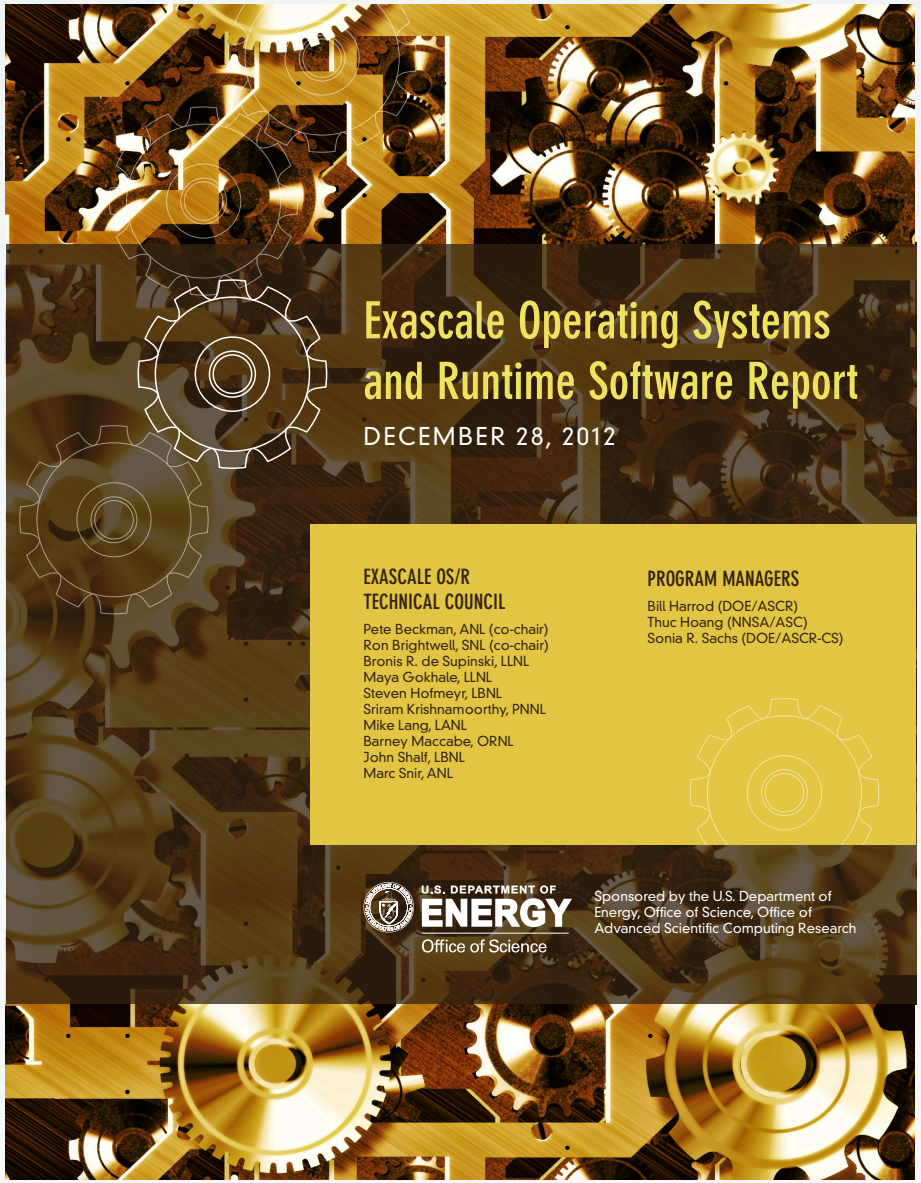
TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Systems Architecture, Operating Systems Group

BUILDING BLOCKS FOR AN EXA-SCALE OPERATING SYSTEM

HERMANN HÄRTIG

ROSS 2014

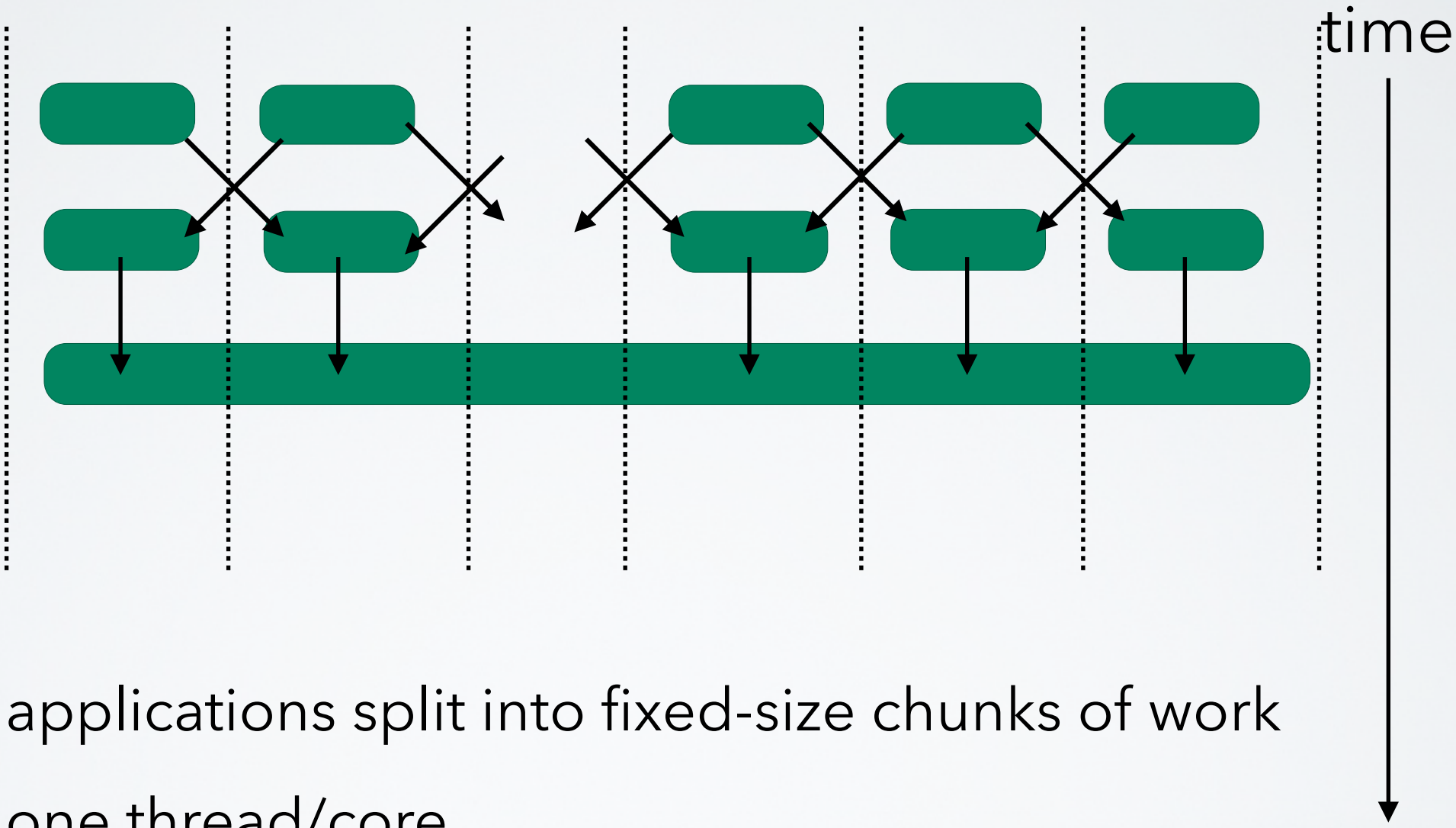

Exascale Operating Systems and Runtime Software Report
DECEMBER 28, 2012
EXASCALE OS/R TECHNICAL COUNCIL
Pete Beckman, ANL (co-chair)
Ron Brightwell, SNL (co-chair)
Bronis R. de Supinski, LLNL
Maya Gokhale, LLNL
Steven Hofmeyr, LBNL
Sriram Krishnamoorthy, PNNL
Mike Lang, LANL
Barney Maccabe, ORNL
John Shalf, LBNL
Marc Snir, ANL
PROGRAM MANAGERS
Bill Harrod (DOE/ASCR)
Thuc Hoang (NNSA/ASC)
Sonia R. Sachs (DOE/ASCR-CS)
U.S. DEPARTMENT OF ENERGY
Office of Science
Sponsored by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research

TRADITIONAL HPC

the ideal world assumption

- identical predictable reliable nodes
- fast deterministic reliable interconnect with isolated partitions of fixed size
- balanced applications

TRADITIONAL HPC



applications split into fixed-size chunks of work

one thread/core

TRADITIONAL HPC

systems software:
optimize communication latency

- RDMA & busy waiting
- batch scheduler for start / stop
- separate servers for IO
  small OS on each node

no OS on critical path

REALITY CHECK

observations (infiniband cluster):

- CPUs run at different speeds:
  3 of 16 @ 1.2GH, others 13 @ 2.9GH
- “turbo boost” switched off
- measurements not reproducible
  (node allocations arbitrary)
- application restart due to failures

SC REALITY CHECK

Application: COSMO-SPECS+FD4

Unbalanced compute times of ranks per time step

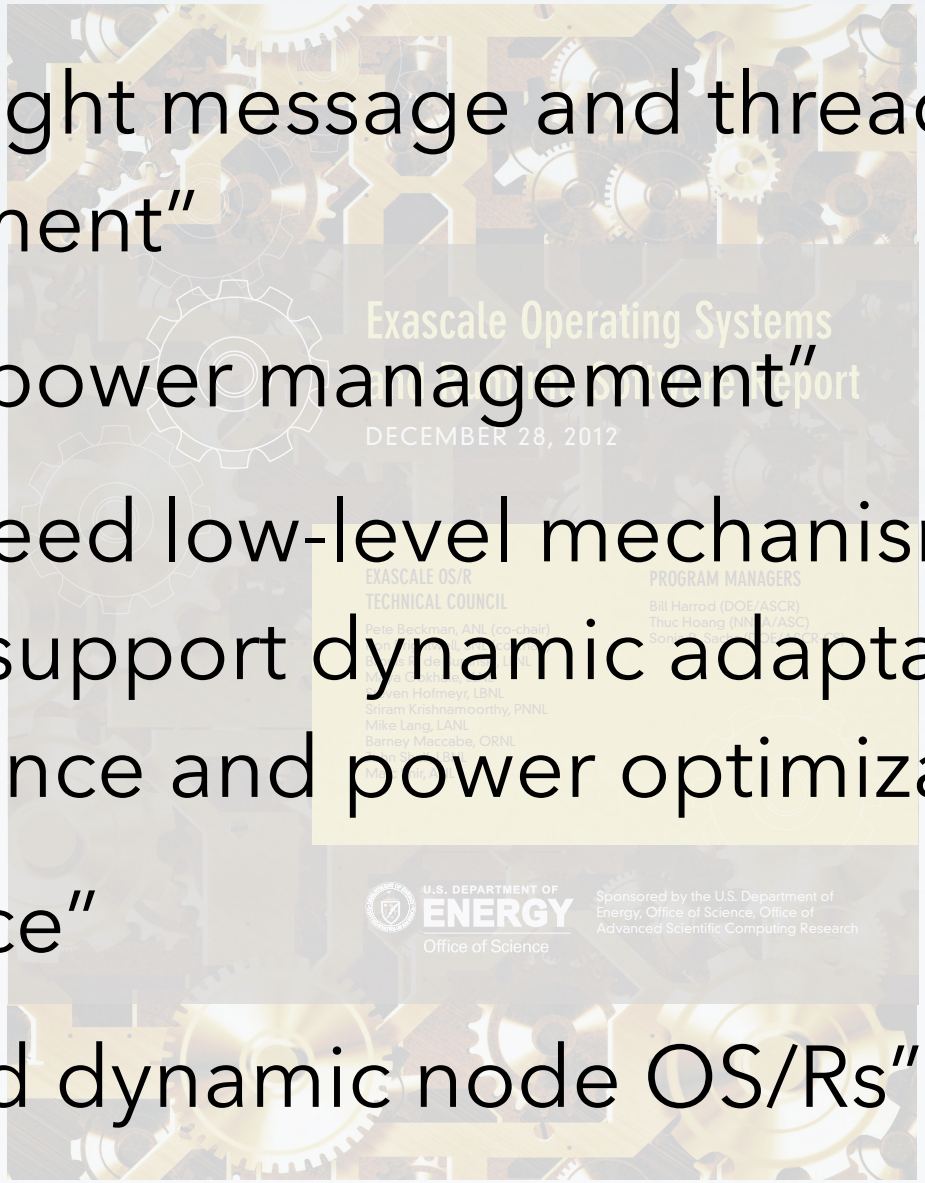
Hand balanced compute times of ranks per time step

EXASCALE HPC

- very large number of processing units
- diverse on-node storage
- heterogeneous processing units
- different and dynamic speed
- higher failure rate
- dark silicon
- new resource trade-offs
- dynamic application behavior

HPC-OS CONCERNS

- "Lightweight message and thread management"
- "Holistic power management"
- "OS/Rs need low-level mechanisms in order to support dynamic adaptation, performance and power optimization"
- "Resilience"
- "agile and dynamic node OS/Rs"



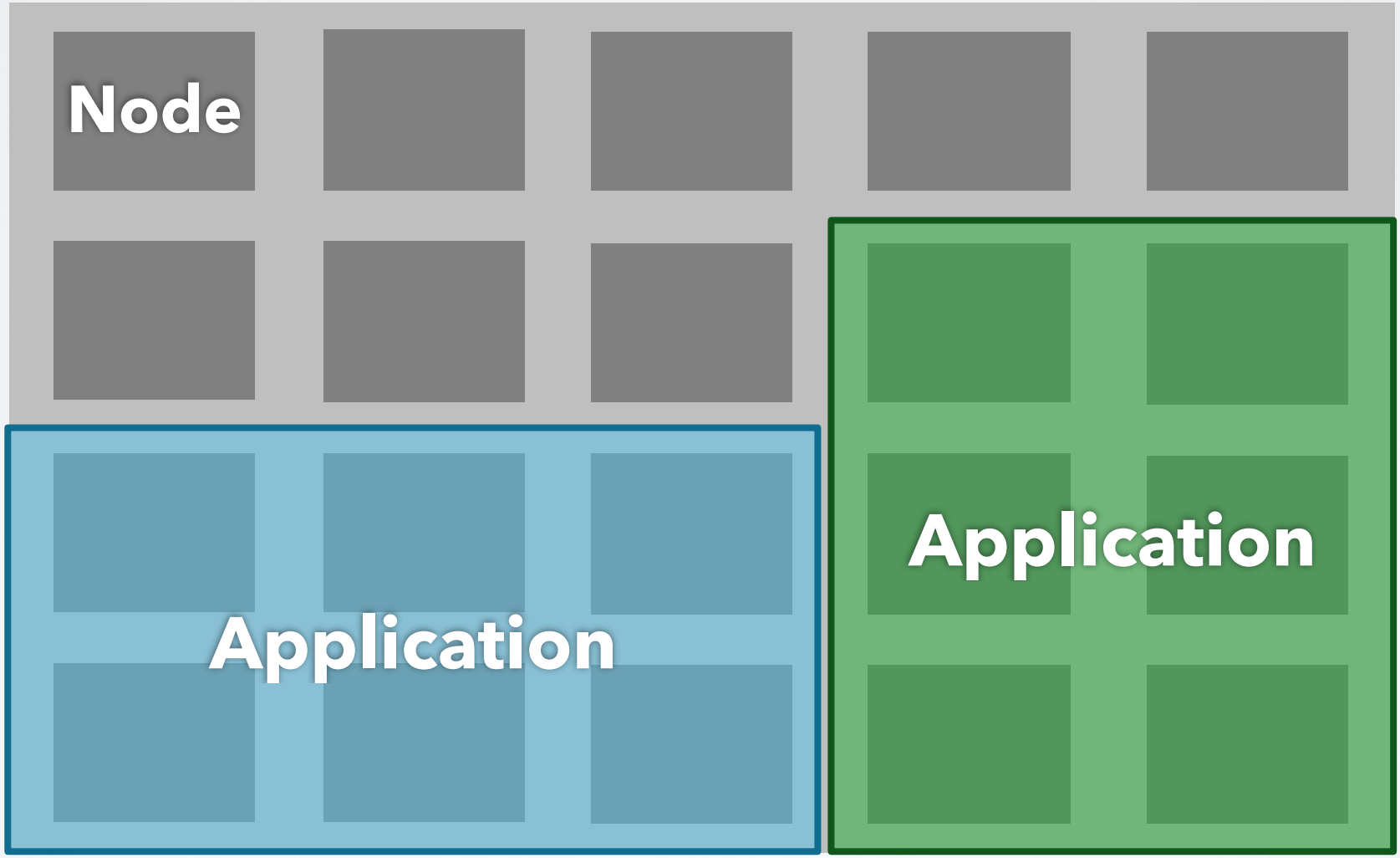
ZIB

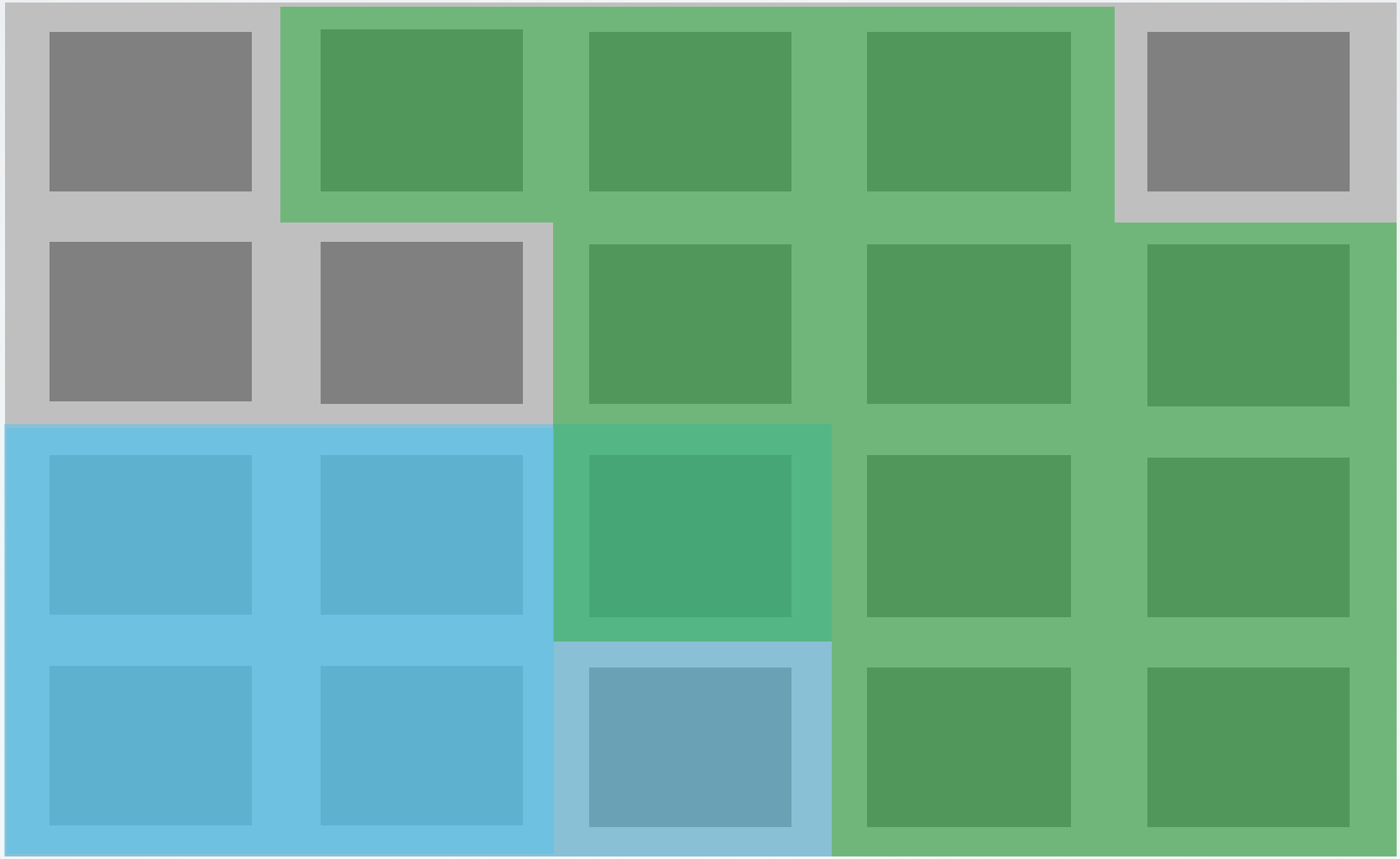
The Hebrew University of Jerusalem

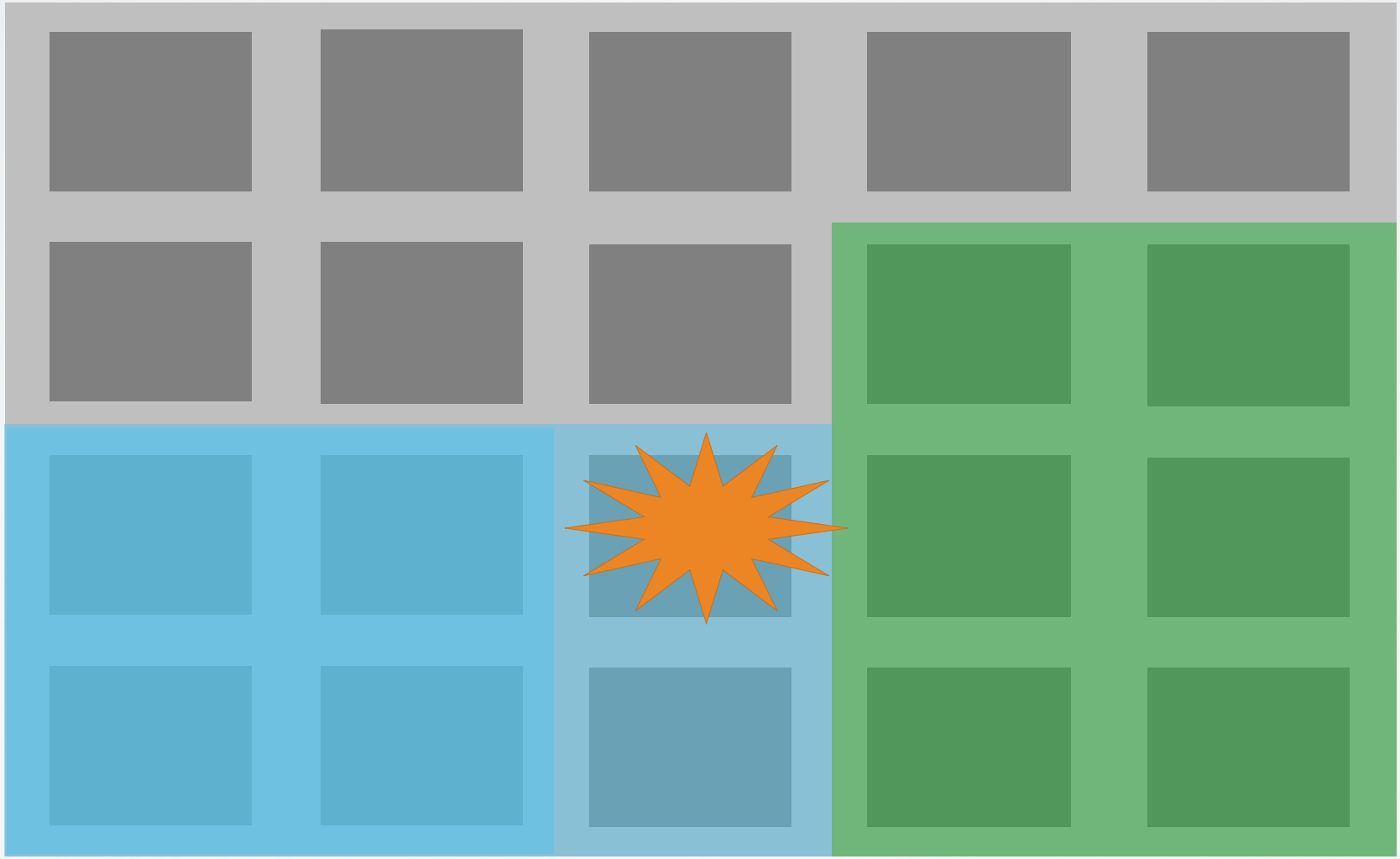
Fast and Fault-Tolerant Microkernel-Based Operating System

