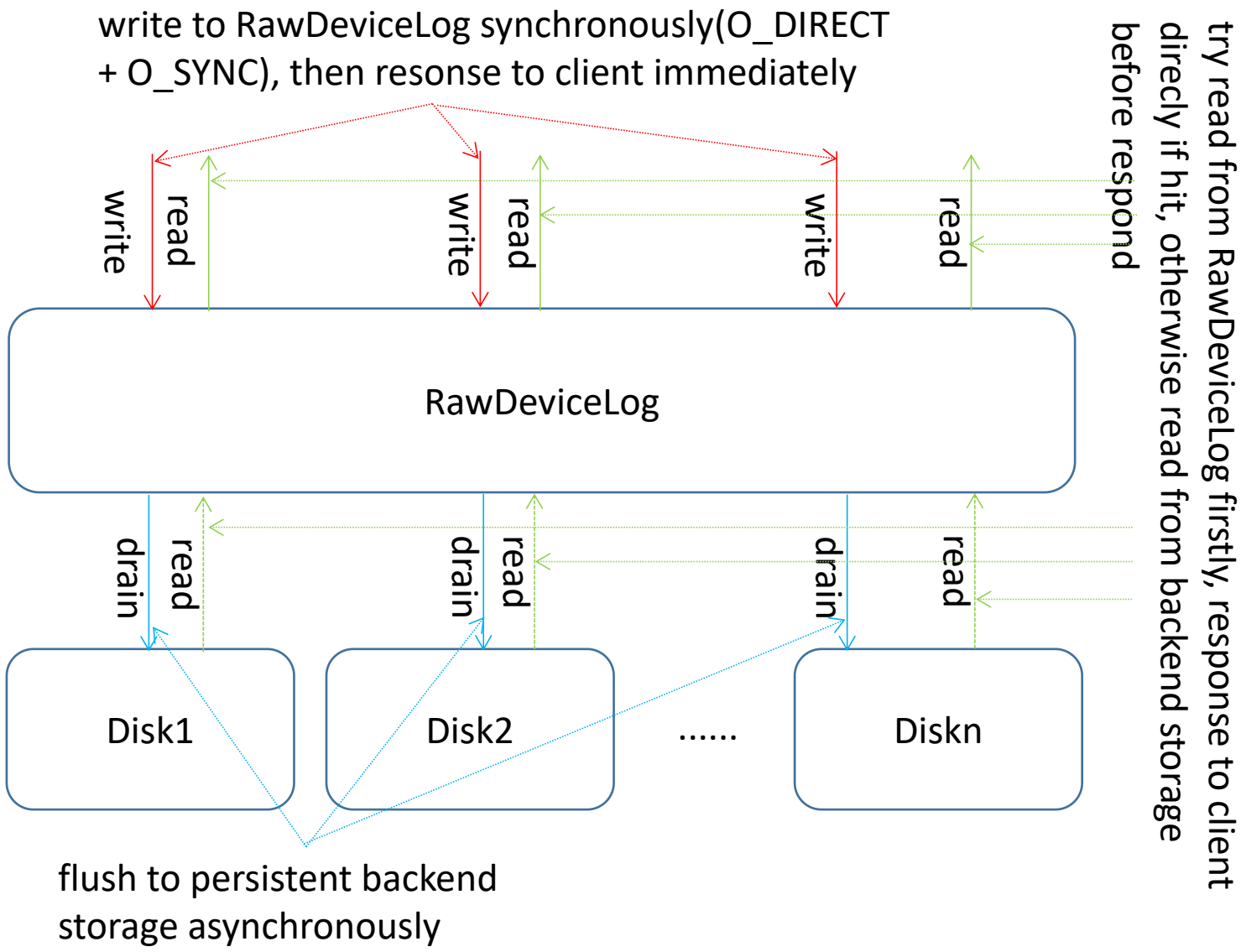
