500
250
100
Instructions
0
200
400
600
800
1000
1200
1400
1600
1800
Operations
CFNM
CFM
DL
TLSF
Half-fit
First-fit
Best-fit
First-fit
Best-fit
DL
TLSF
Half-fit
CFM
CFNM

Current/Future Work

- Concurrent memory management
- I/O subsystem
- Constant-time scheduler
- Java bytecode VM

Thank you