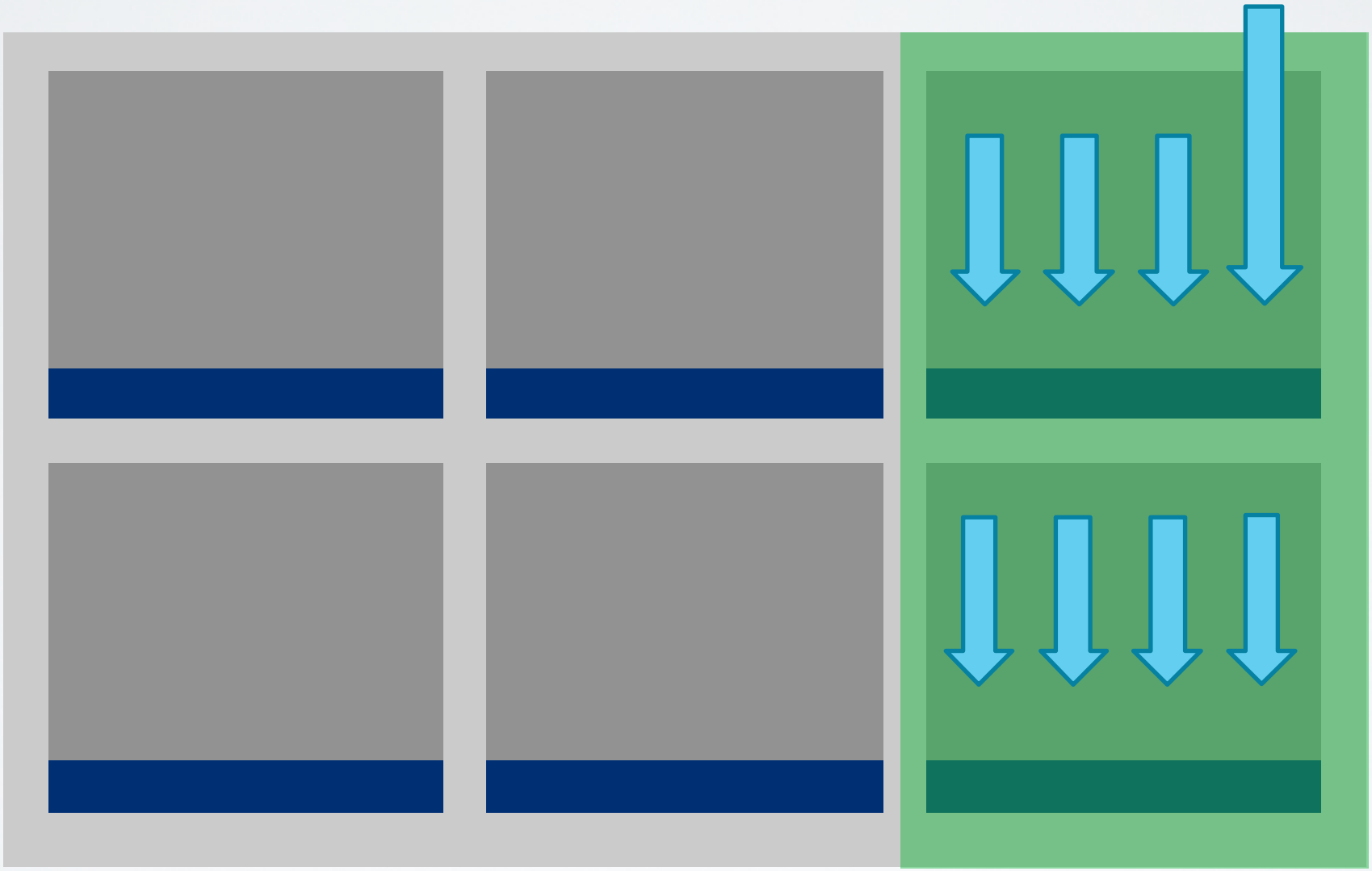
DFG Deutsche Forschungsgemeinschaft

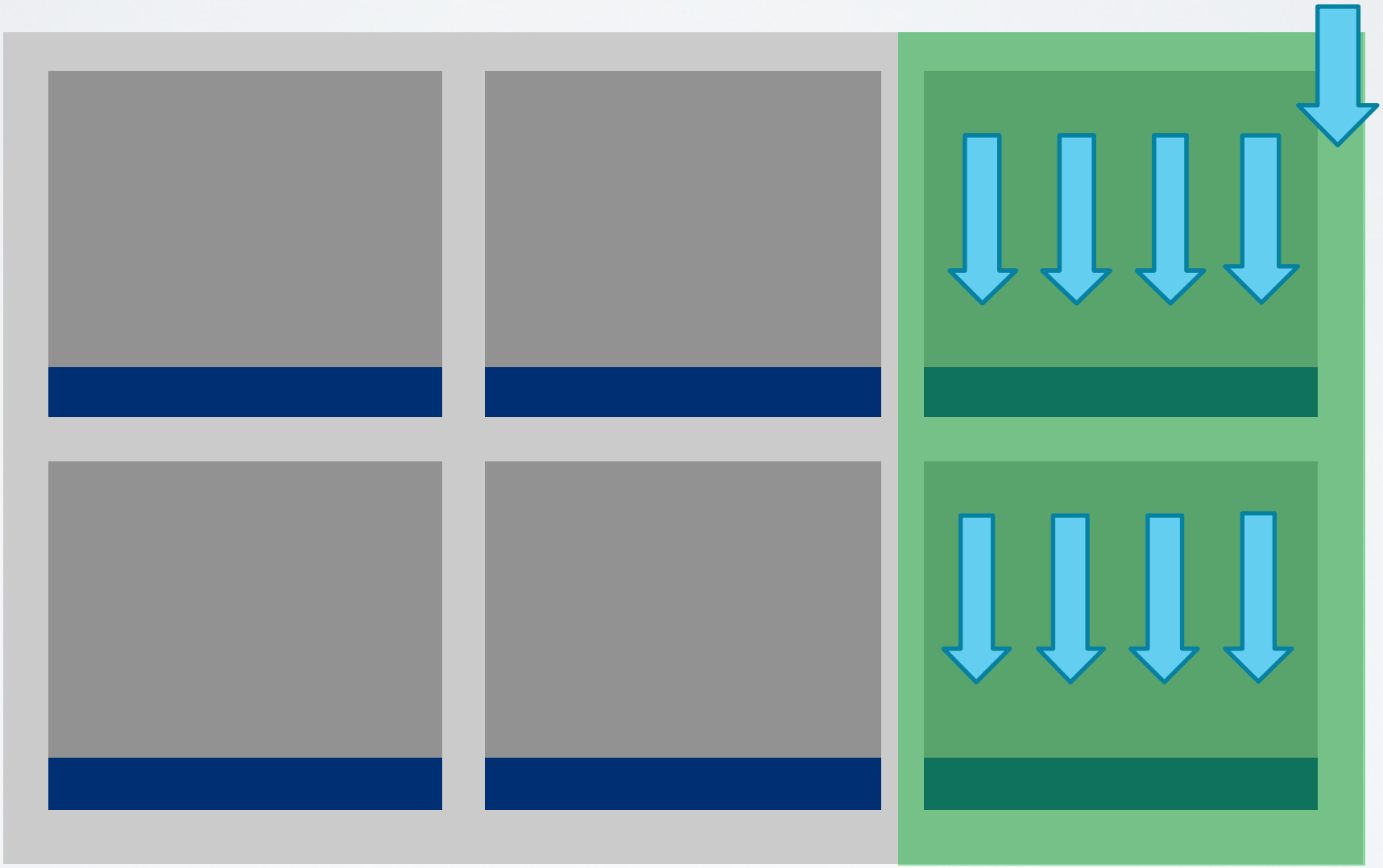
SPPEXA German Priority Programme 1648
Software for Exascale Computing


Node
Application
Application





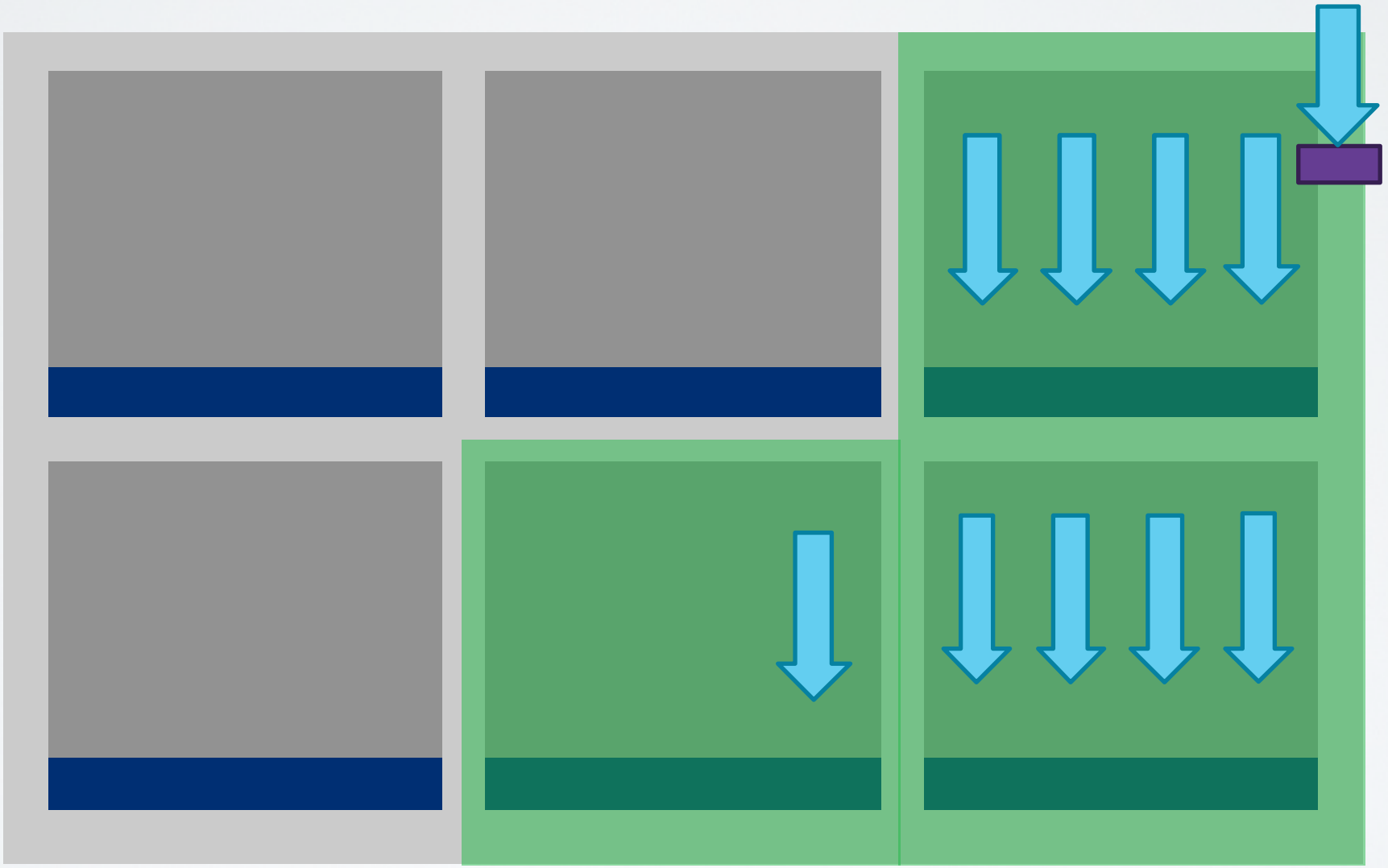




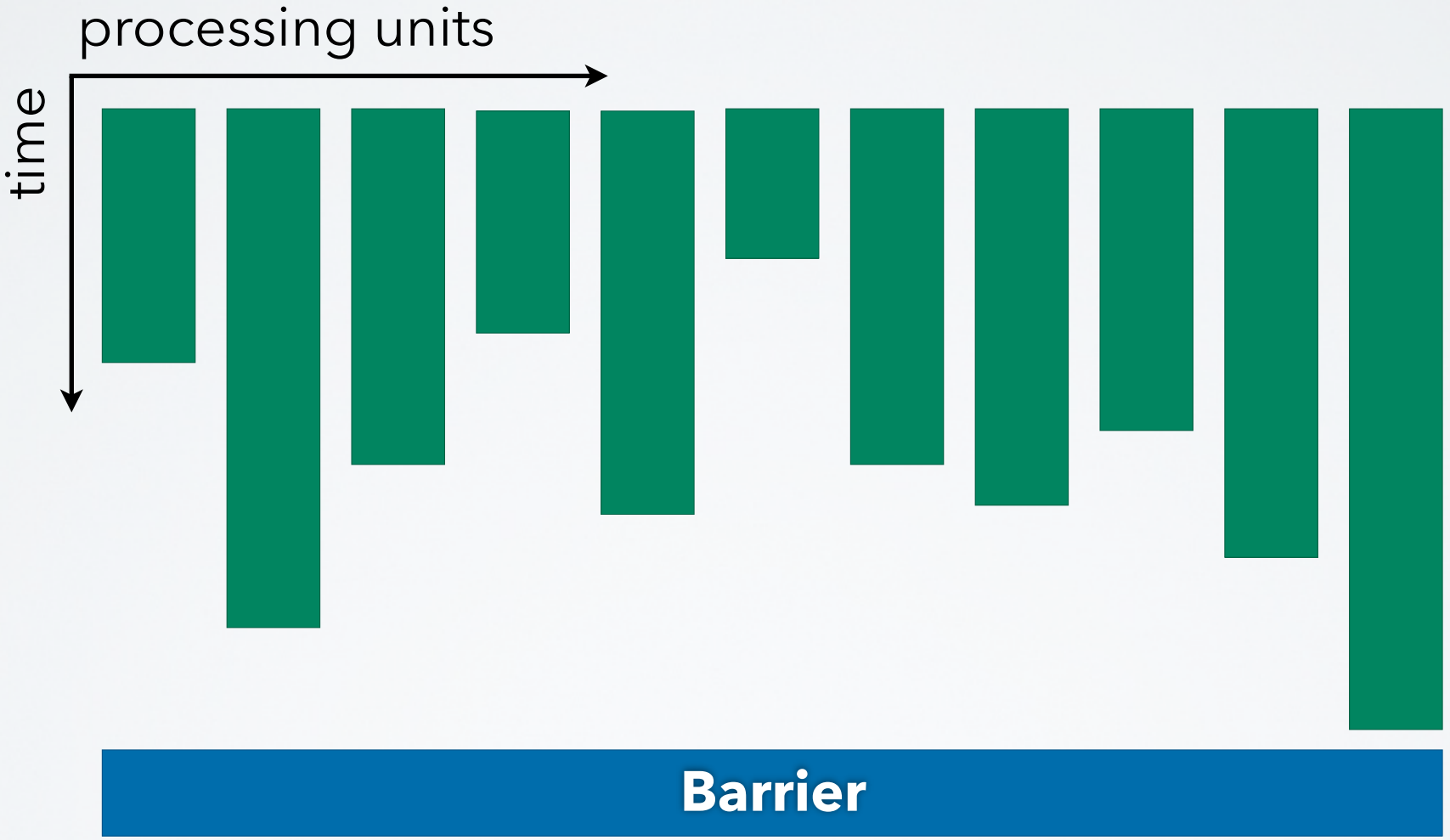
yell

for

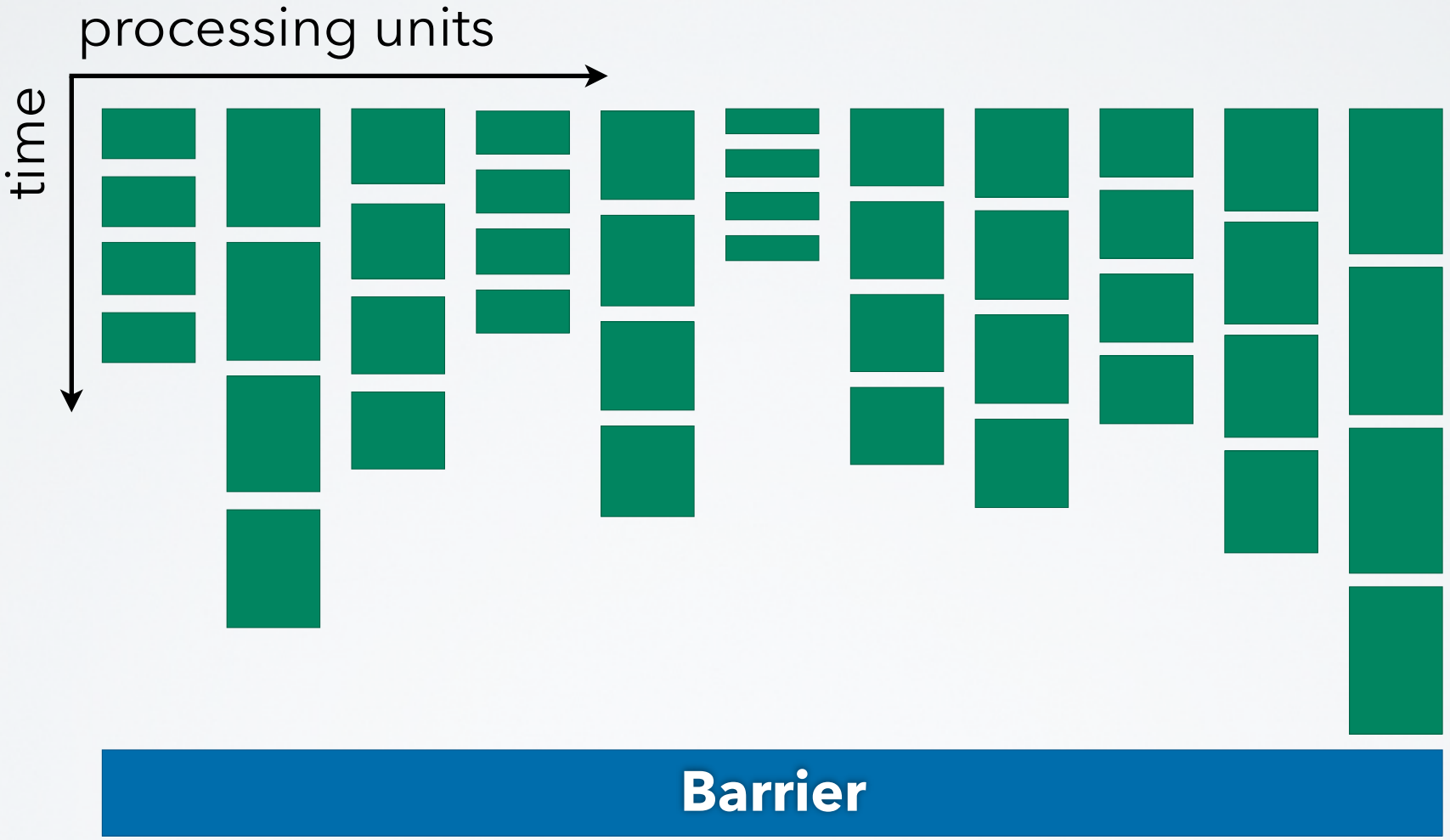
help



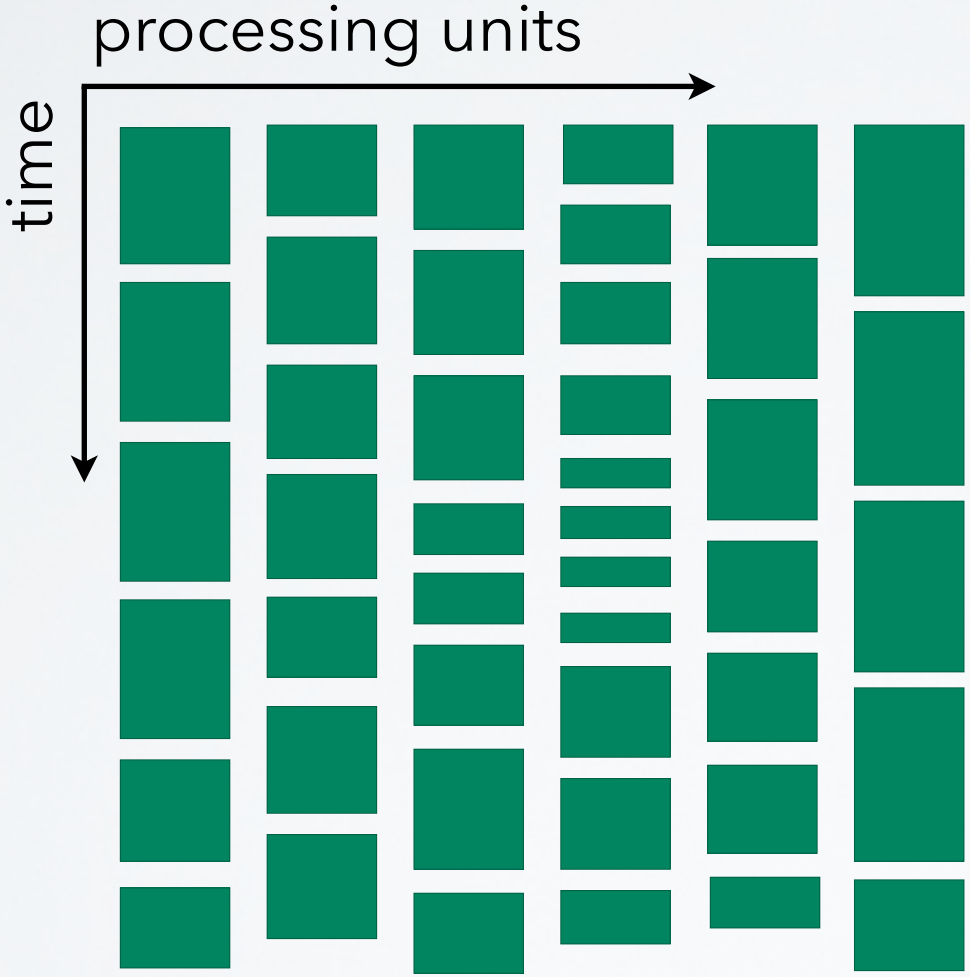
OVERDECOMP



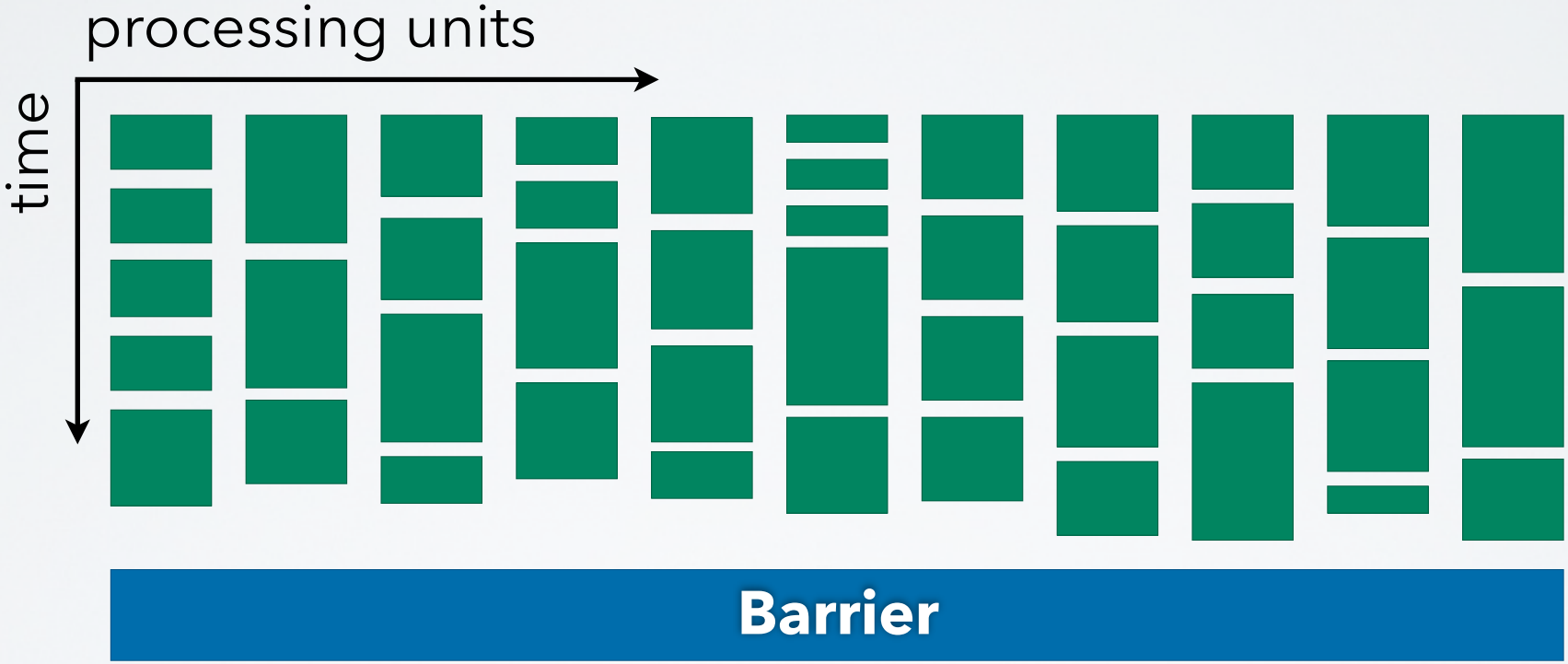
OVERDECOMP



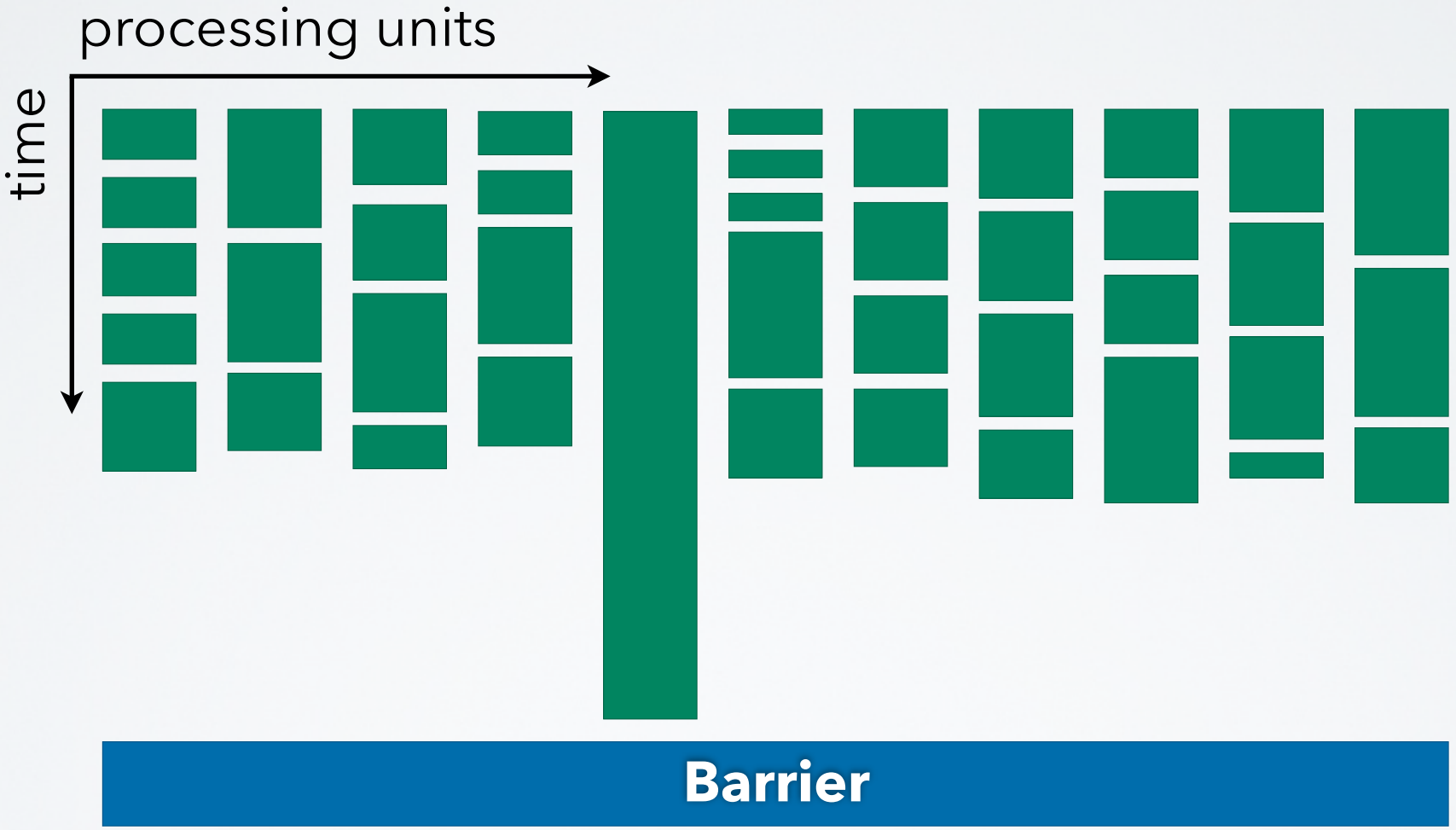
OVERDECOMP



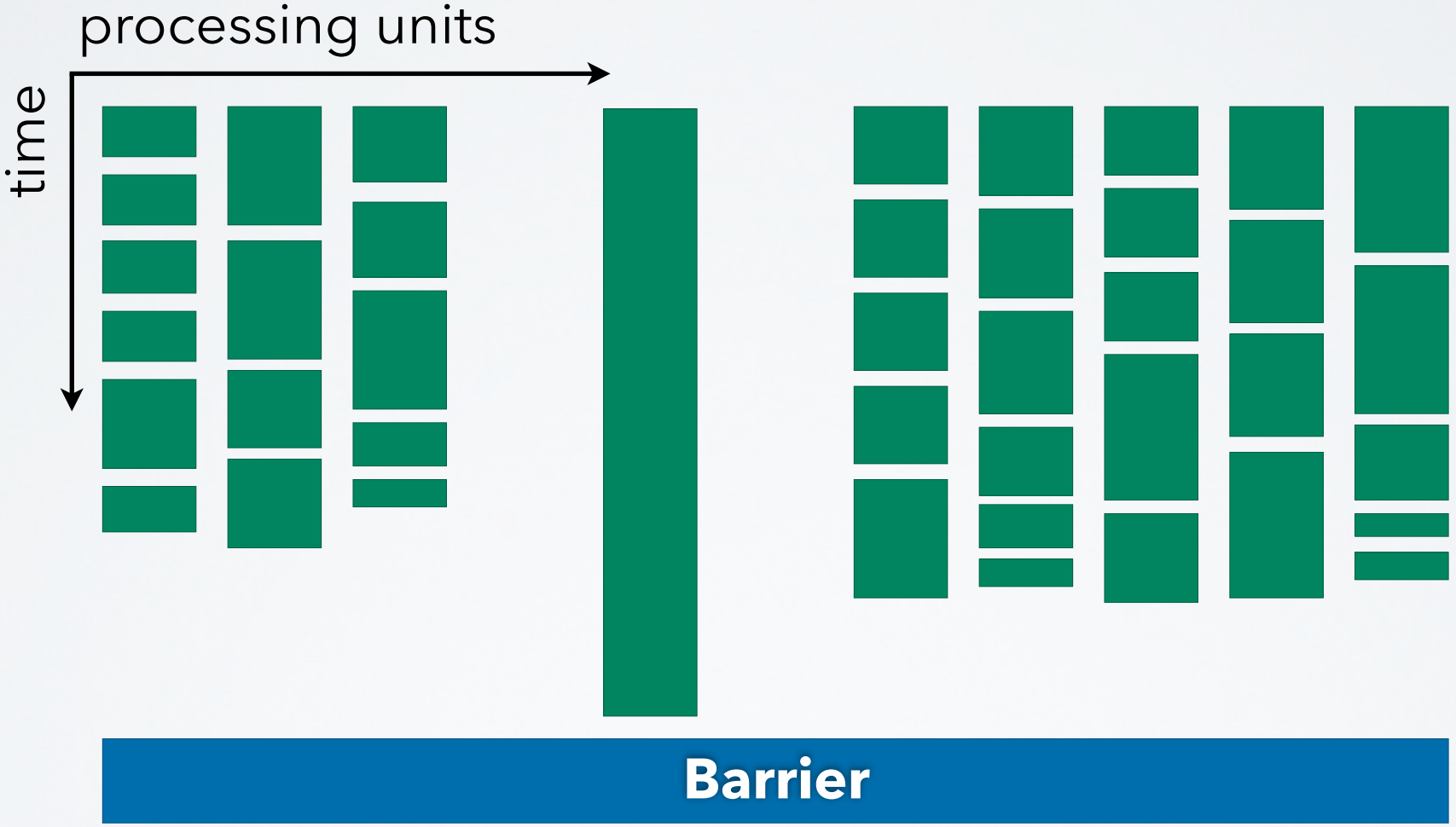
OVERDECOMP



OVERDECOMP



OVERDECOMP



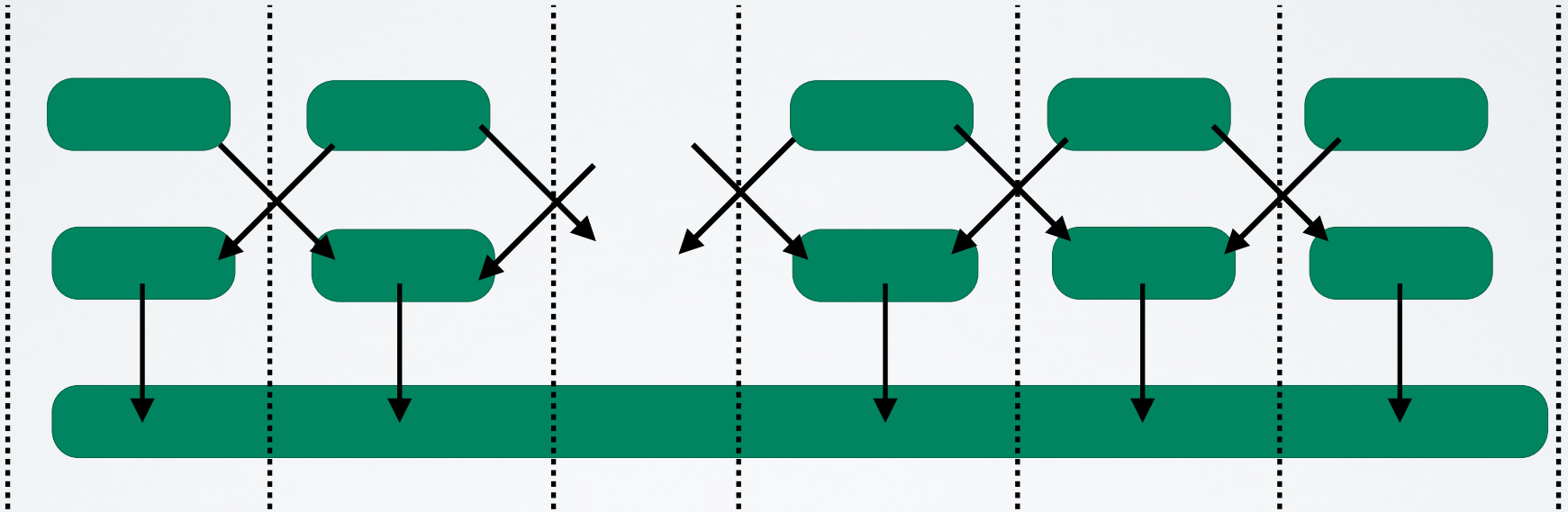
OVERDECOMP

simplistic experiment

(Linux on IB cluster)

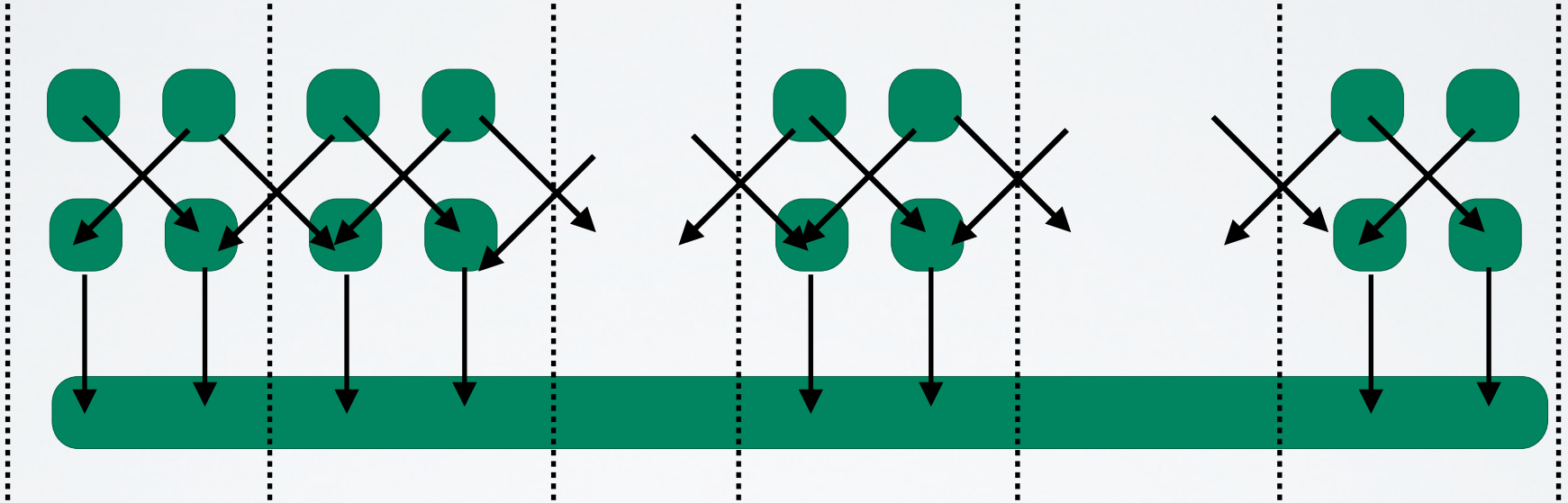
- split application into more chunks than cores
- run more MPI processes than cores

OVERDECOMP



1 MPI process per core

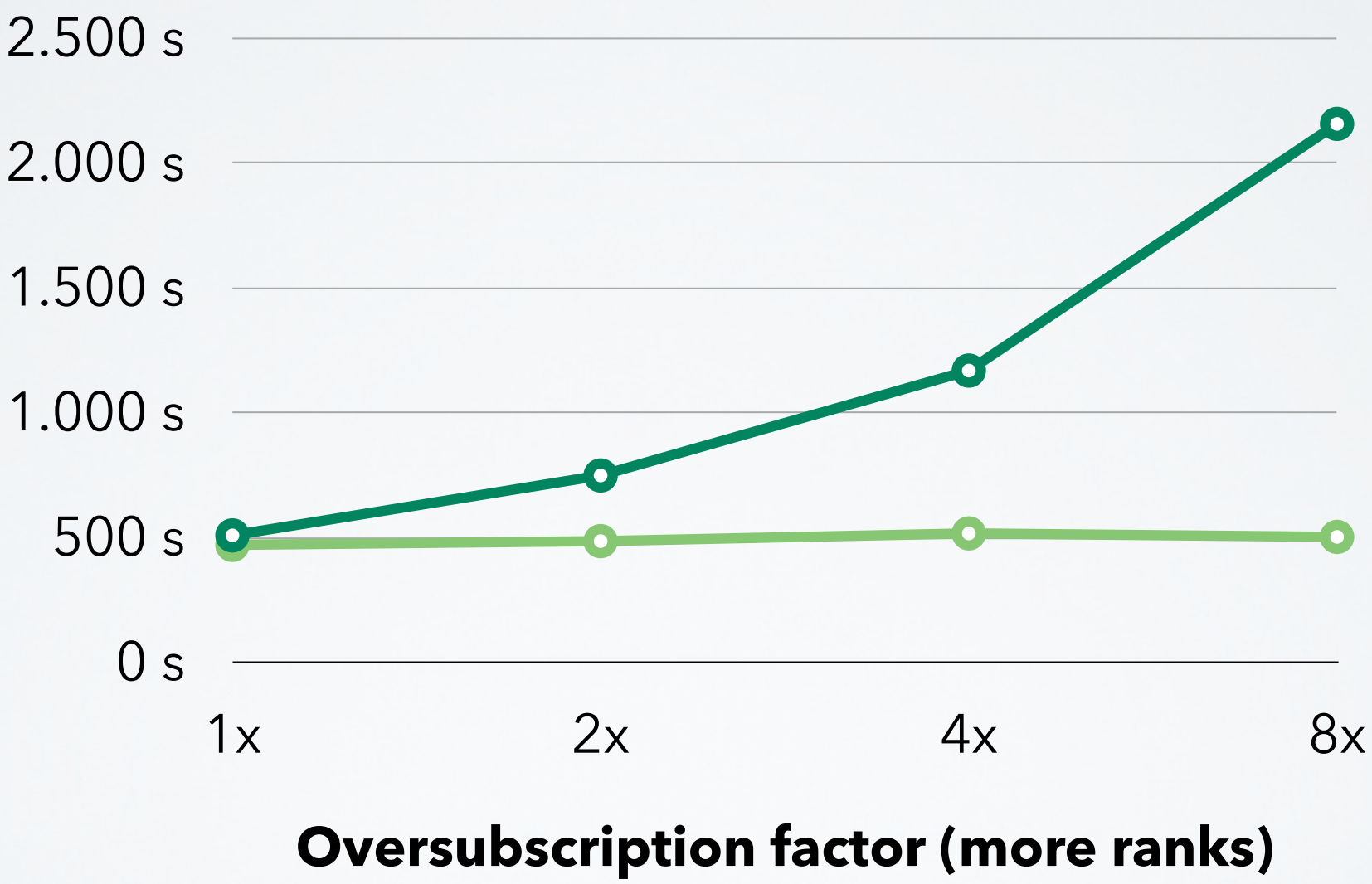
OVERDECOMP



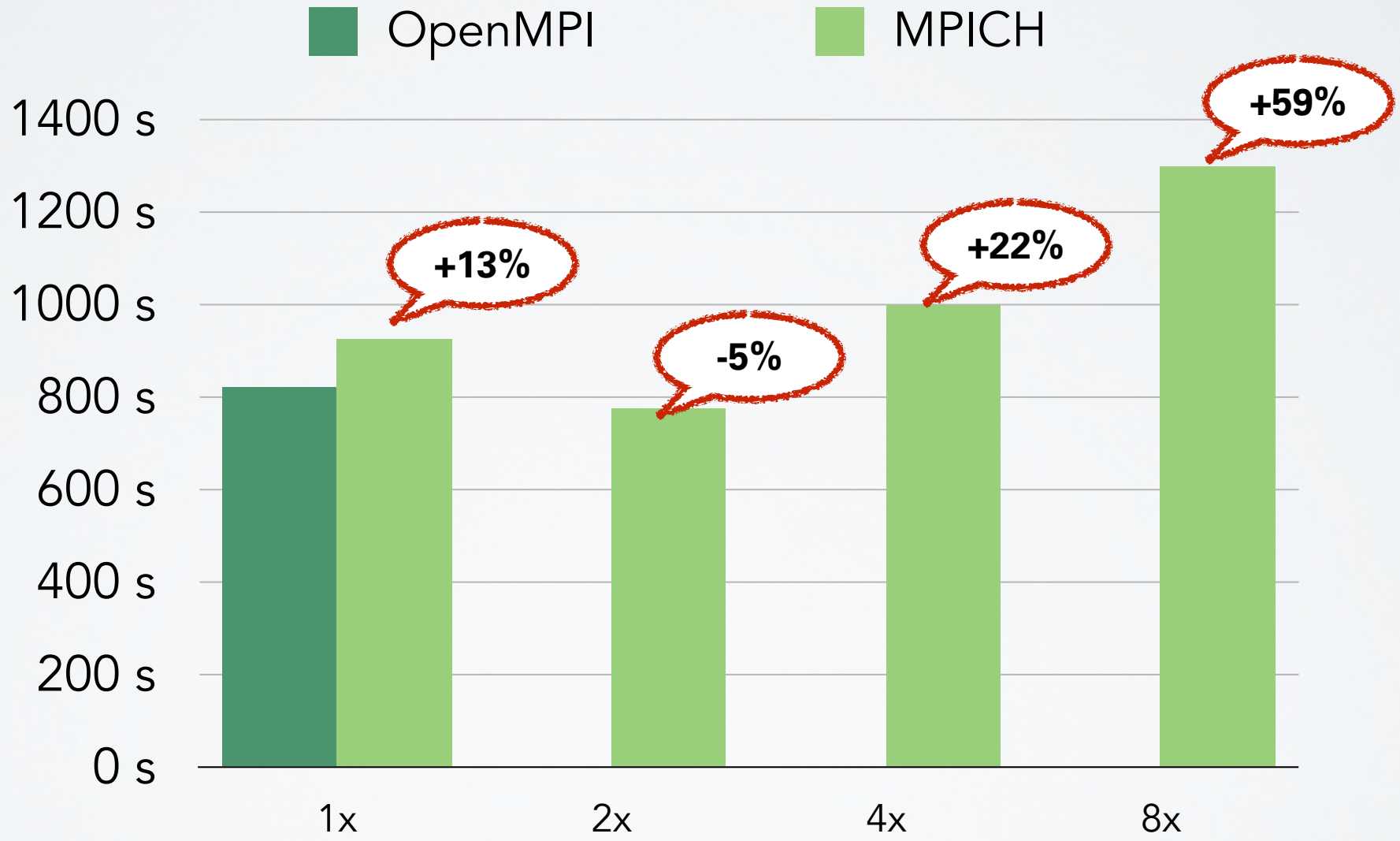
Application: COSMO-SPECS+FD4 (no load balancing)

- IB-Cluster 16 nodes w/ 16 Xeon E5-2690 (Sandy Bridge) @ 2.90GHz
- 1x - 8x oversubscription (same problem size)

OVERDECOMP

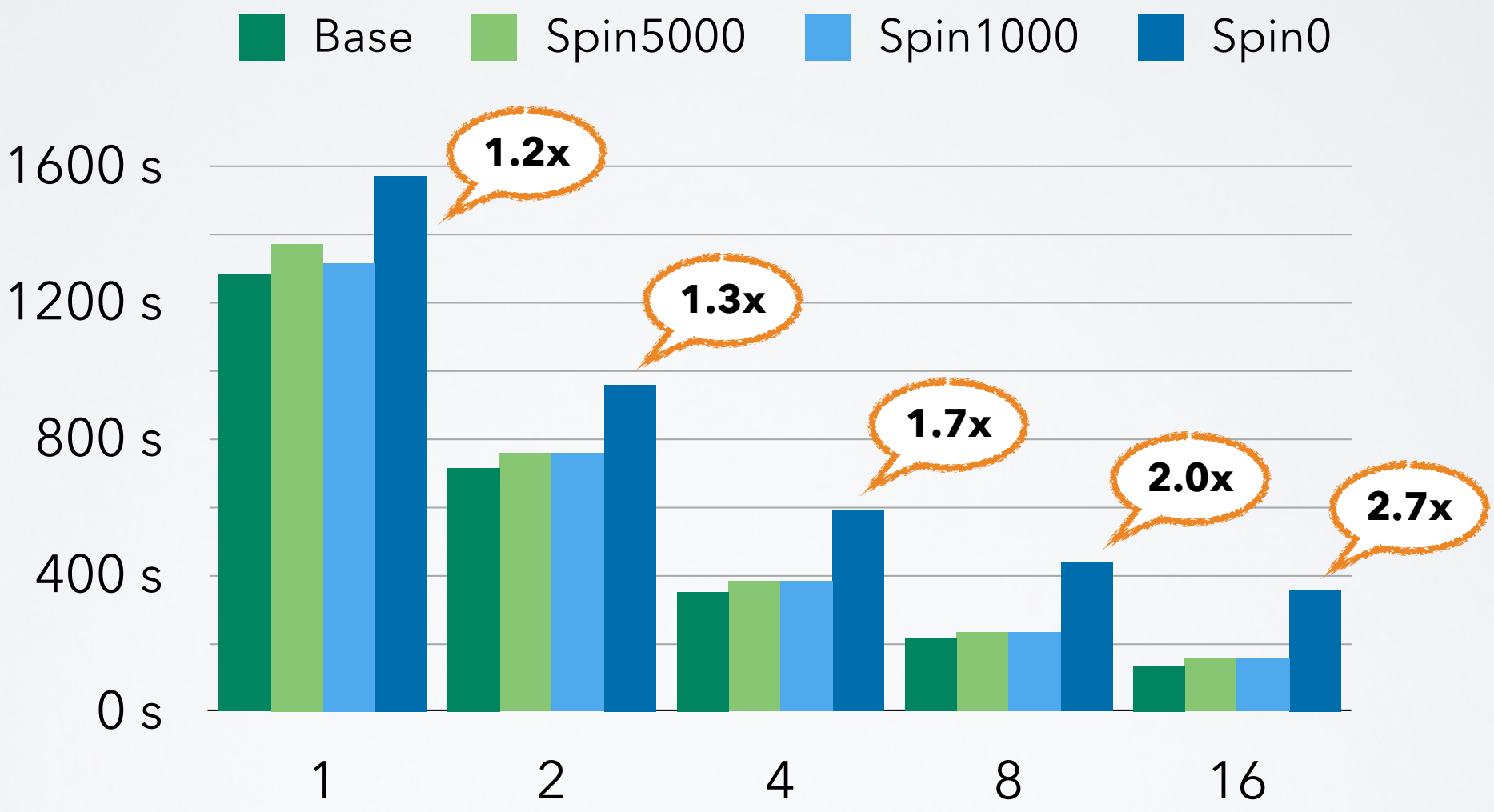

2.500 s
2.000 s
1.500 s
1.000 s
500 s
0 s
1x
2x
4x
8x
Oversubscription factor (more ranks)

OVERDECOMP


OpenMPI
MPICH
1400 s
1200 s
1000 s
800 s
600 s
400 s
200 s
0 s
+13%
-5%
+22%
+59%
1x
2x
4x
8x

TECHNISCHE UNIVERSITÄT DRESDEN

COST OF BLOCKING



COST OF BLOCKING

Table 2 Receive 1 Gbps latency breakdown

Description of receive packet activities	Source	Time (ns)
MAC filter determines target packet is for this machine	Estimation	200
NIC starts DMA packet header and payload into memory	Estimation	400
NIC interrupts core with MSI-X packet to APIC	Estimation	500
Hardware MSI-X interrupt service routine to parse what caused interrupt	Estimation	270
Interrupt cause register read requirement	Measurement	1,000
ISR packet processing of descriptor to update receive queue	Measurement	300
SoftIRQ (deferred procedure call in Windows)	Measurement	1,287
TCP and IP receive side processing	Measurement	570
Wakeup application to process socket information	Measurement	1,274
Kernel to application space data copy	Measurement	208
ACK the pong received by the remote sender	Measurement	1,117
Application receive message overhead to register completion	Measurement	621
Total receive packet time		7,747

Architectural Breakdown of End-to-End Latency in a TCP/IP Network

Int J Parallel Prog, 2009 (intel)

OVERHEAD

overdecomposition overhead:

- blocking overhead
- additional messages

fast inter process communication

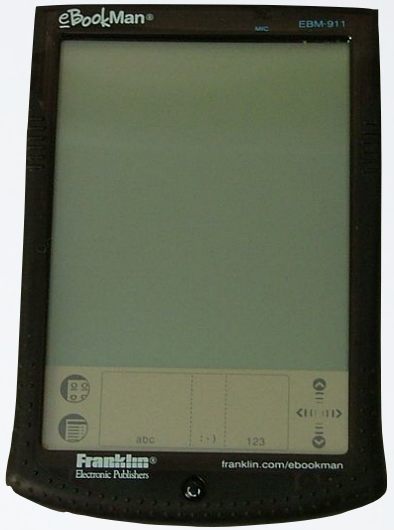
OVERHEAD

"Lightweight message and thread management"

L4 MICROKERNEL FAMILY

Jochen Liedtke
1953 -2001

Simko3 aka “Merkel phone”



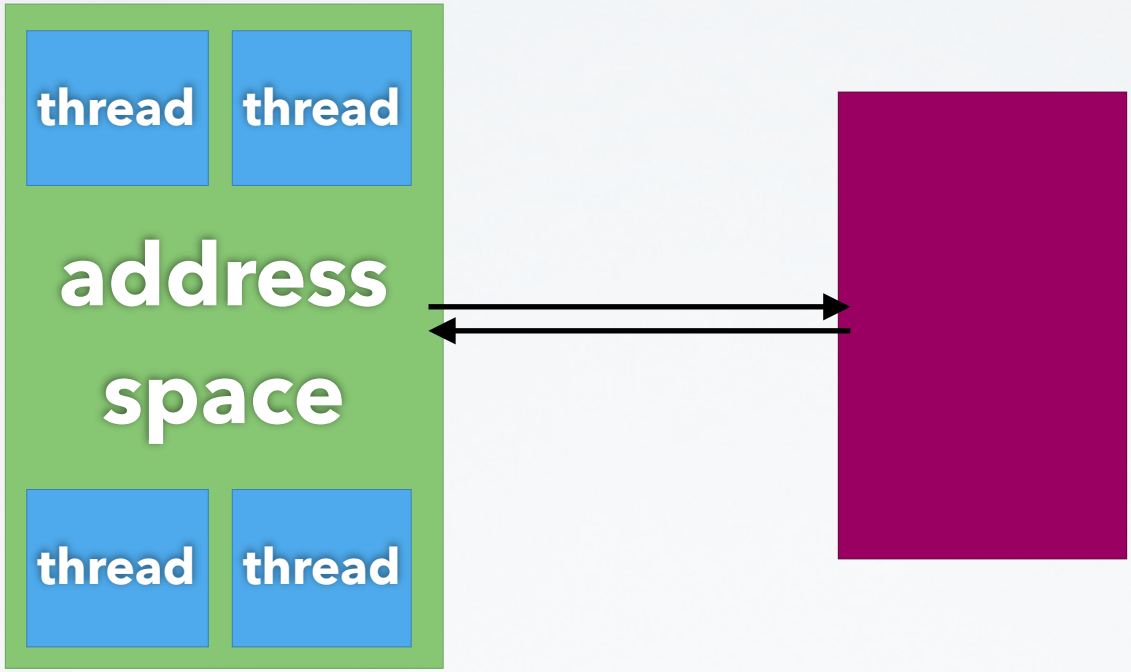
Franklin eBookMan

nearly 2 billion
Qualcomm boards

Cellphone Baseband processors

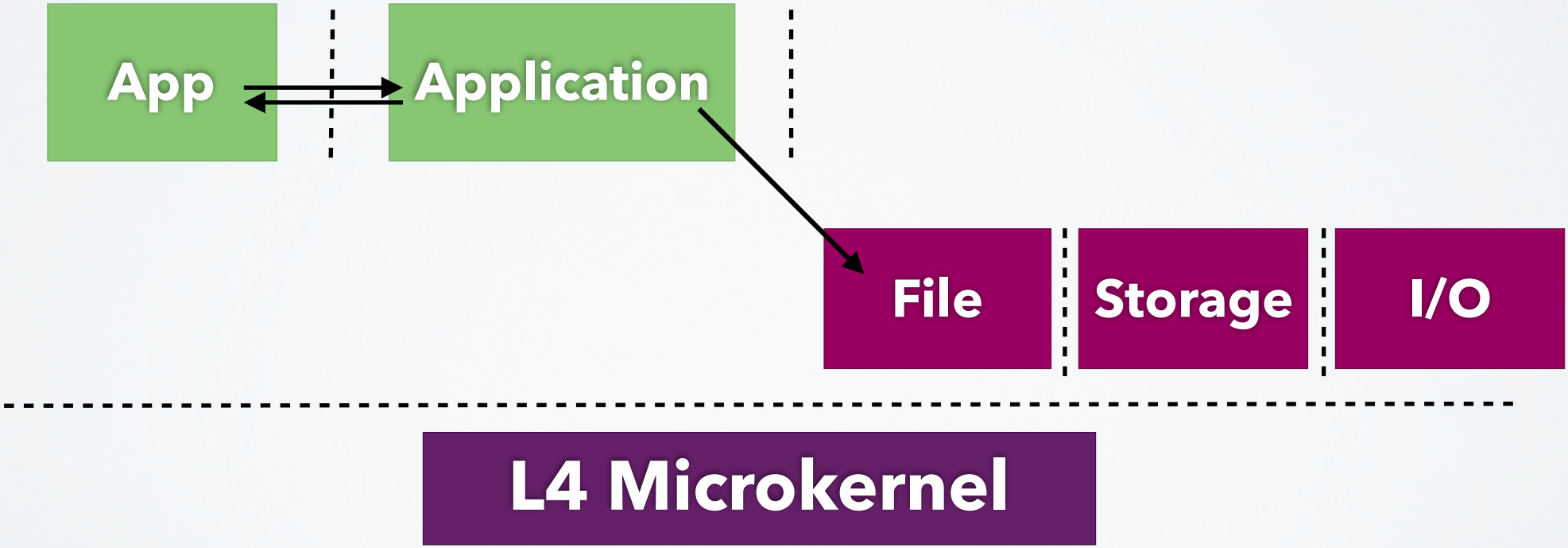
Airbus

3 ABSTRACTIONS

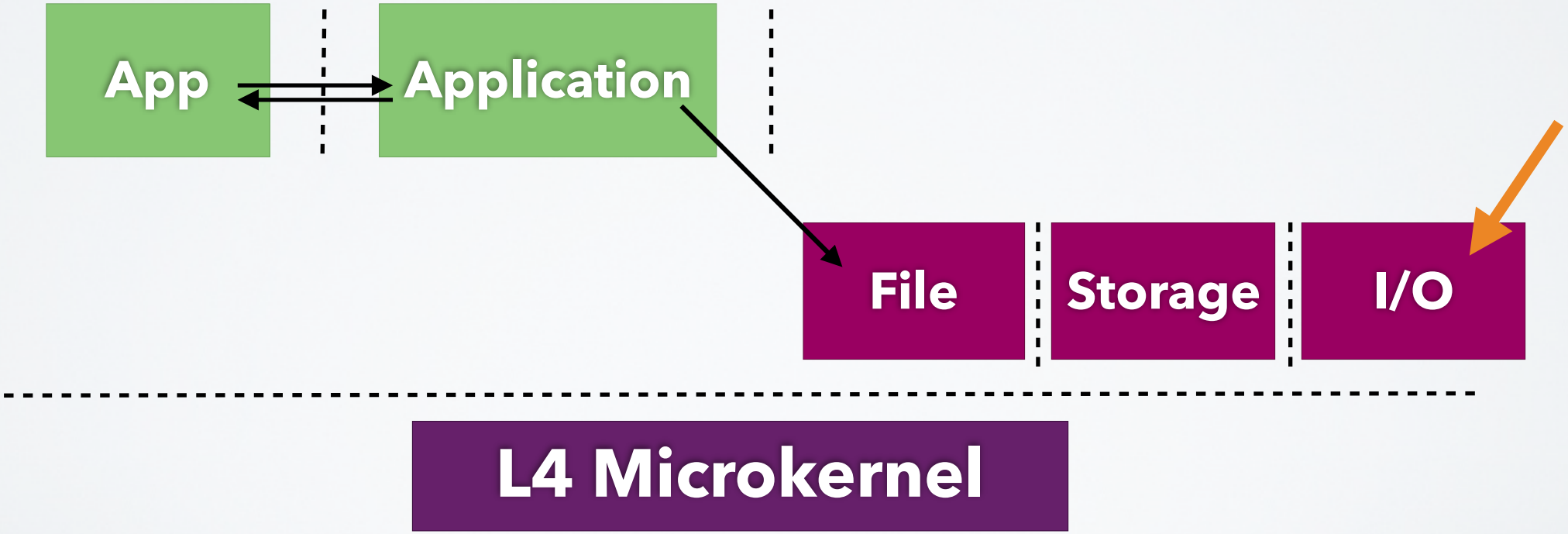


L4 Microkernel

MESSAGE PASSING



MESSAGE PASSING



MESSAGE PASSING

interrupt response (unblocking):

- HW-IRQ → Kernel-Handler → SemUp → iret idle → schedule → iret user_handler
- 900 cycles, 0.3 micro second (best case)
  4500 cycles (in deeper sleep level)
- Quad-Core Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz

LINUX ON L4

Linux OS

File | Storage | I/O

L4 Microkernel

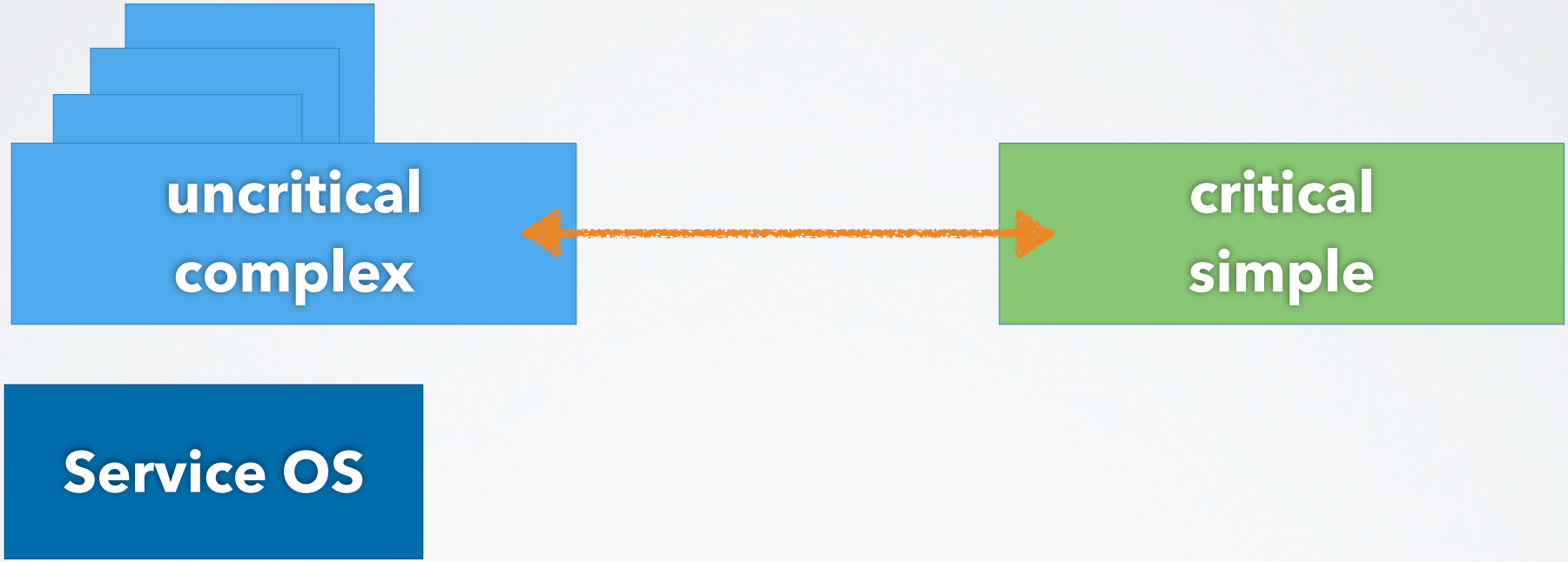
HYBRID SYSTEM

real-time

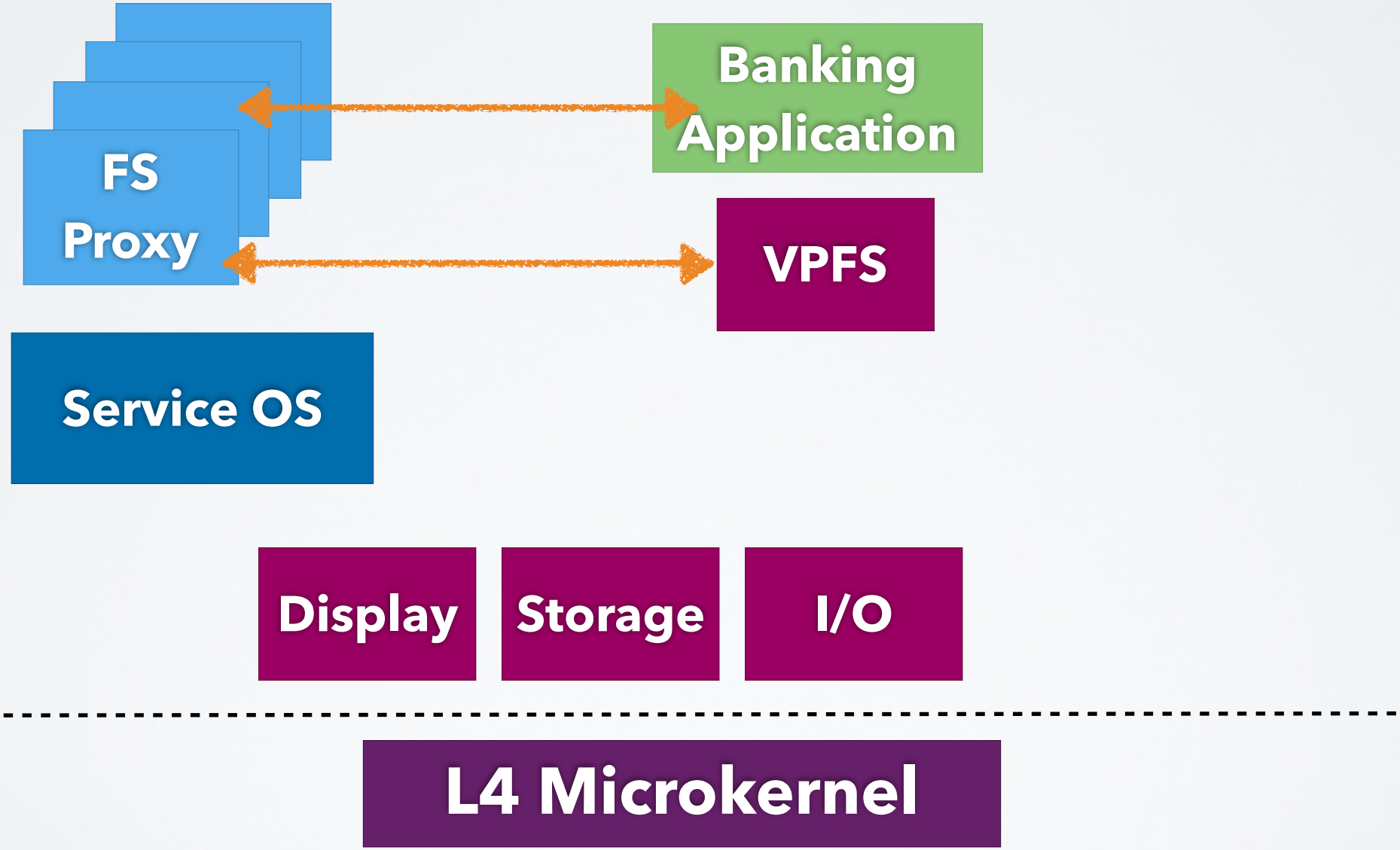
security: small Trusted Computing Base

resilience: small Reliable Computing Base

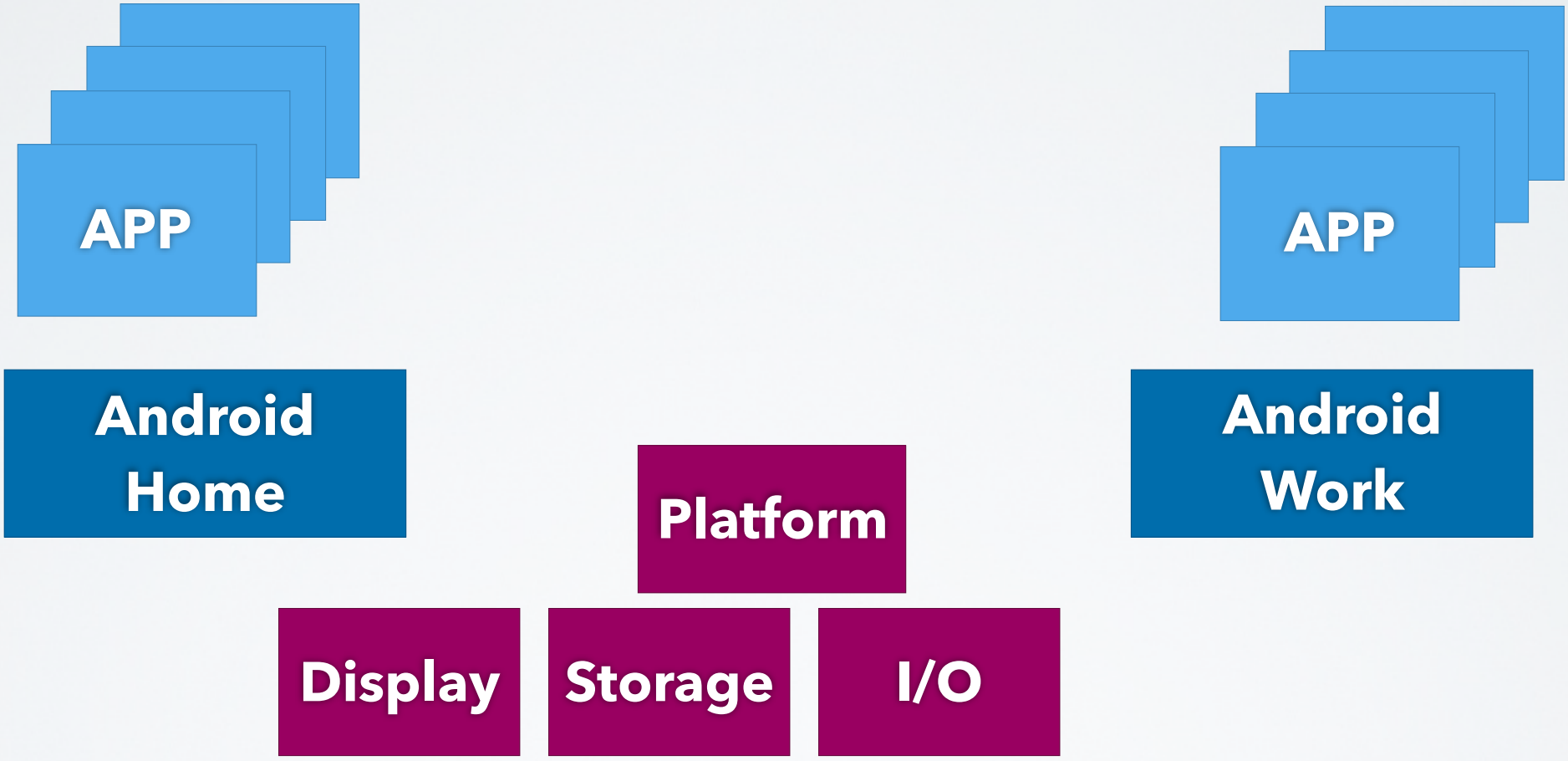
HYBRID SYSTEM



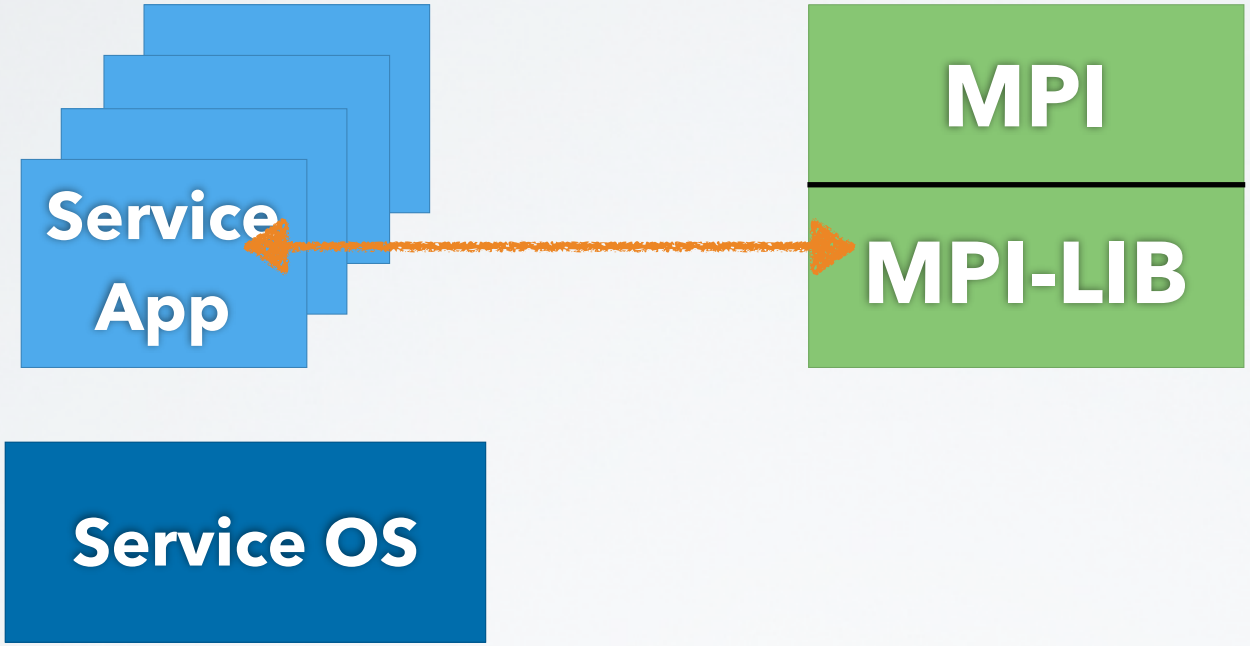
HYBRID SYSTEM

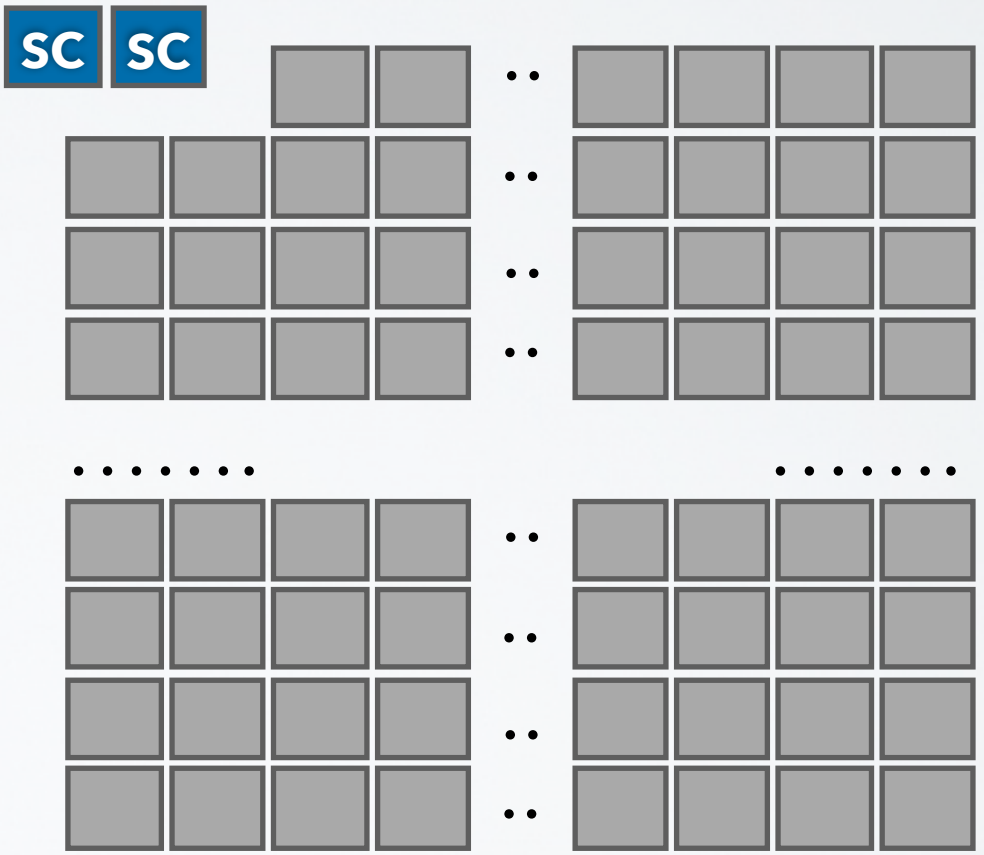


HYBRID SYSTEM

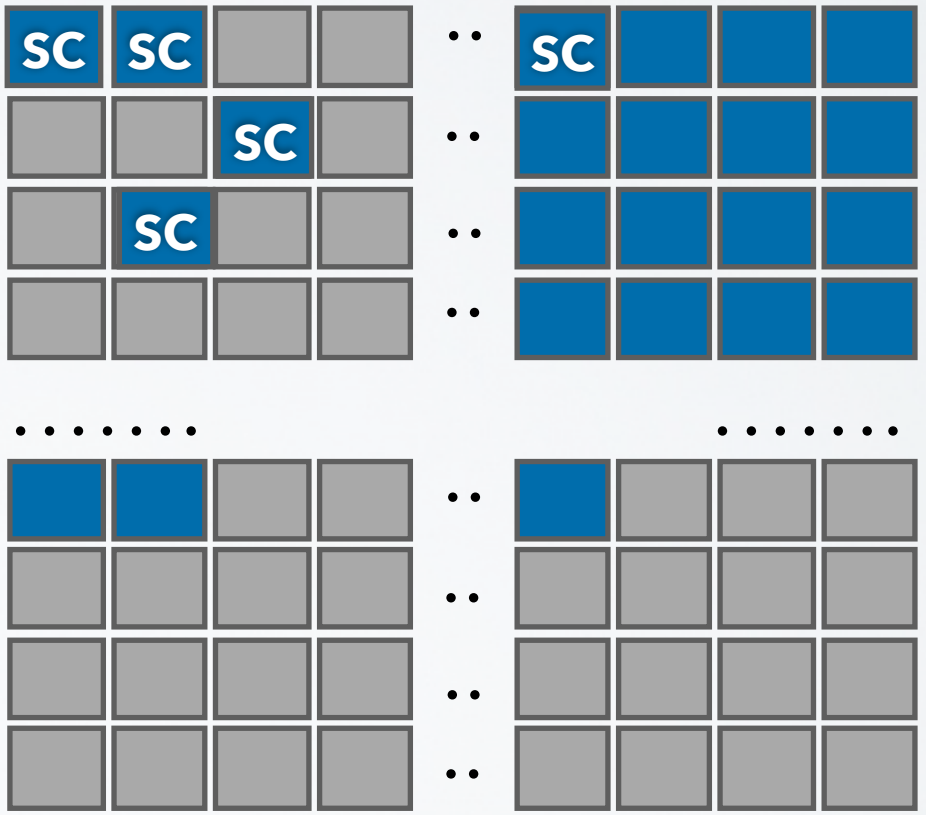


HYBRID SYSTEM

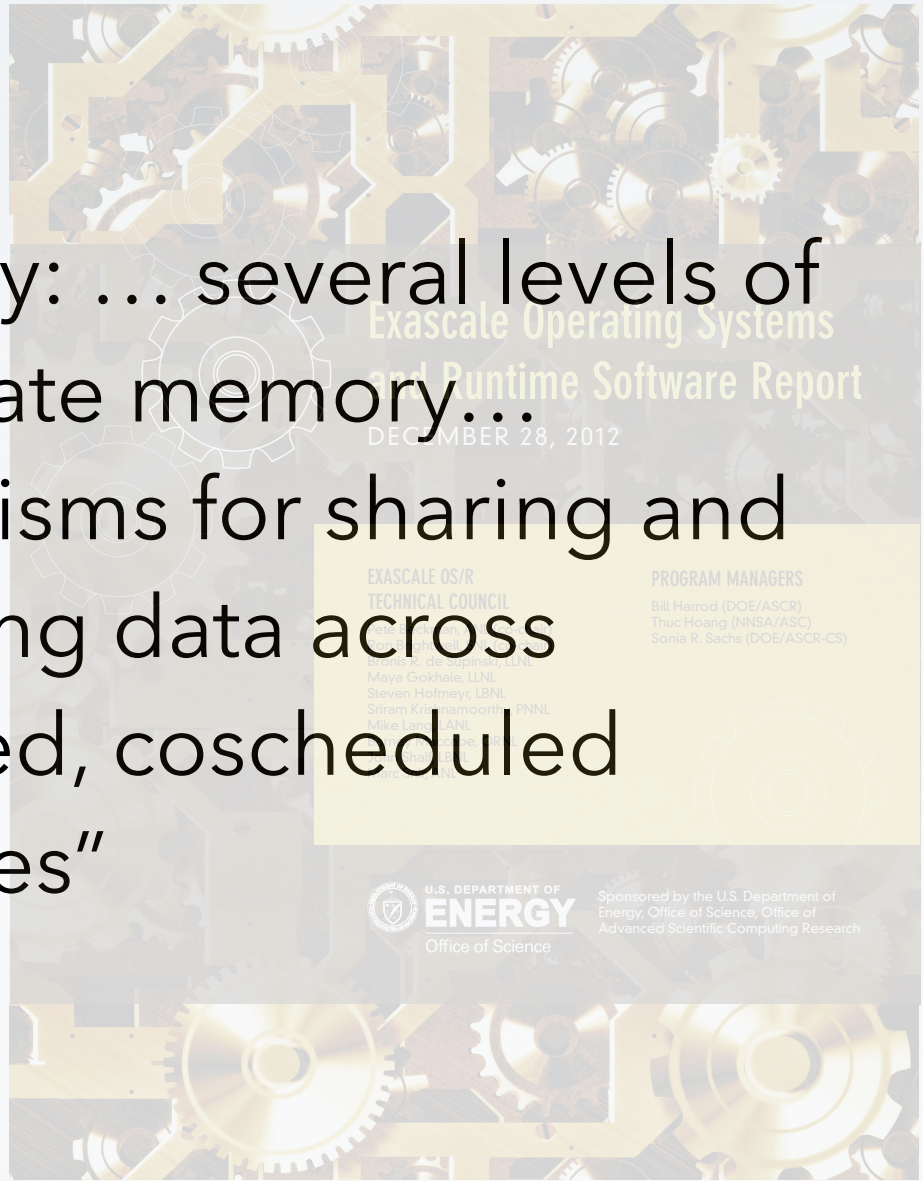



SC
SC

LINUX & SERVICE

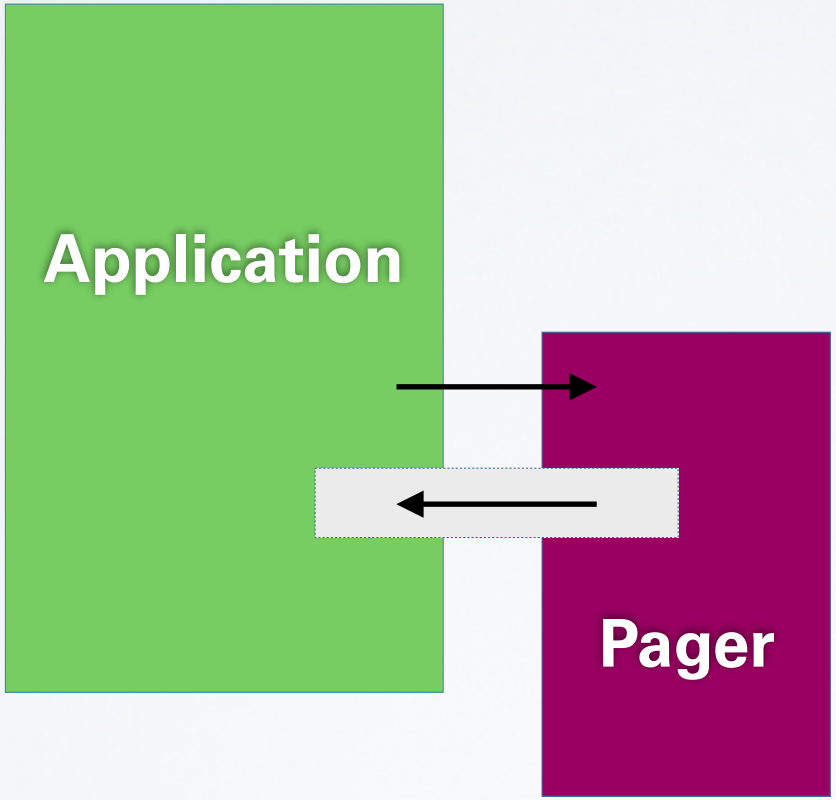


PAGERS



"Memory: … several levels of solid- state memory… mechanisms for sharing and protecting data across colocated, coscheduled processes"

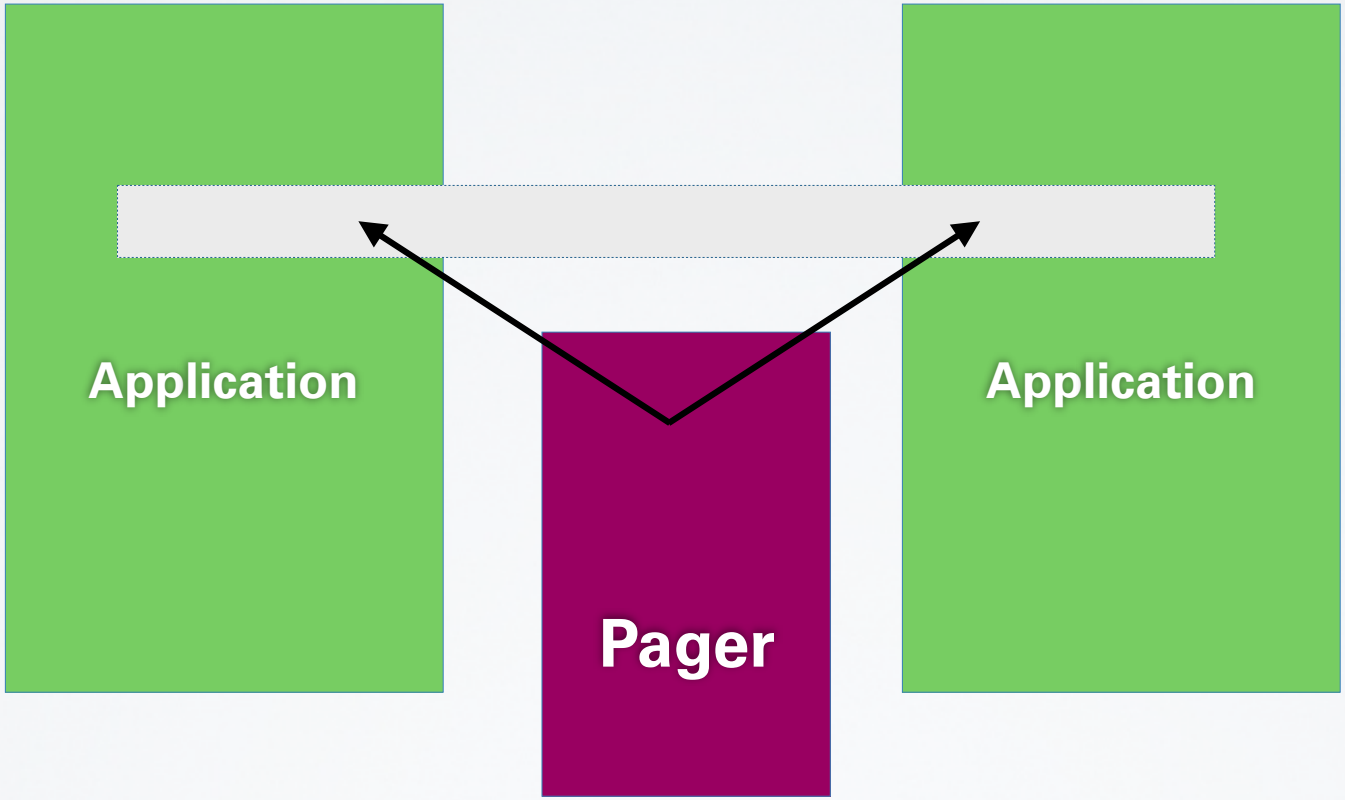
PAGERS



page fault
send page

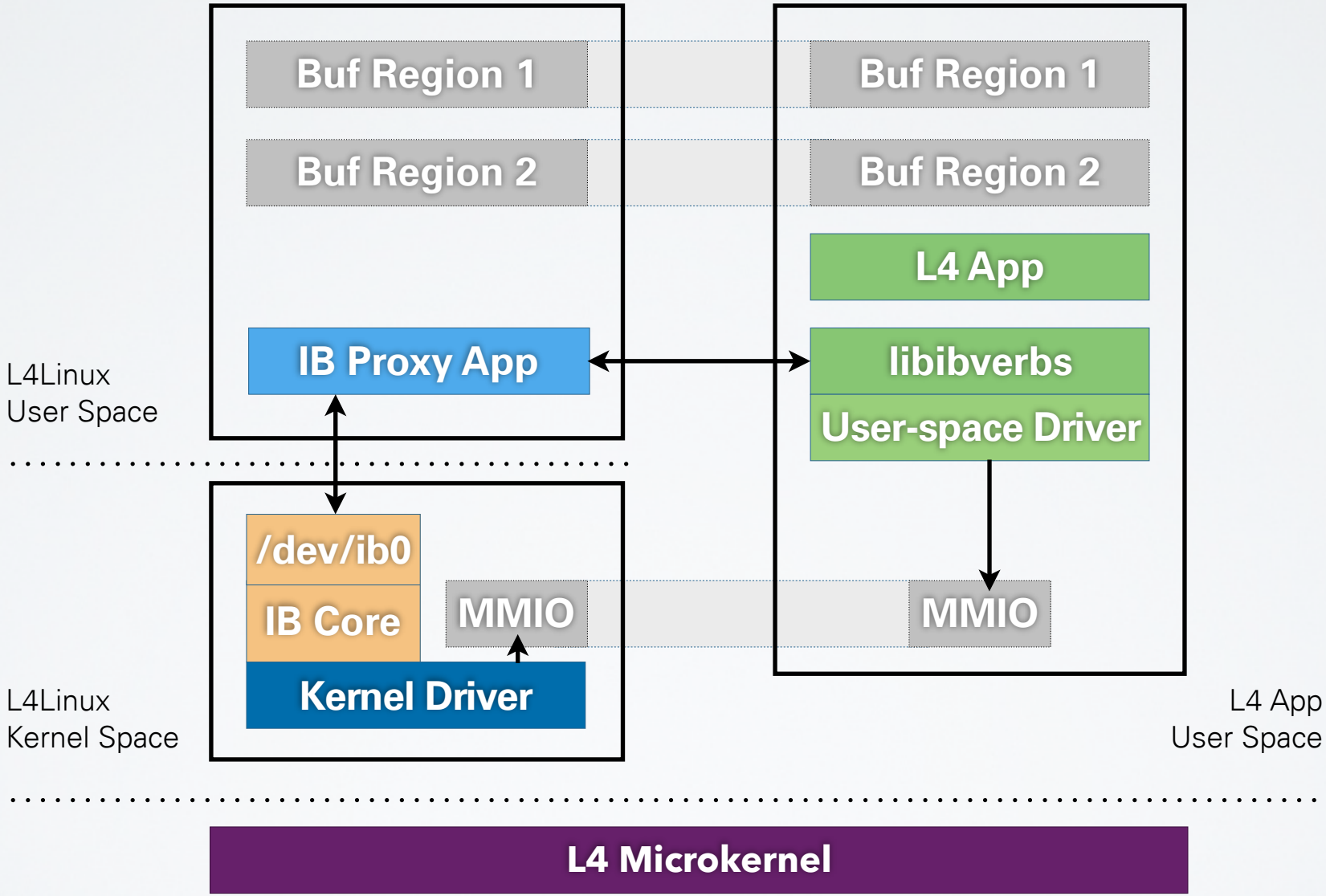
L4 Microkernel

PAGER

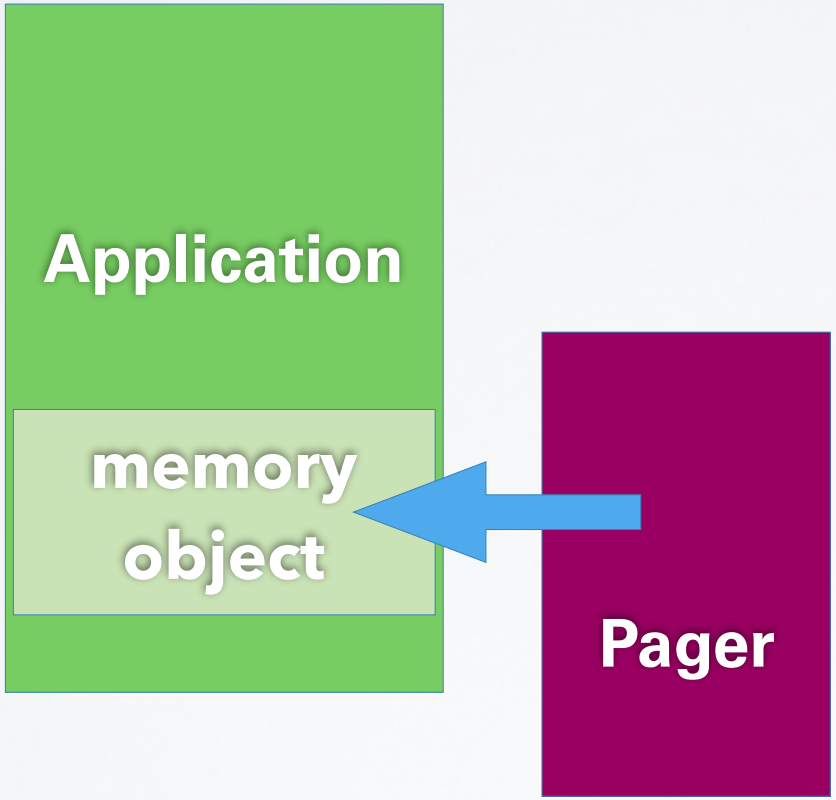


L4 Microkernel

SPLIT INFINIBAND



PAGER



L4 Microkernel

CHALLENGE

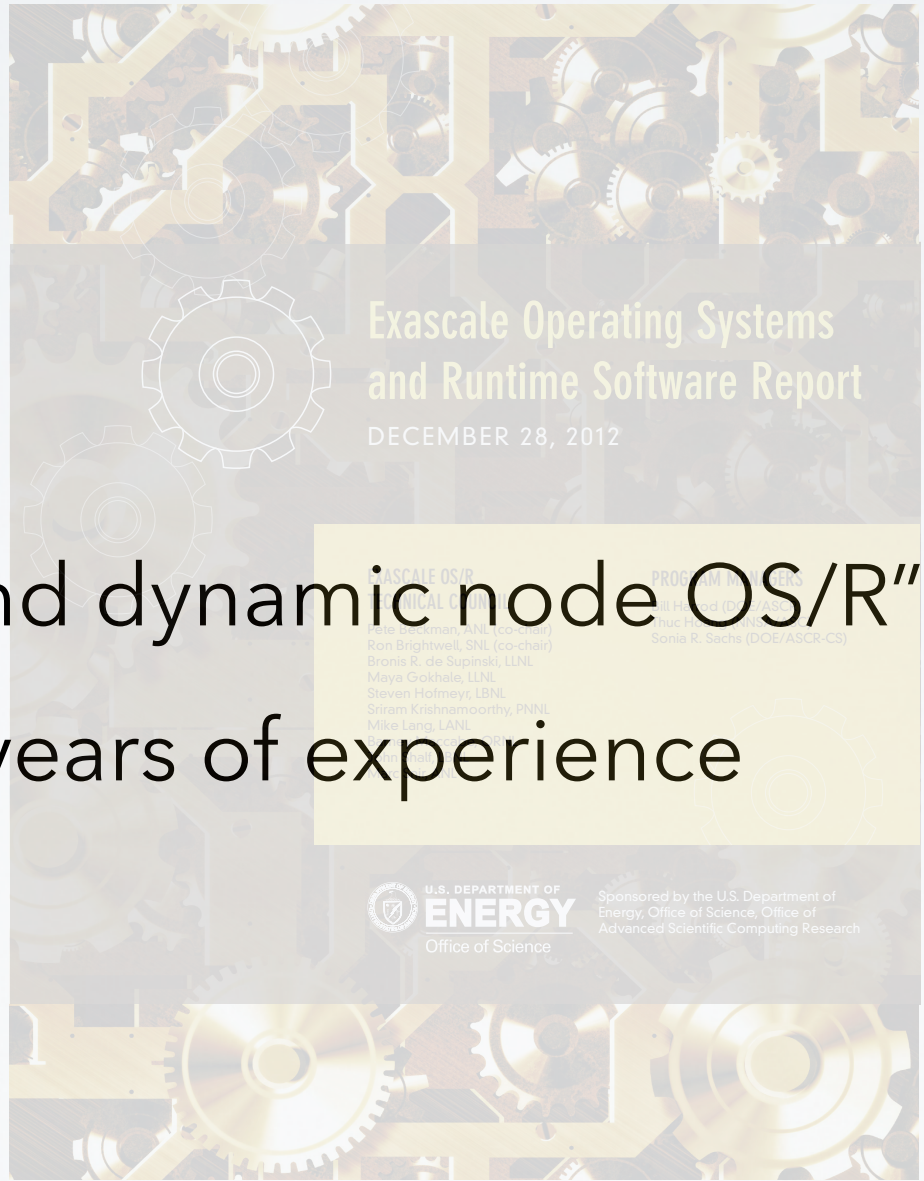
interfacing with app and smart run-time

- division of work with smart run-time (e.g., Charm++ decomposition, balancing)
- impact on communication latency
- application level knowledge
  resource usage, communication affinity

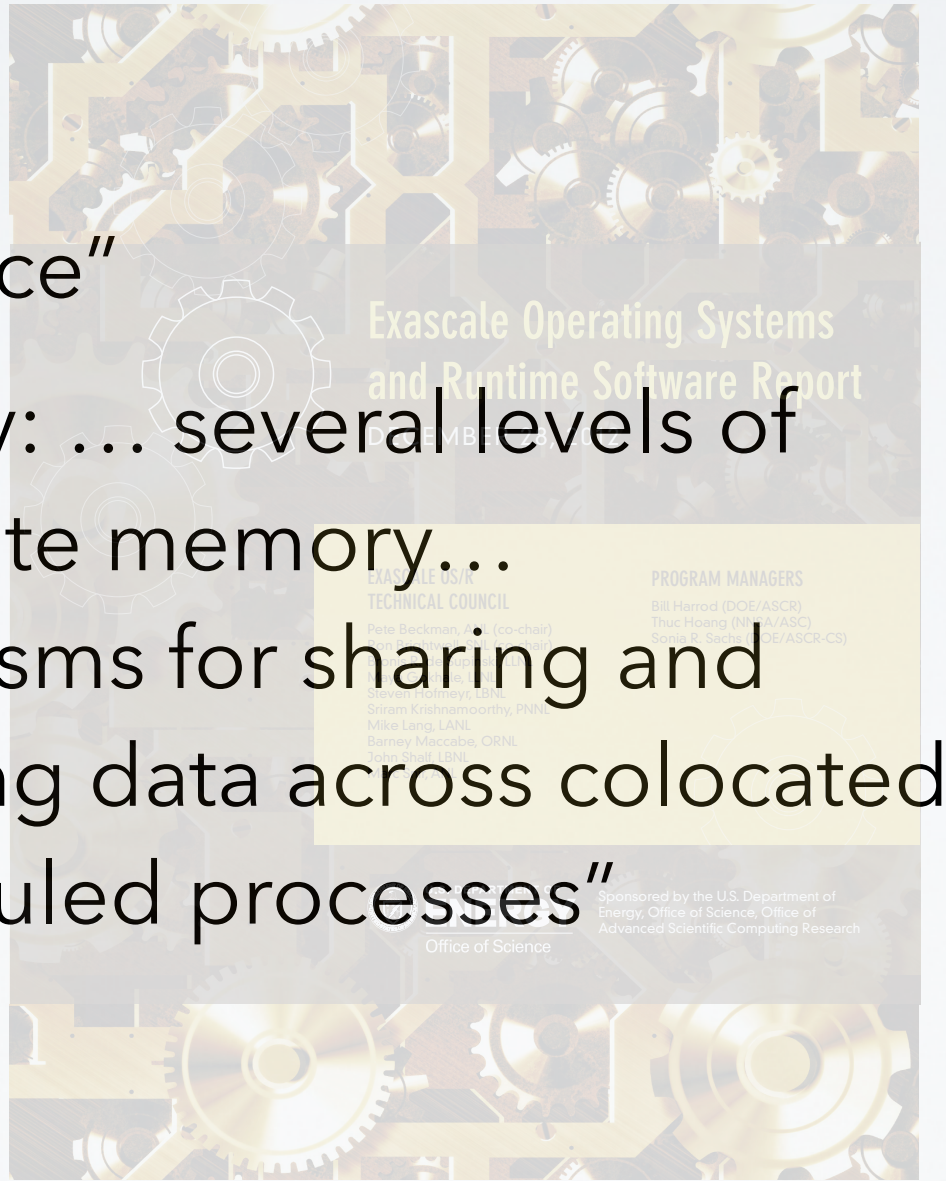
basis for

"agile and dynamic node OS/R"

with 20 years of experience



HPC FUTURE

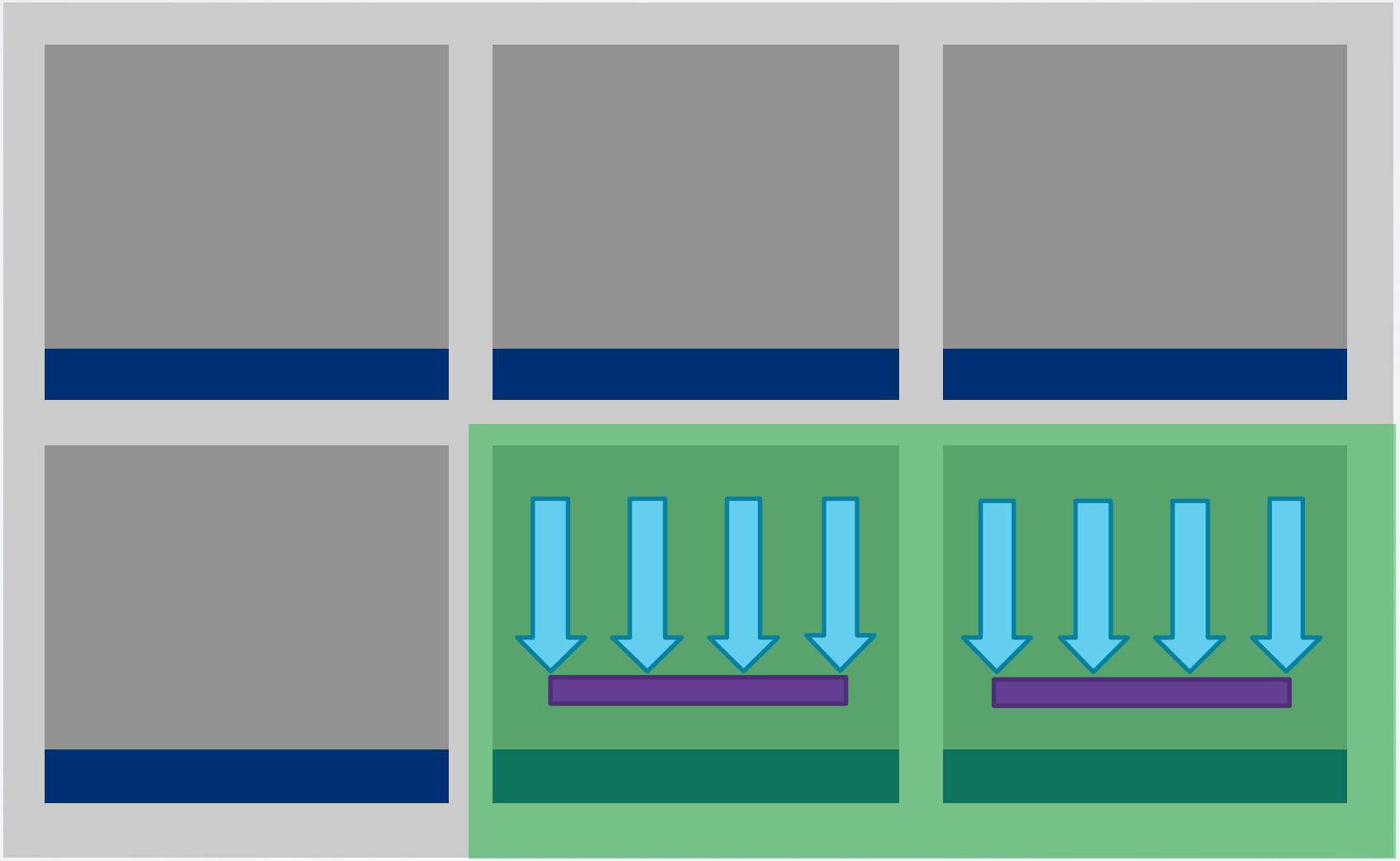


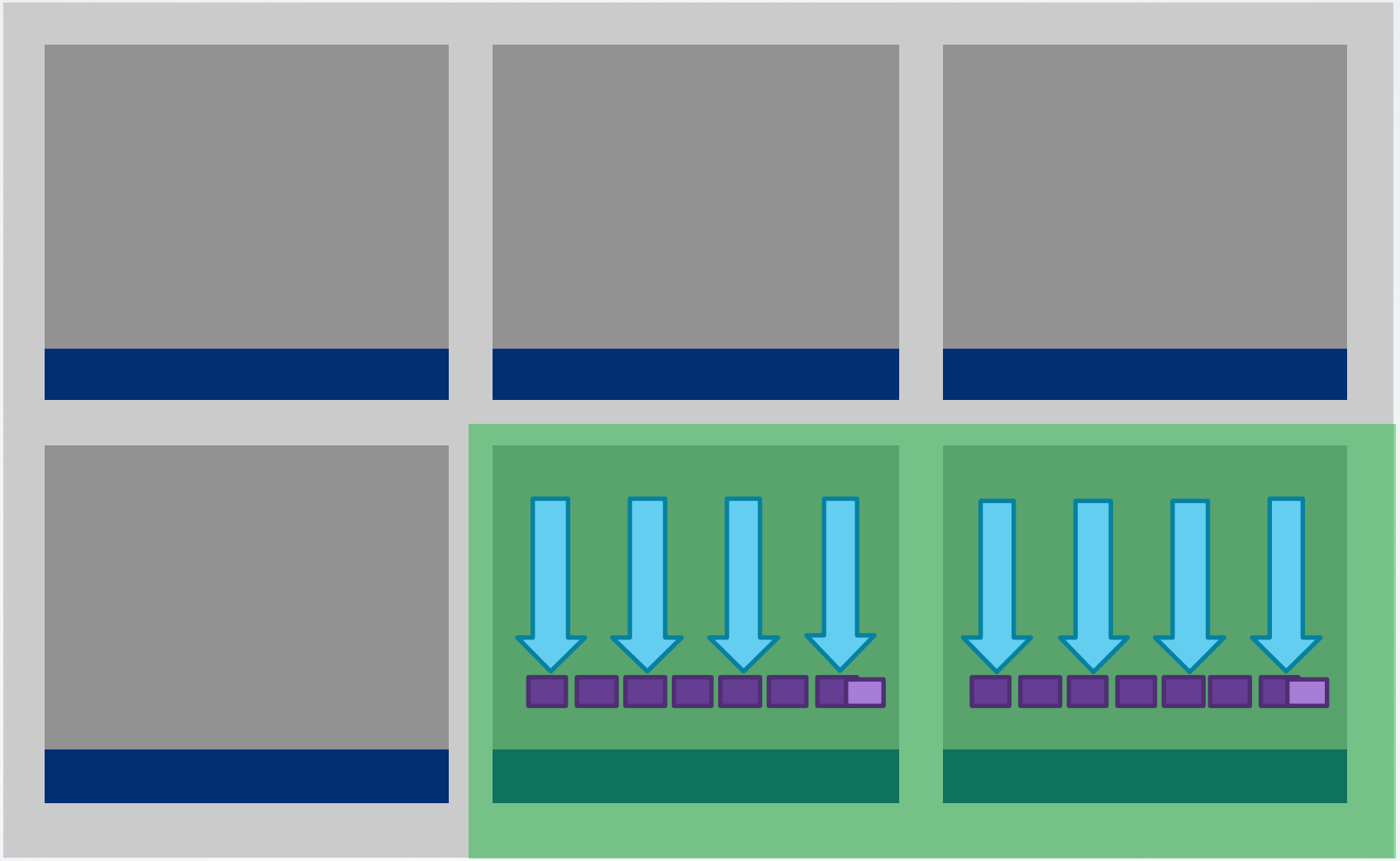
"Resilience"

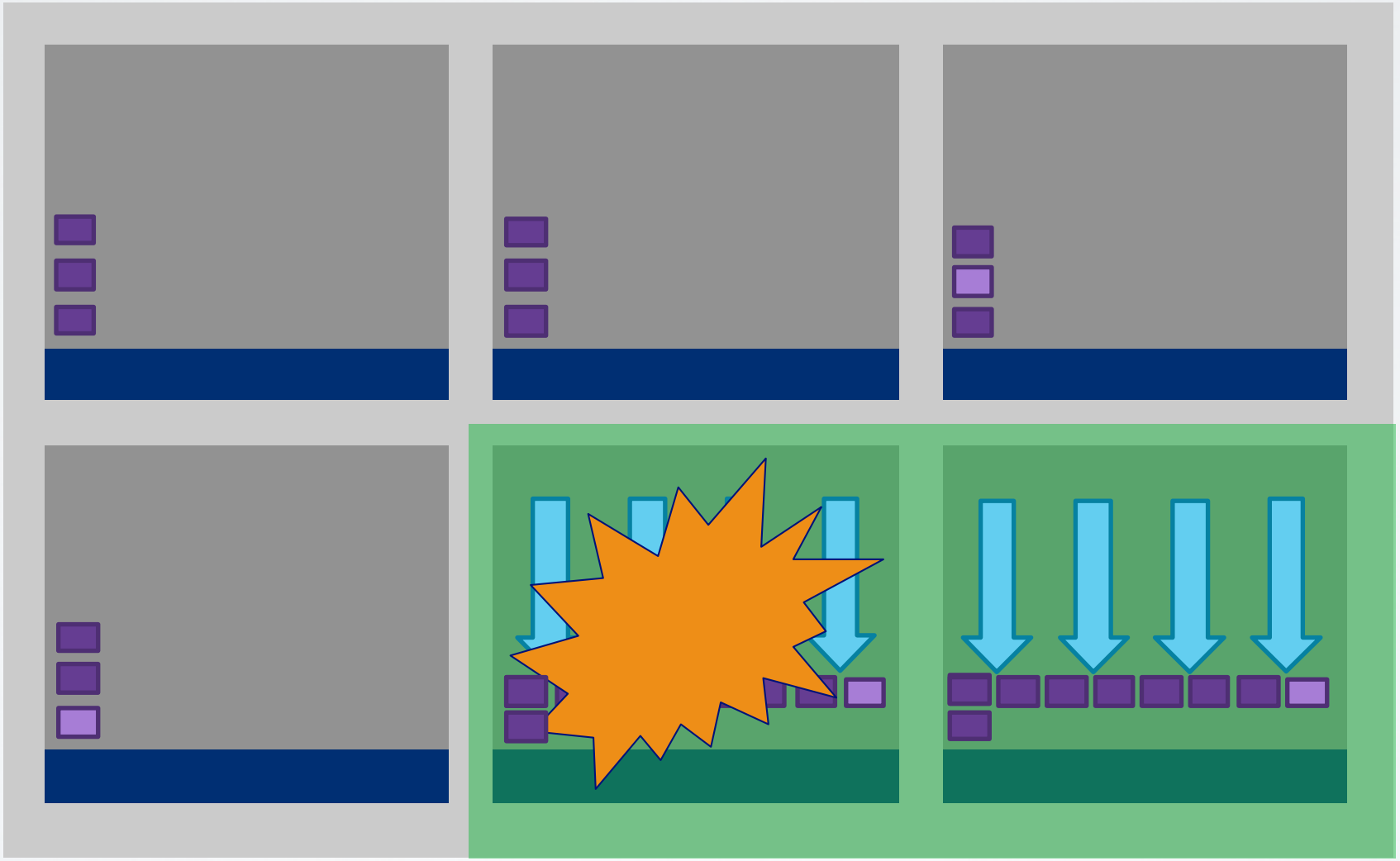
"Memory: … several levels of solid- state memory… mechanisms for sharing and protecting data across colocated, coscheduled processes"

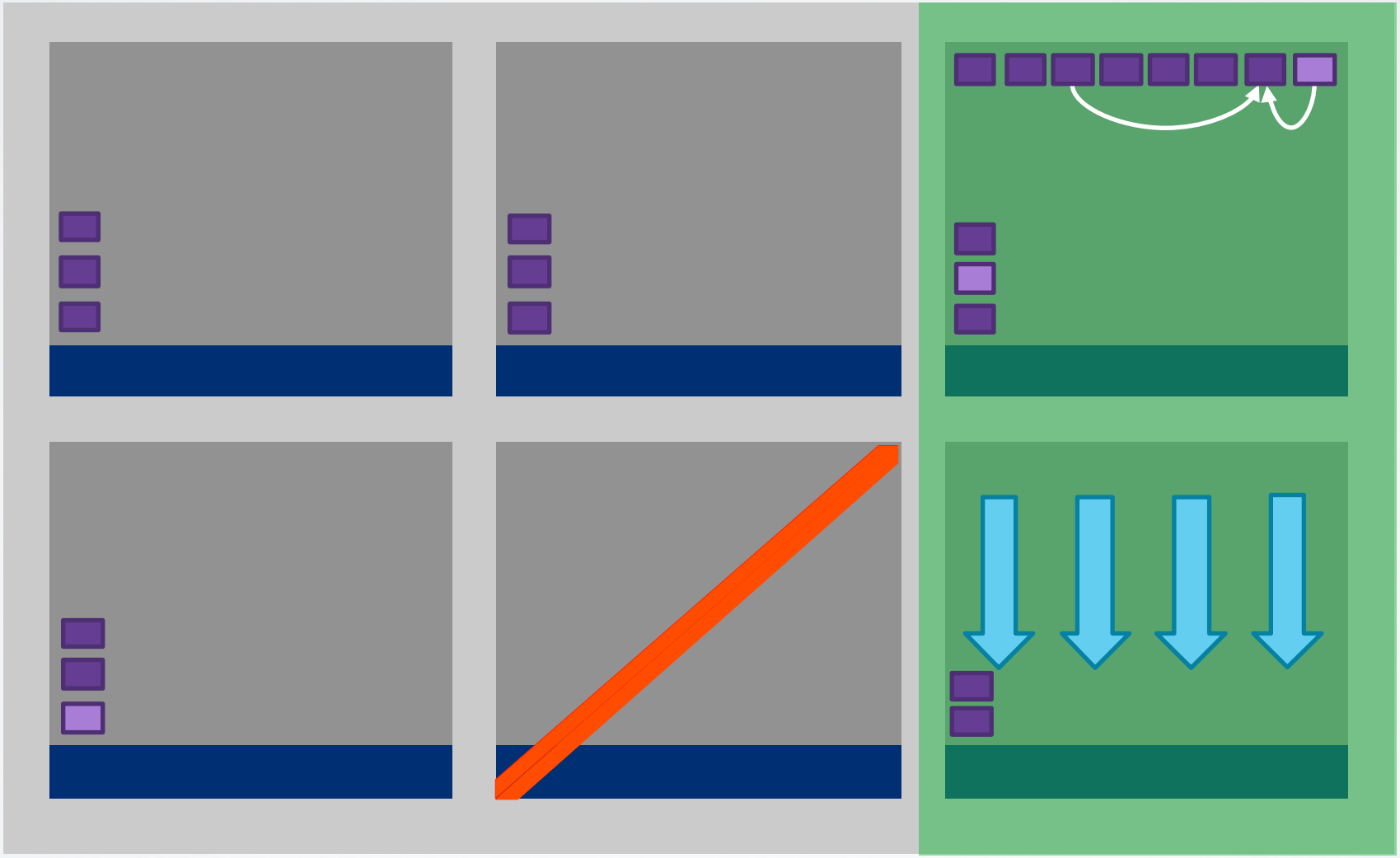
HPC FUTURE

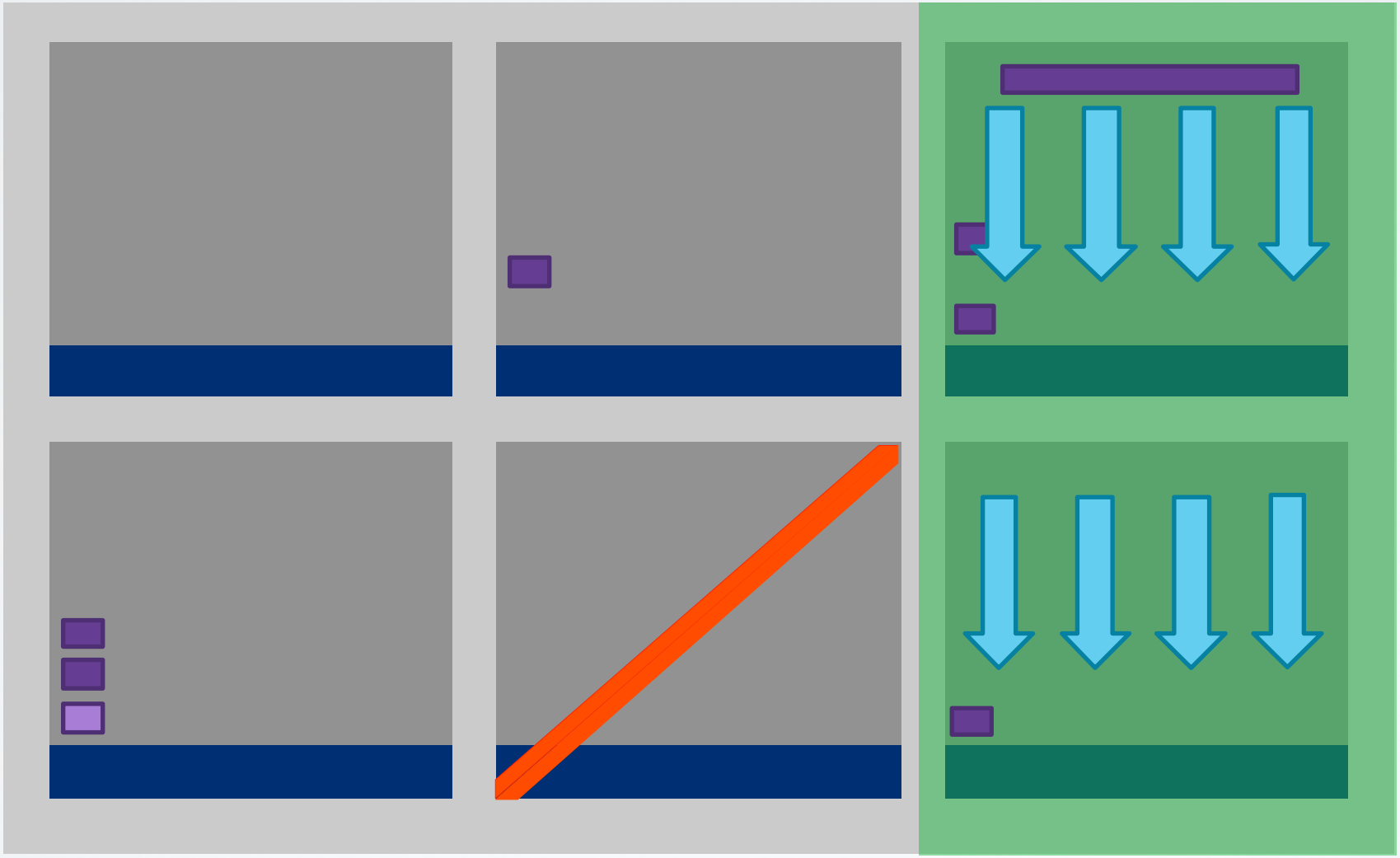
- higher failure rates
- diverse node-storage types
  - persistance
  - low-power DRAM



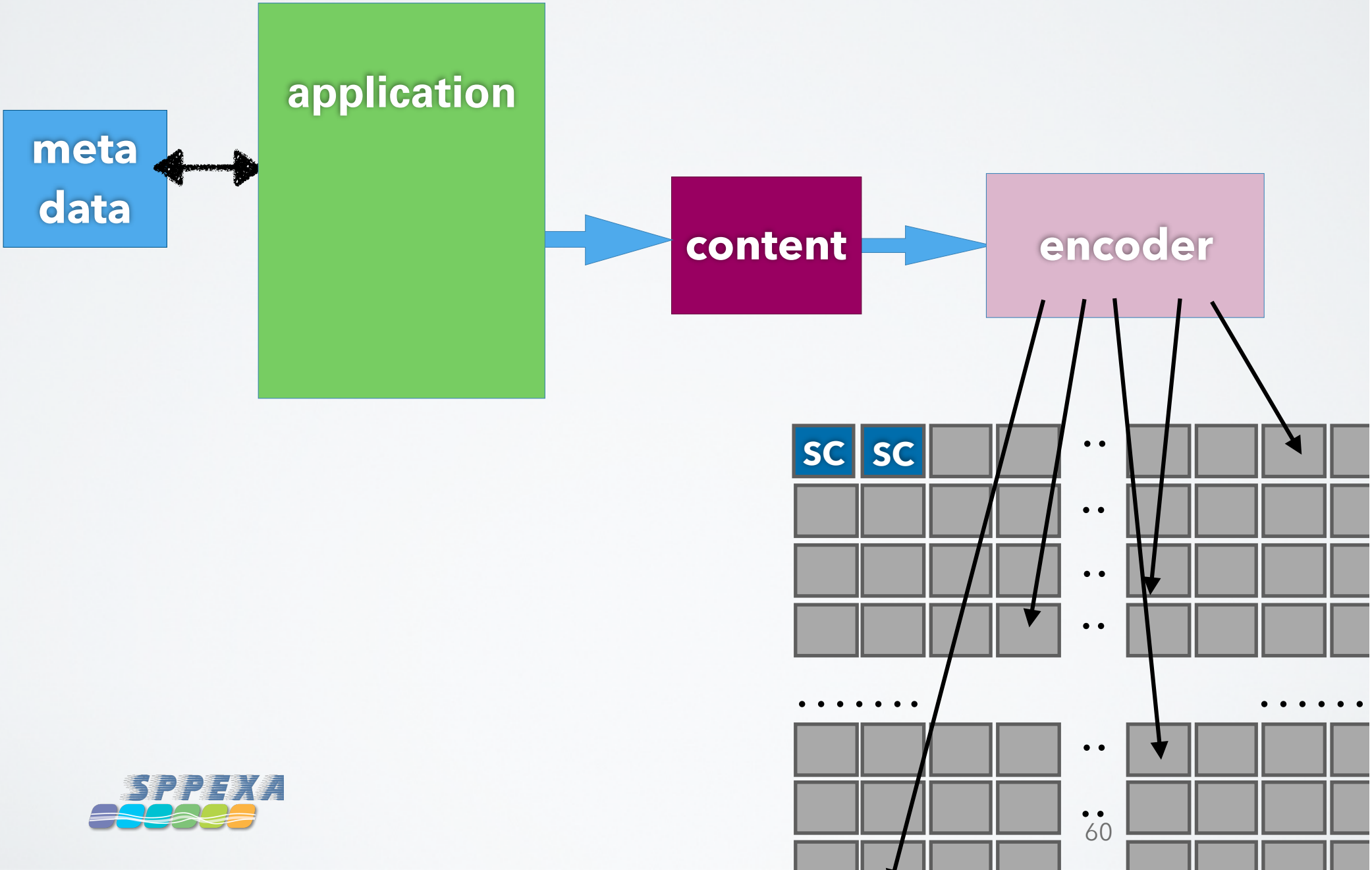




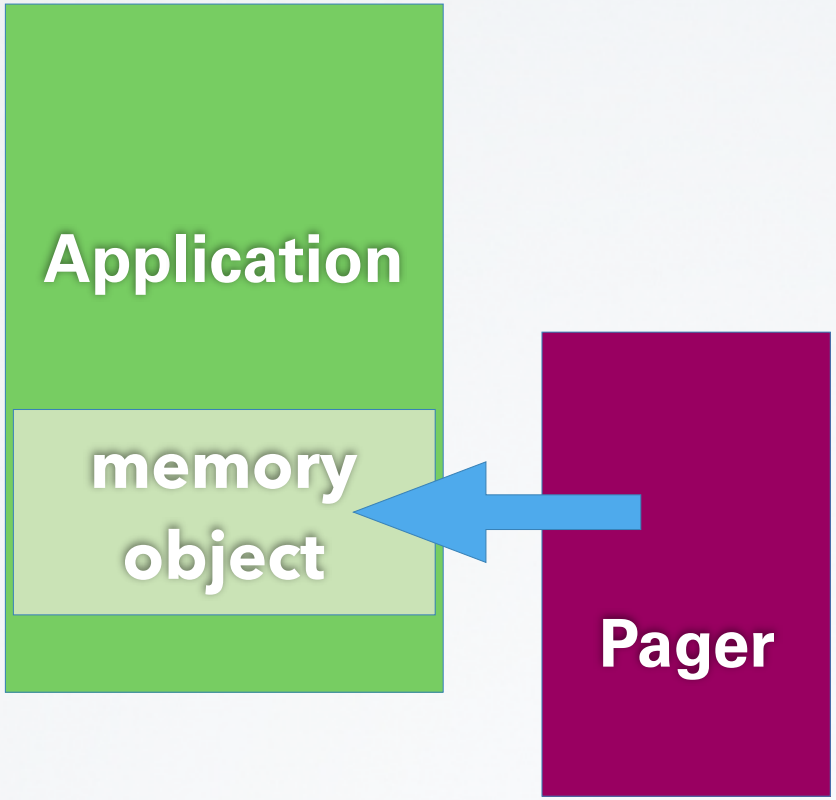




IN-MEMORY FS

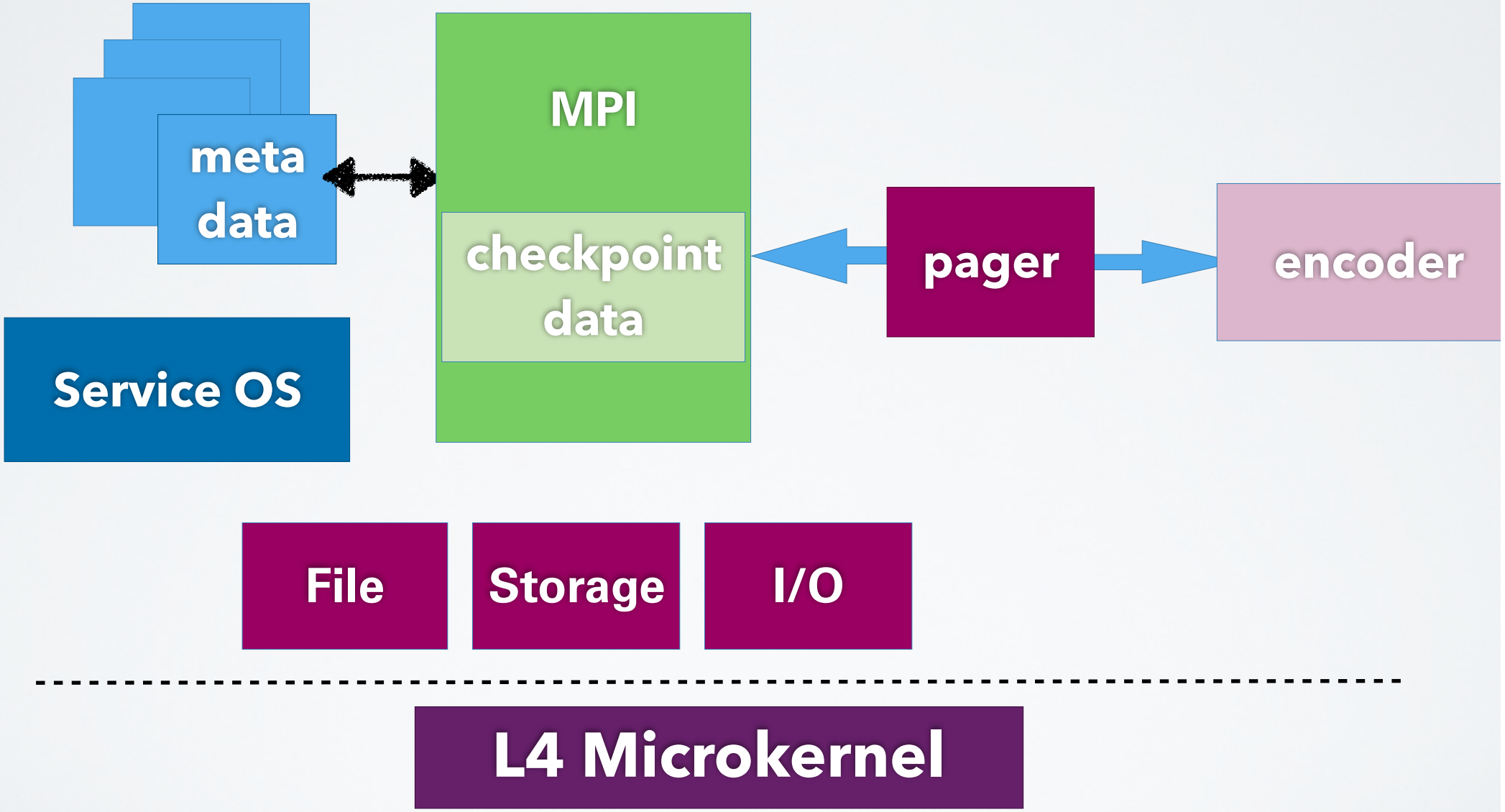


PAGER



L4 Microkernel

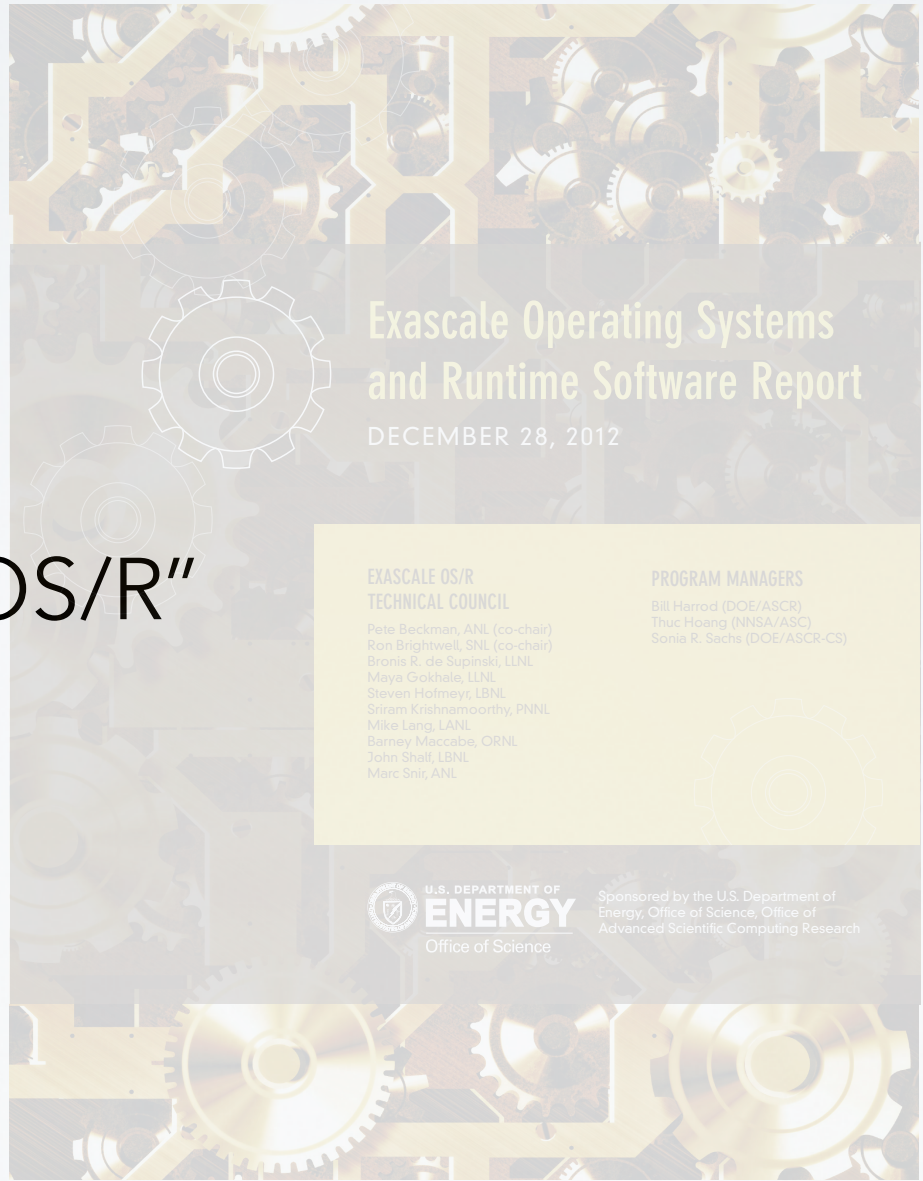
HYBRID

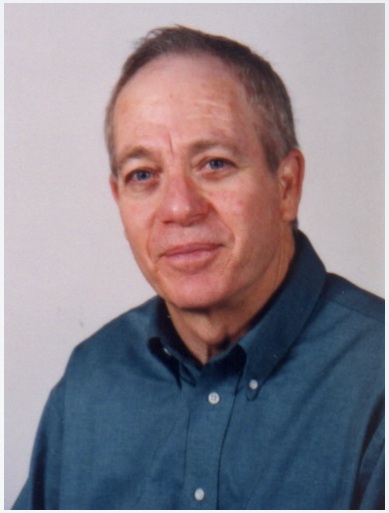

meta data
MPI
checkpoint data
pager
encoder
Service OS
File
Storage
I/O
L4 Microkernel

CHALLENGES

- application ./. run time ./. OS interface
- allocation of encoded chunks
- global information on failures

- "Global OS/R"


Exascale Operating Systems and Runtime Software Report
DECEMBER 28, 2012
EXASCALE OS/R TECHNICAL COUNCIL
Pete Beckman, ANL (co-chair)
Ron Brightwell, SNL (co-chair)
Bronis R. de Supinski, LLNL
Maya Gokhale, LLNL
Steven Hofmeyr, LBNL
Sriram Krishnamoorthy, PNNL
Mike Lang, LANL
Barney Maccabe, ORNL
John Shalf, LBNL
Marc Snir, ANL
PROGRAM MANAGERS
Bill Harrod (DOE/ASCR)
Thuc Hoang (NNSA/ASC)
Sonia R. Sachs (DOE/ASCR-CS)
U.S. DEPARTMENT OF ENERGY
Office of Science
Sponsored by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research



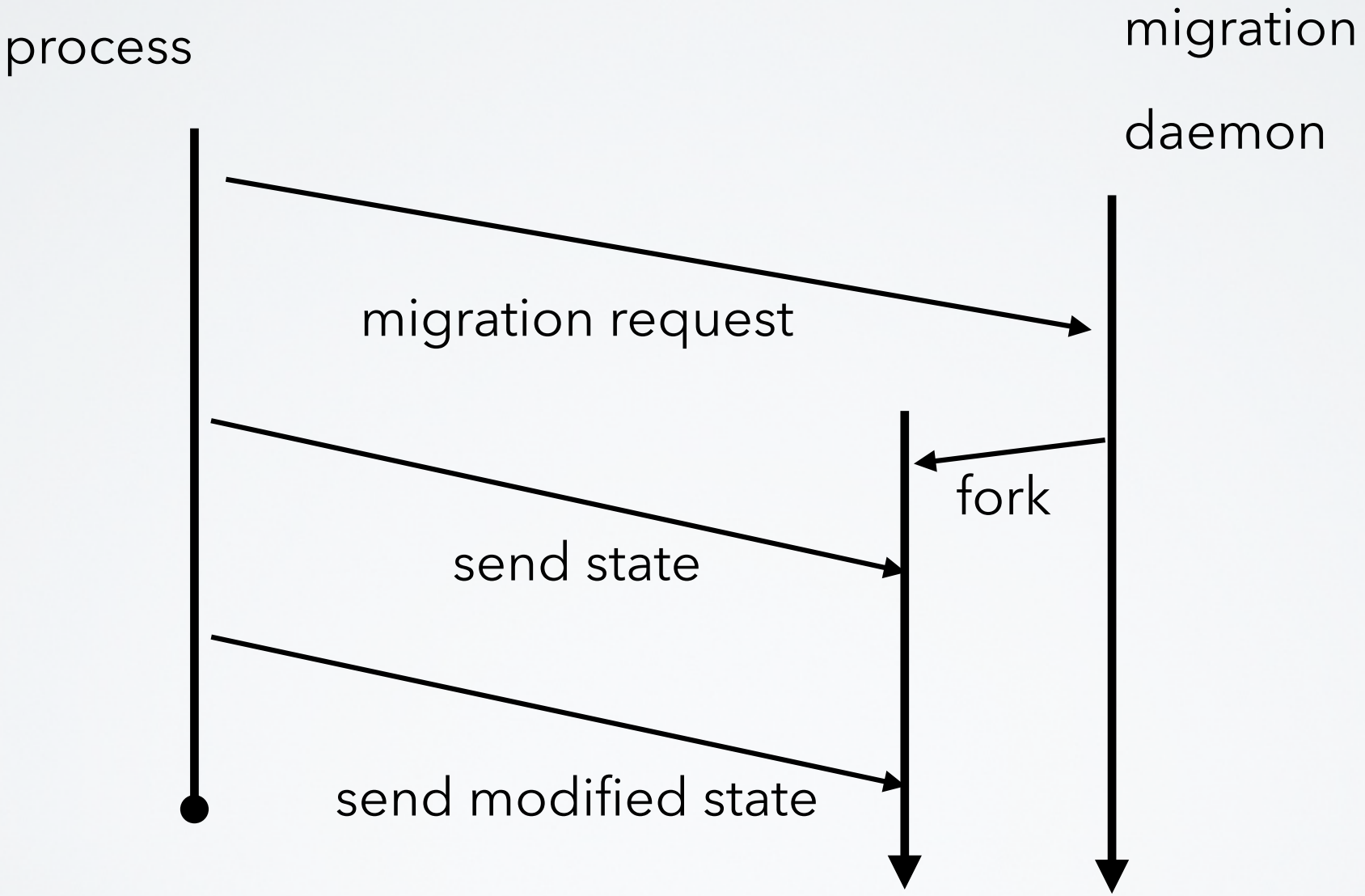
Amnon Barak

migration

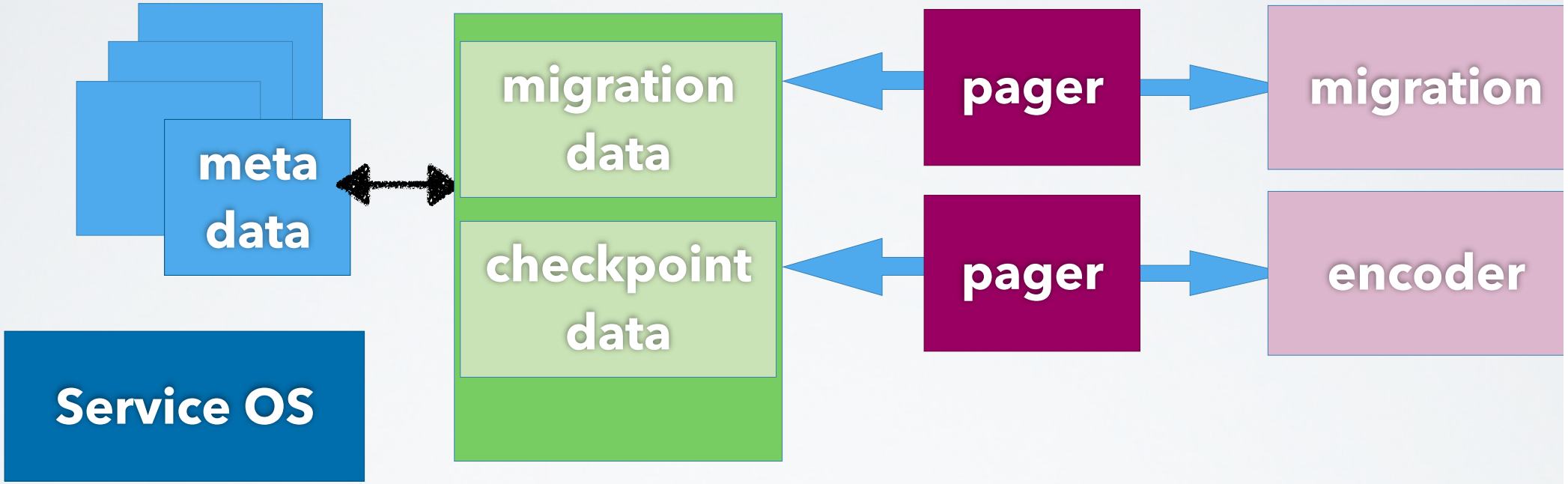
gossip

WWW

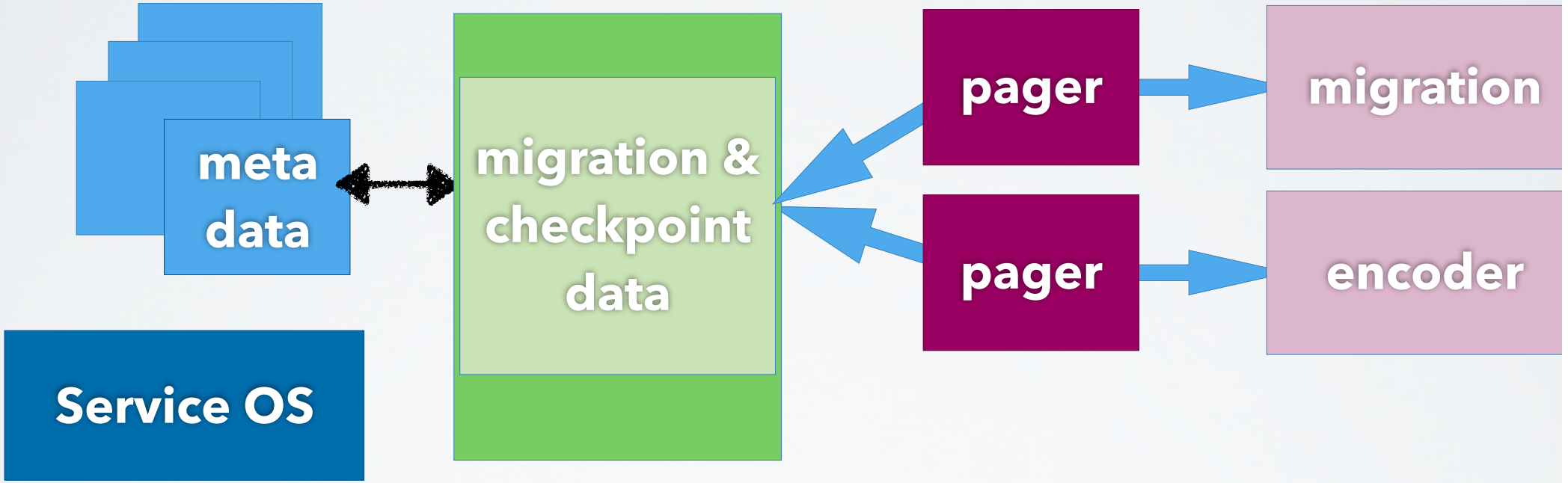
MIGRATION


process
migration daemon
migration request
fork
send state
send modified state

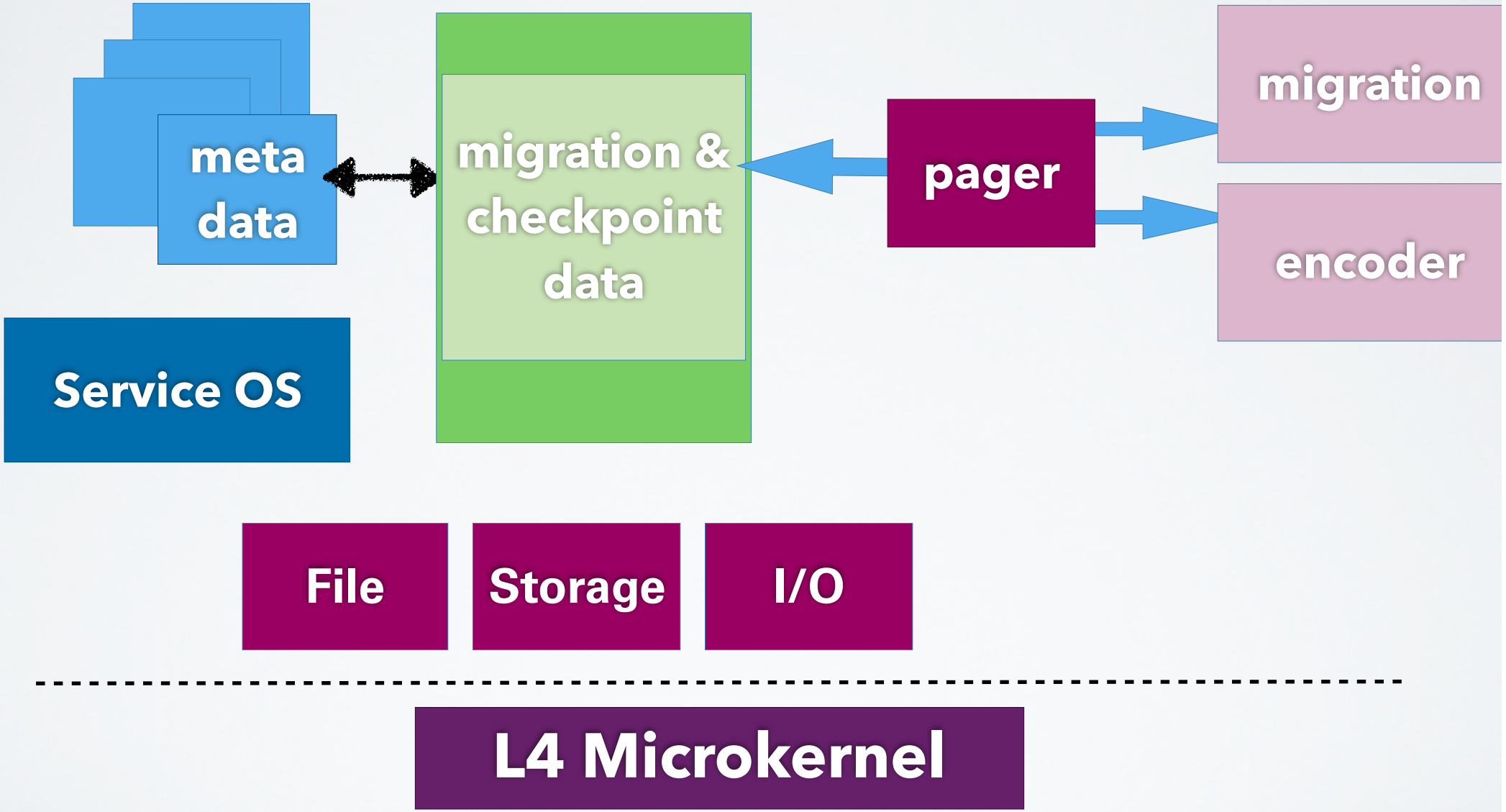
HYBRID


meta data
migration data
checkpoint data
pager
pager
migration
encoder
Service OS
File
Storage
I/O
L4 Microkernel

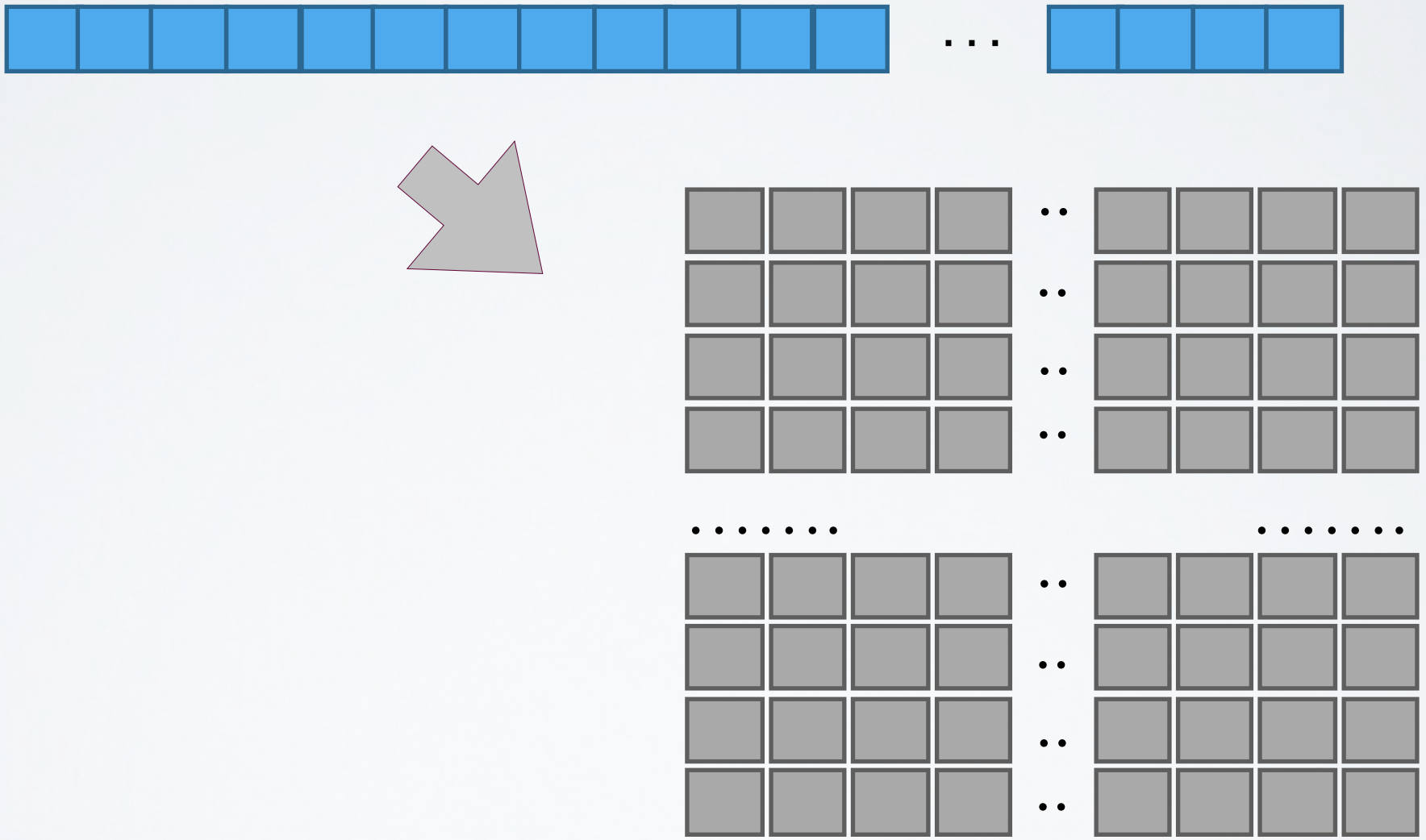
HYBRID


meta data
migration & checkpoint data
pager
pager
migration
encoder
Service OS
File
Storage
I/O
L4 Microkernel

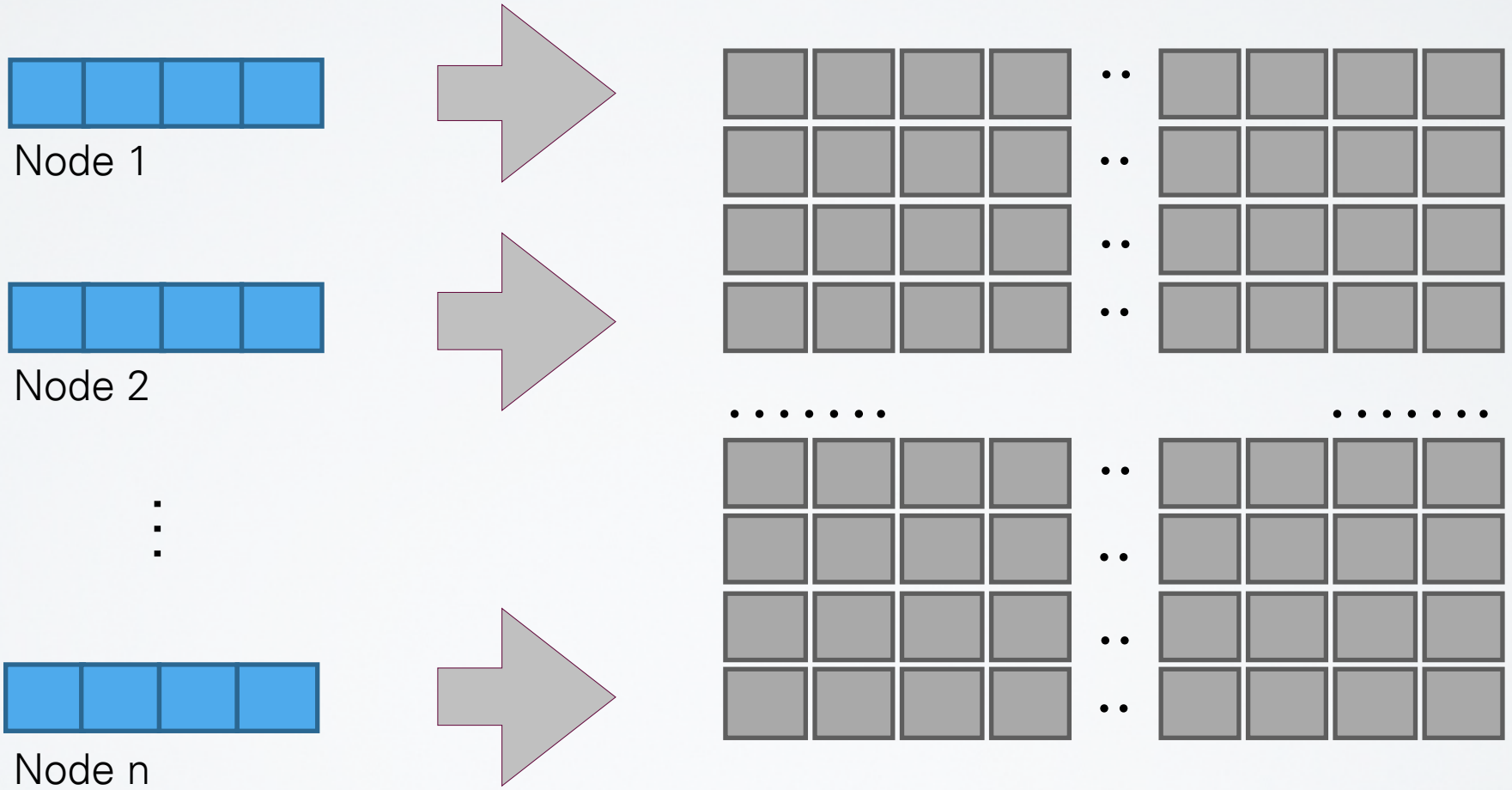
HYBRID


meta data
migration & checkpoint data
pager
migration
encoder
Service OS
File
Storage
I/O
L4 Microkernel

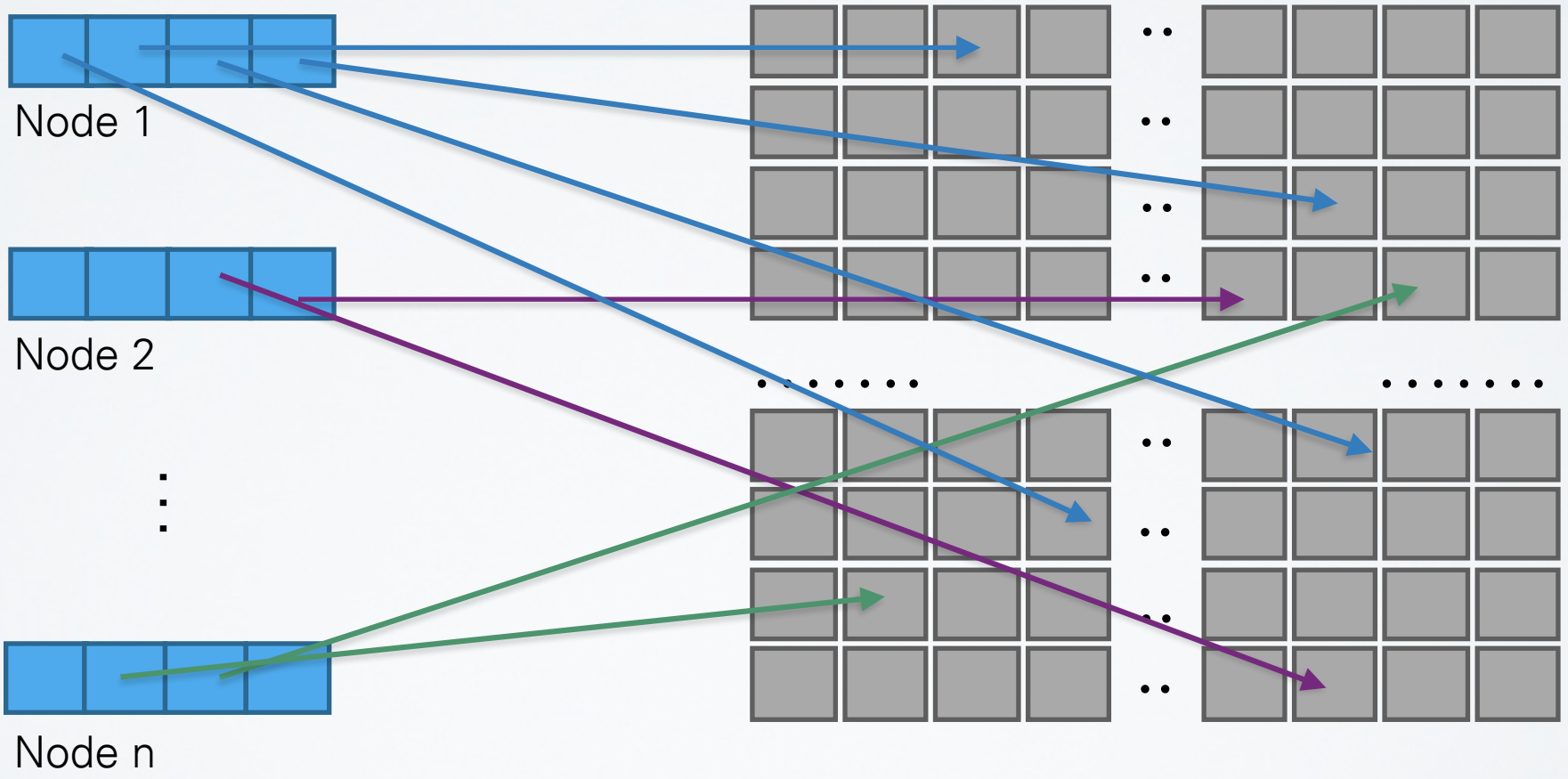
CENTRALIZED



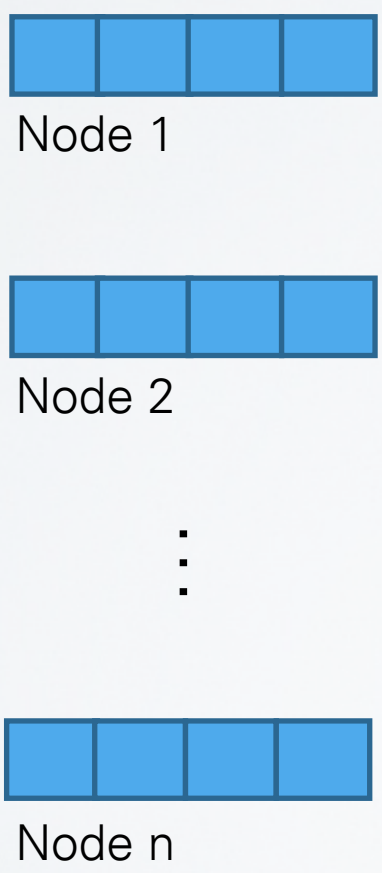
DECENTRALIZED

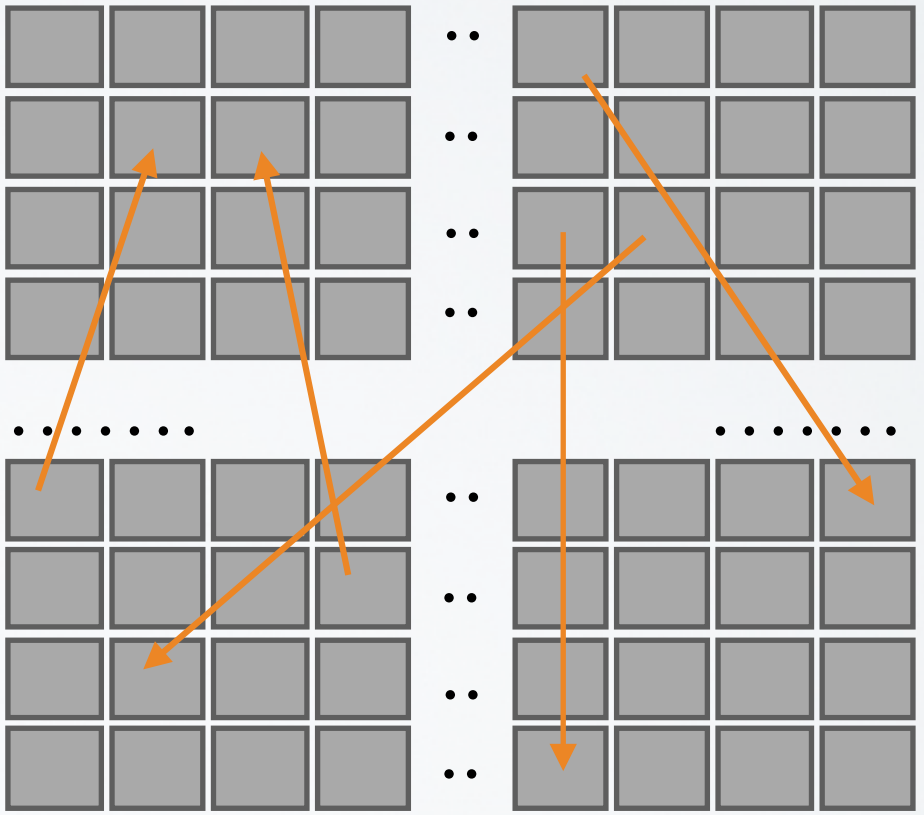


DECENTRALIZED

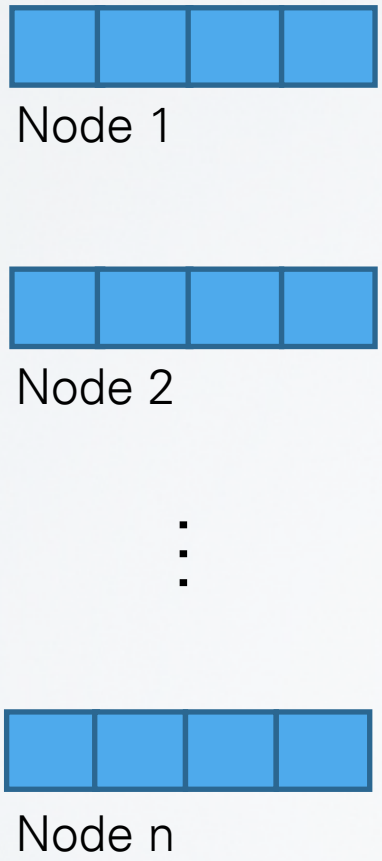


DECENTRALIZED





DECENTRALIZED



When

M: load difference discovered
anomaly discovered
anticipated

Where

M: memory, cycles, comm
consider topology
application knowledge

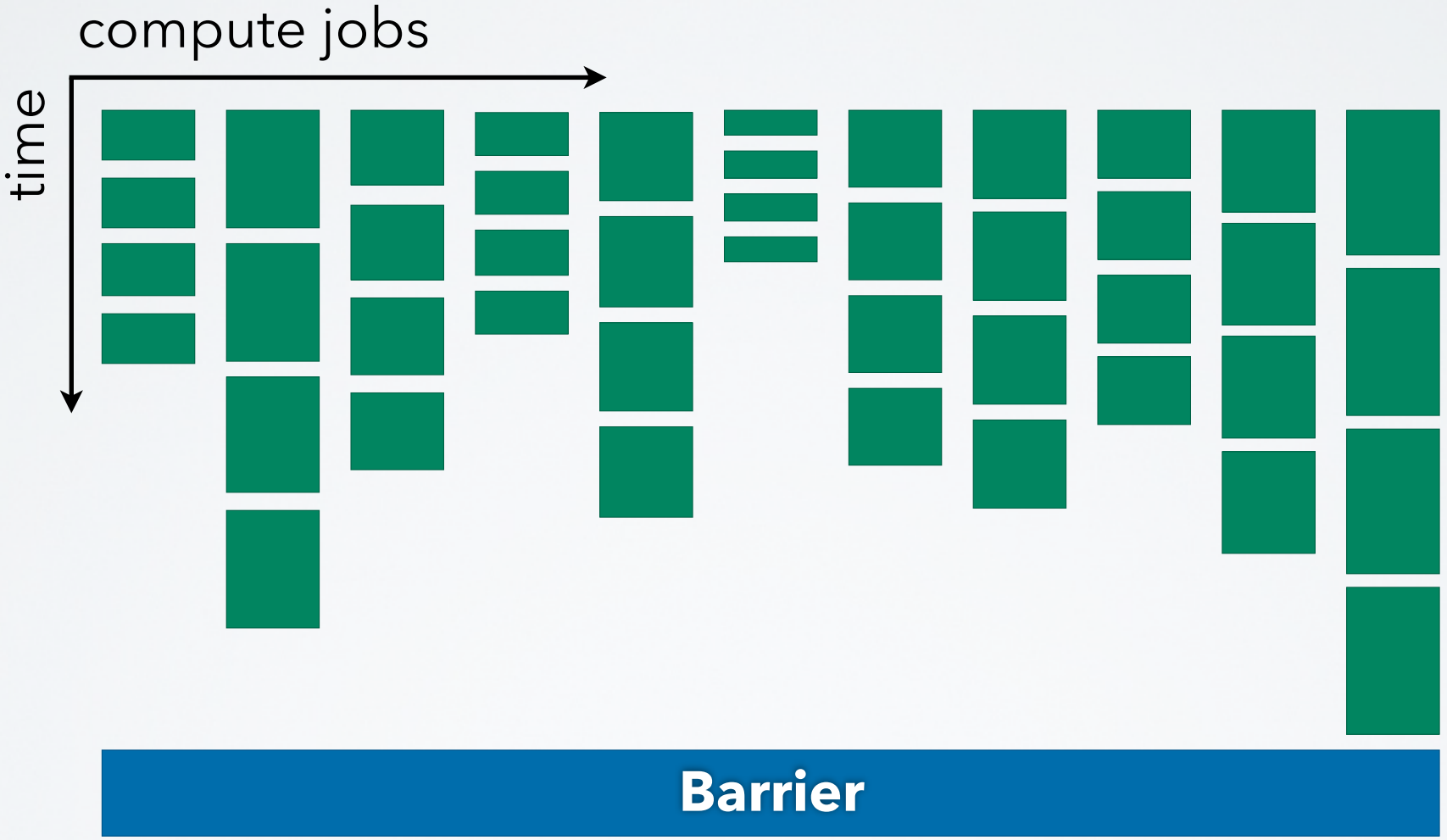
Which

M: past predicts future
application knowledge

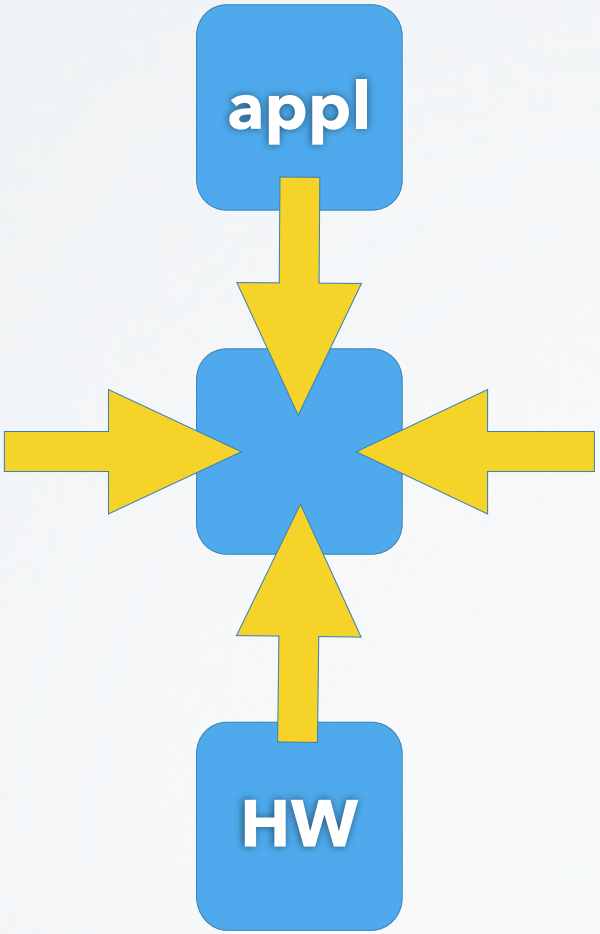
CHALLENGES

- gossip scalability
  -> talk by Carsten Weinhold
- decentralized topology-aware allocation
- impact of migration on communication
- what is load ?
- how to “yell for help” ?

SCHEDULING



SCHEDULING



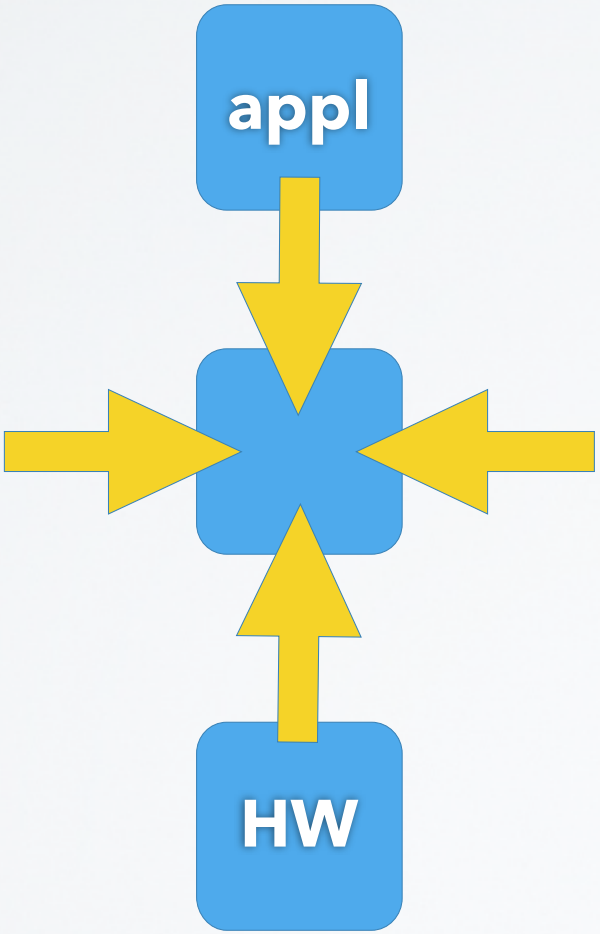
past predicts future

gossip

event counters

clocks

SCHEDULING



applications predict future

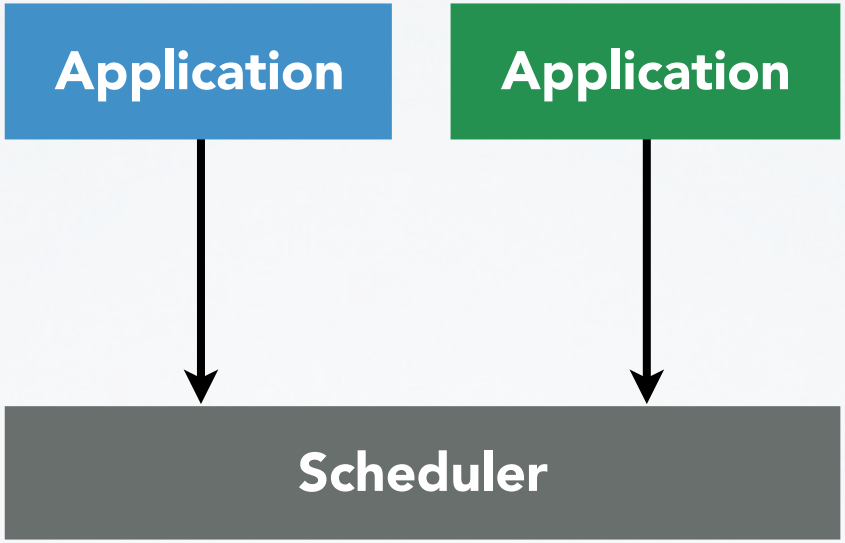
cloud density

particle density

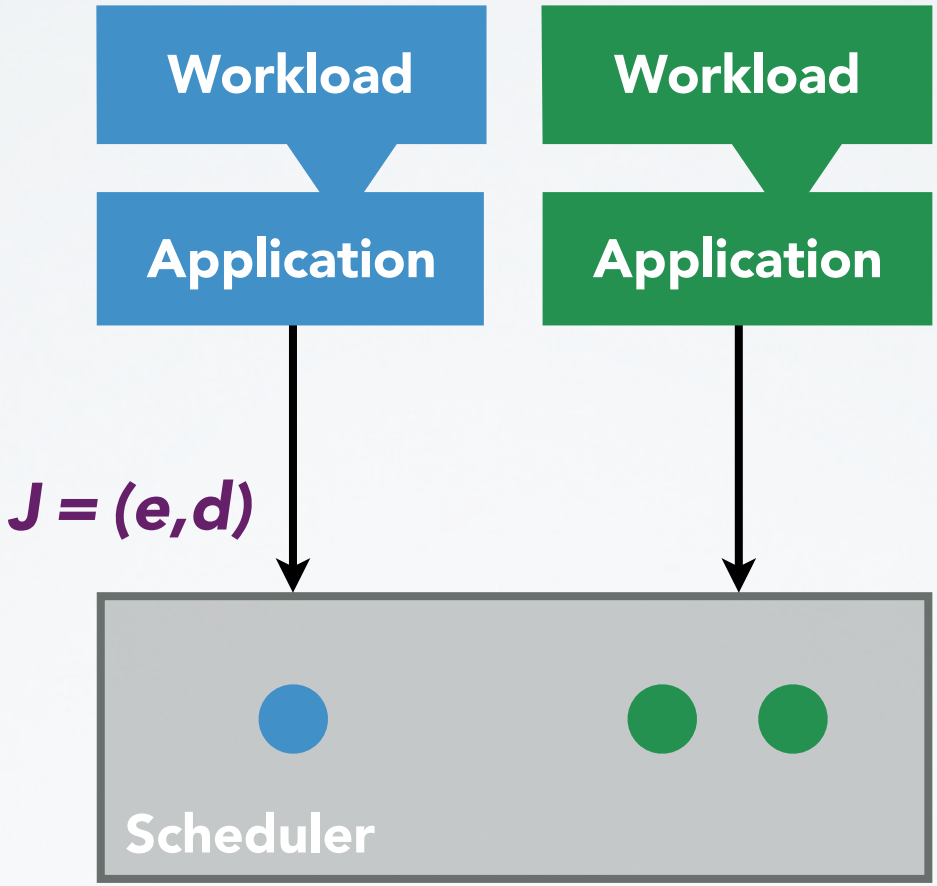
activity centres

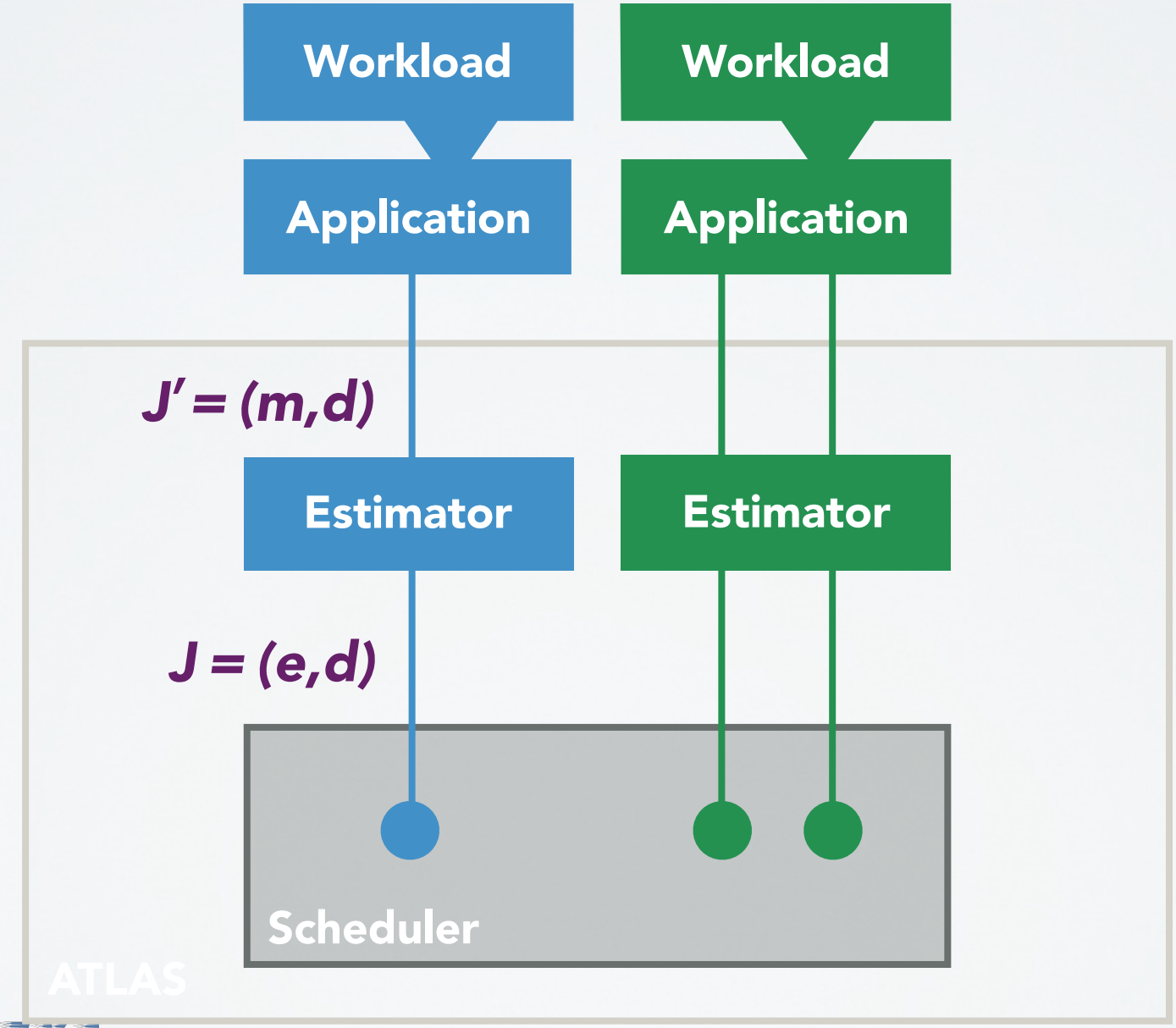
....

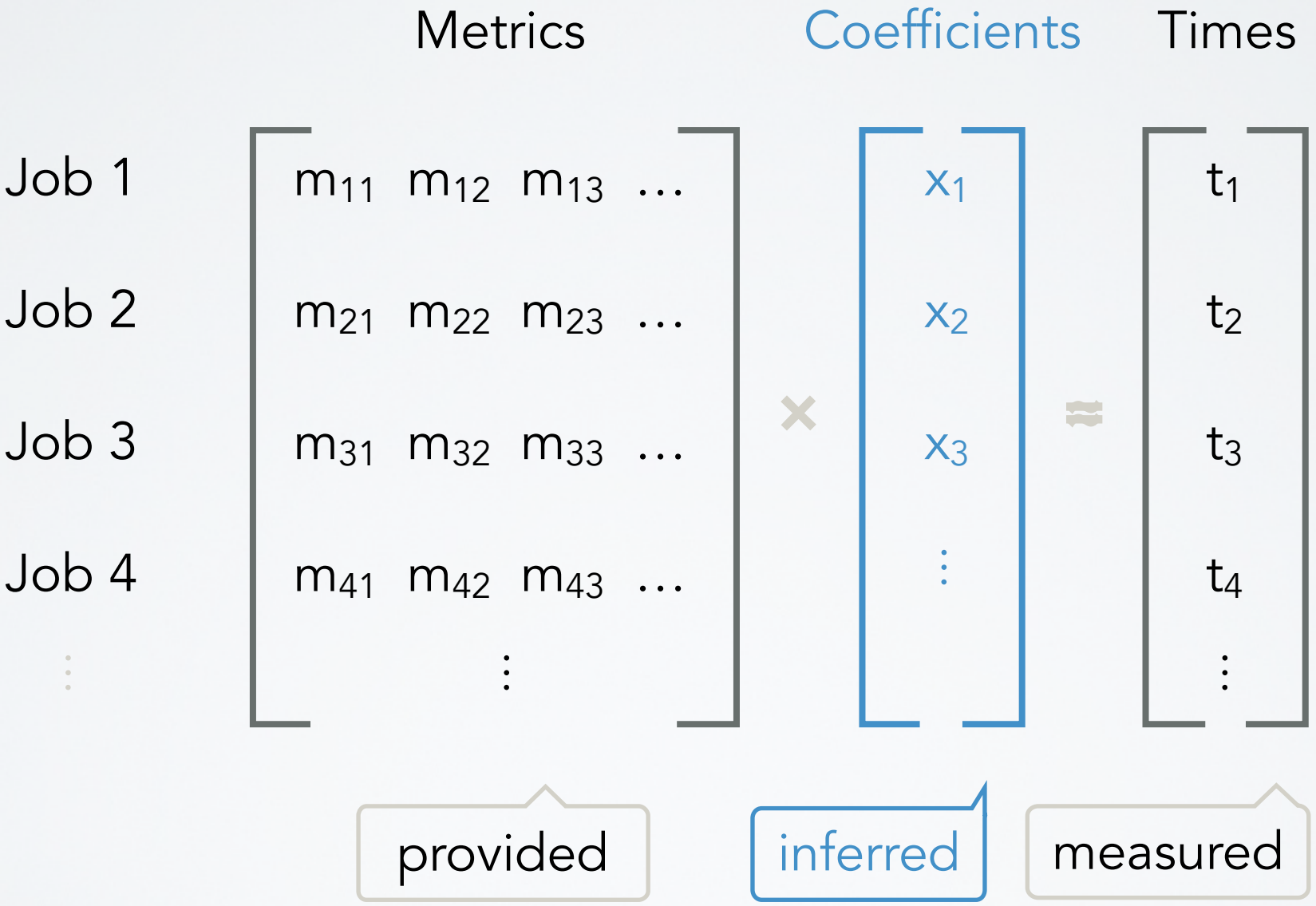
REAL-TIME



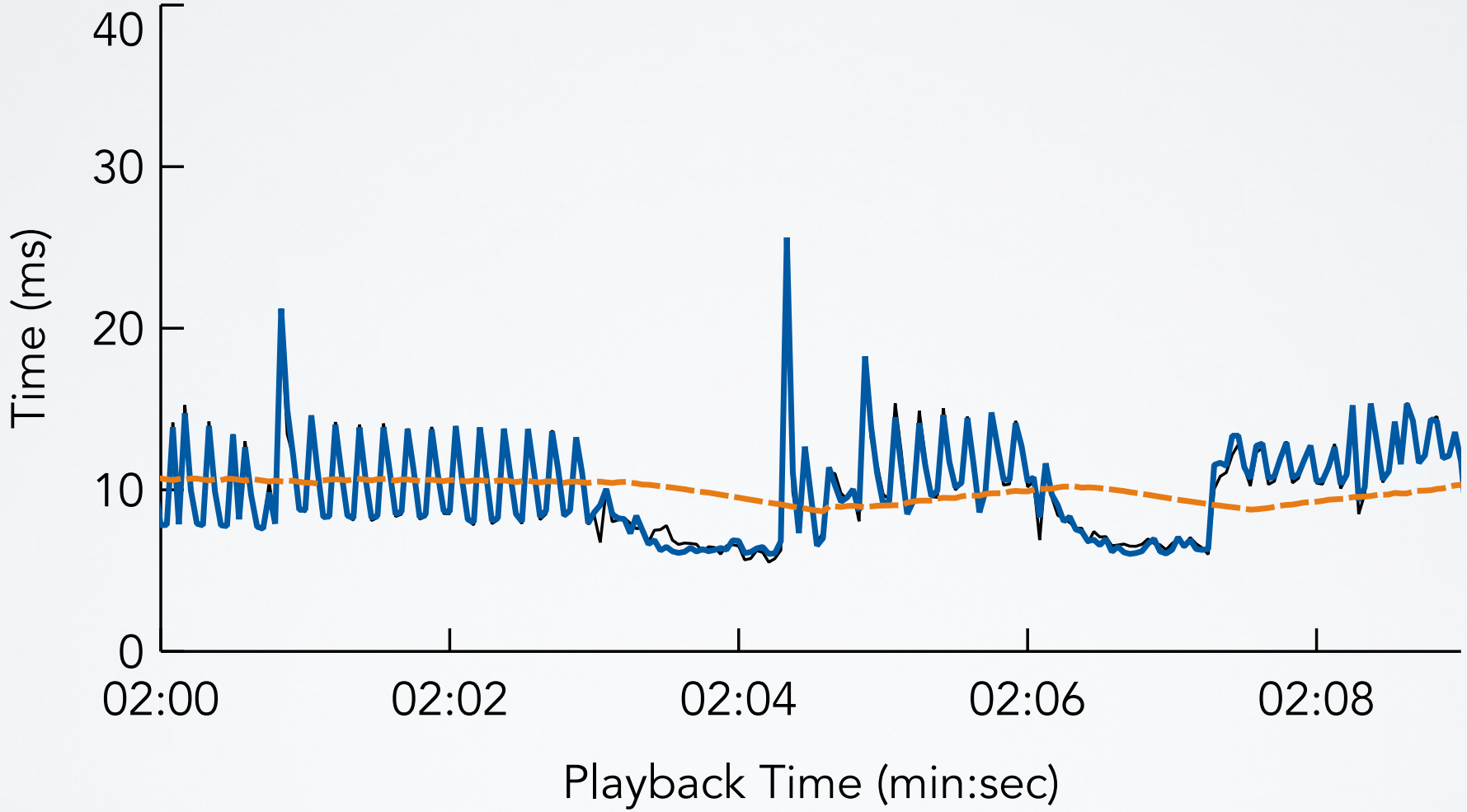
Michael Roitzsch, Practical Real-Time with Look-Ahead Scheduling
PhD Dissertation 2013


Workload
Workload
Application
Application
J = (e,d)
Scheduler


Workload
Workload
Application
Application
J' = (m,d)
Estimator
Estimator
J = (e,d)
Scheduler
ATLAS


Metrics
Coefficients
Times
Job 1
Job 2
Job 3
Job 4
m_{11} m_{12} m_{13} …
m_{21} m_{22} m_{23} …
m_{31} m_{32} m_{33} …
m_{41} m_{42} m_{43} …
x_1
x_2
x_3
t_1
t_2
t_3
t_4
provided
inferred
measured

VIDEO


40
30
20
10
0
Time (ms)
02:00
02:02
02:04
02:06
02:08
Playback Time (min:sec)

CHALLENGE

cloud density

particle density

acticity centres

....

cache misses

cpu cycles

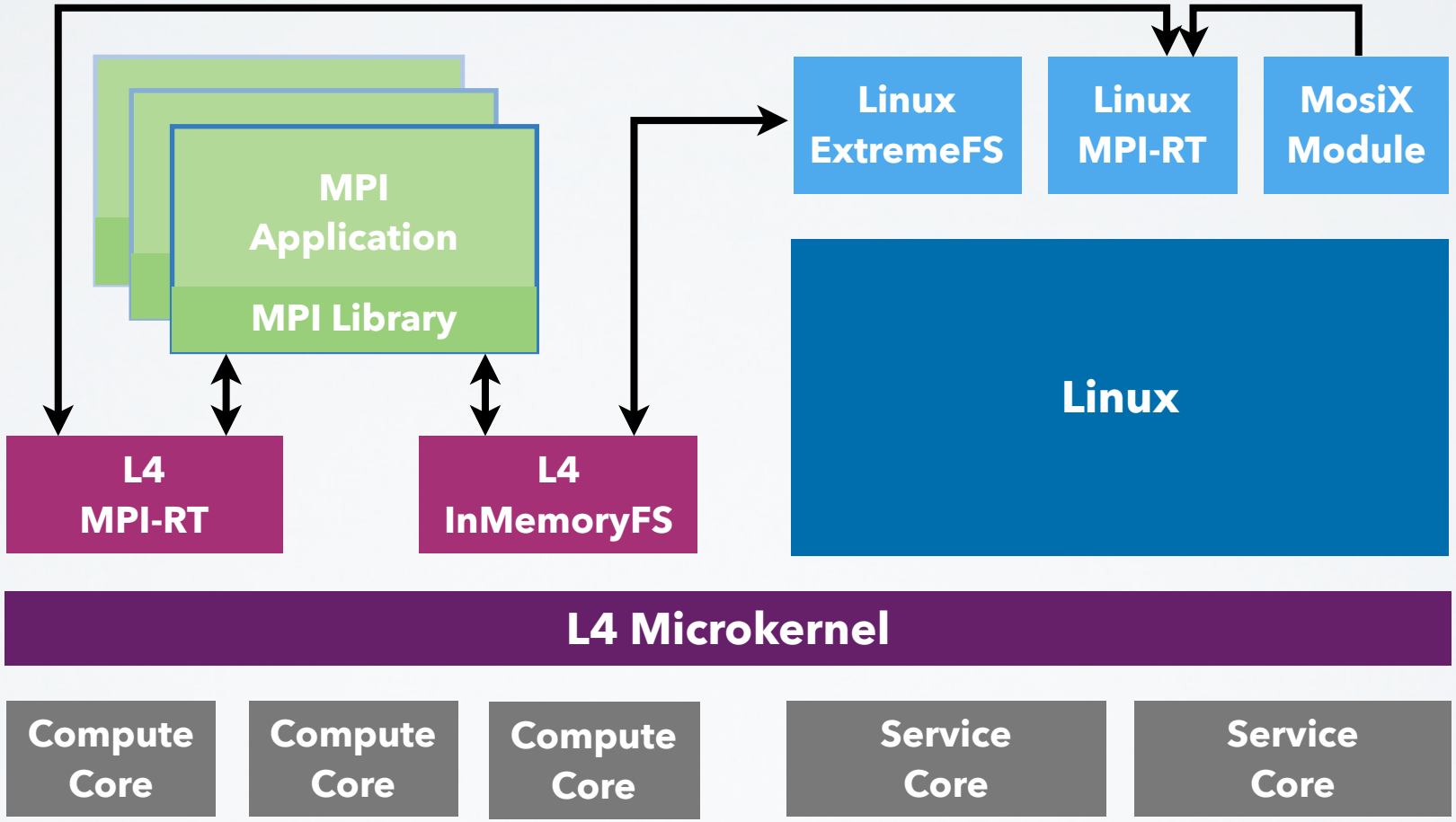
network traffic

....

power

Summary

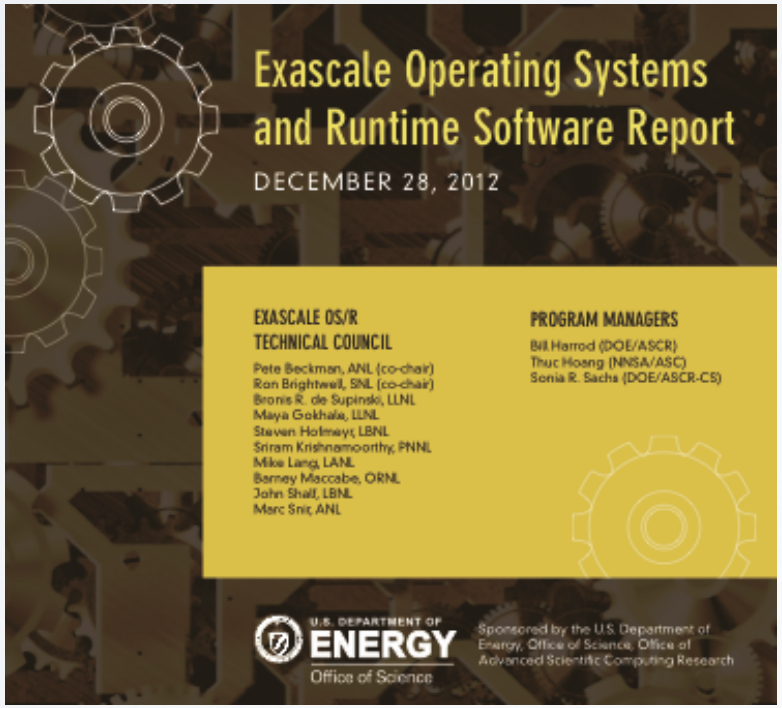
FFMK PROTOTYPE



CHALLENGES

- online decentralized topology- and energy-aware allocation
- programming models interaction with OS
- migration impact on network communication

SUMMARY



Fundamental
OS Building Blocks