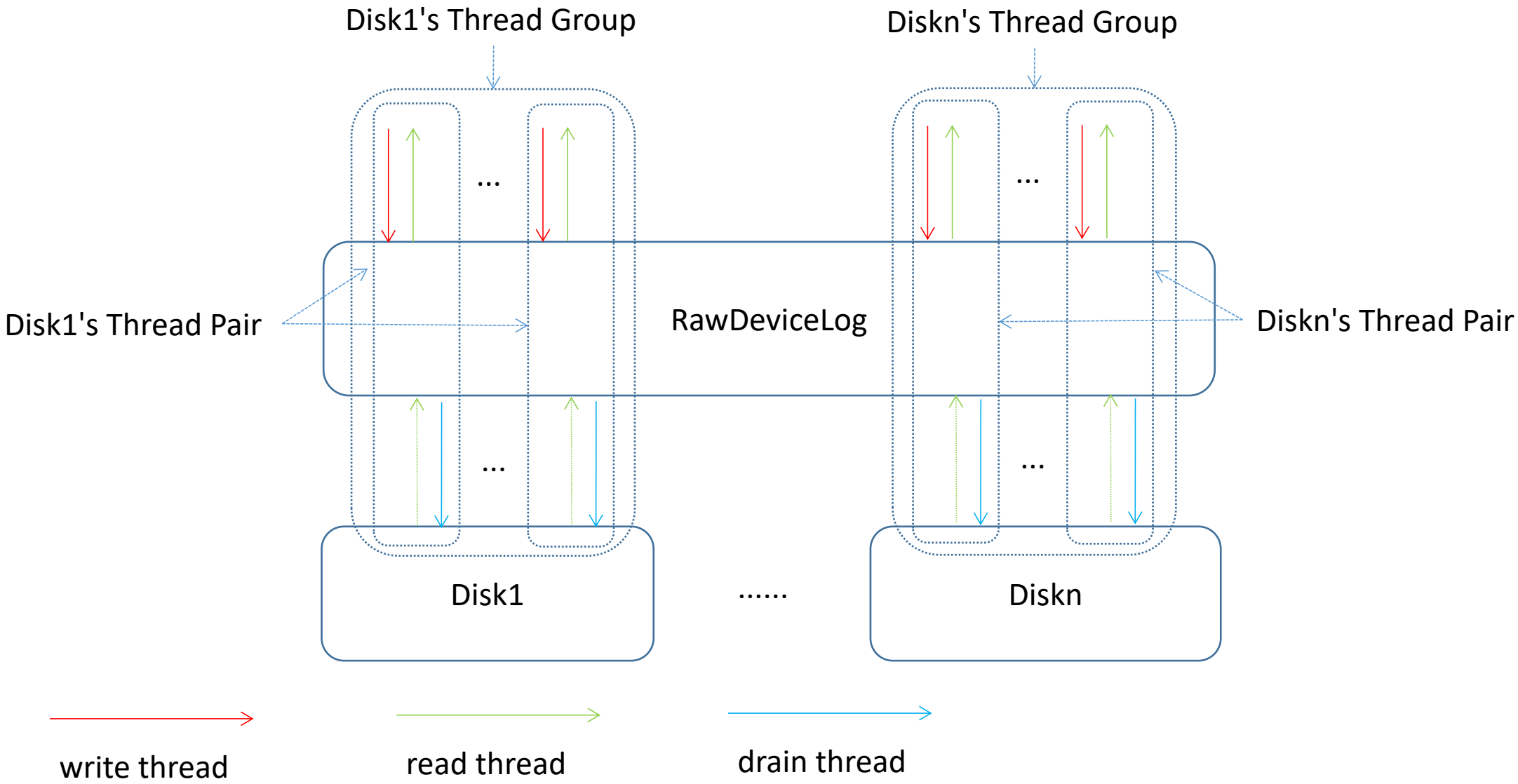
LogFs Design

xiaofei

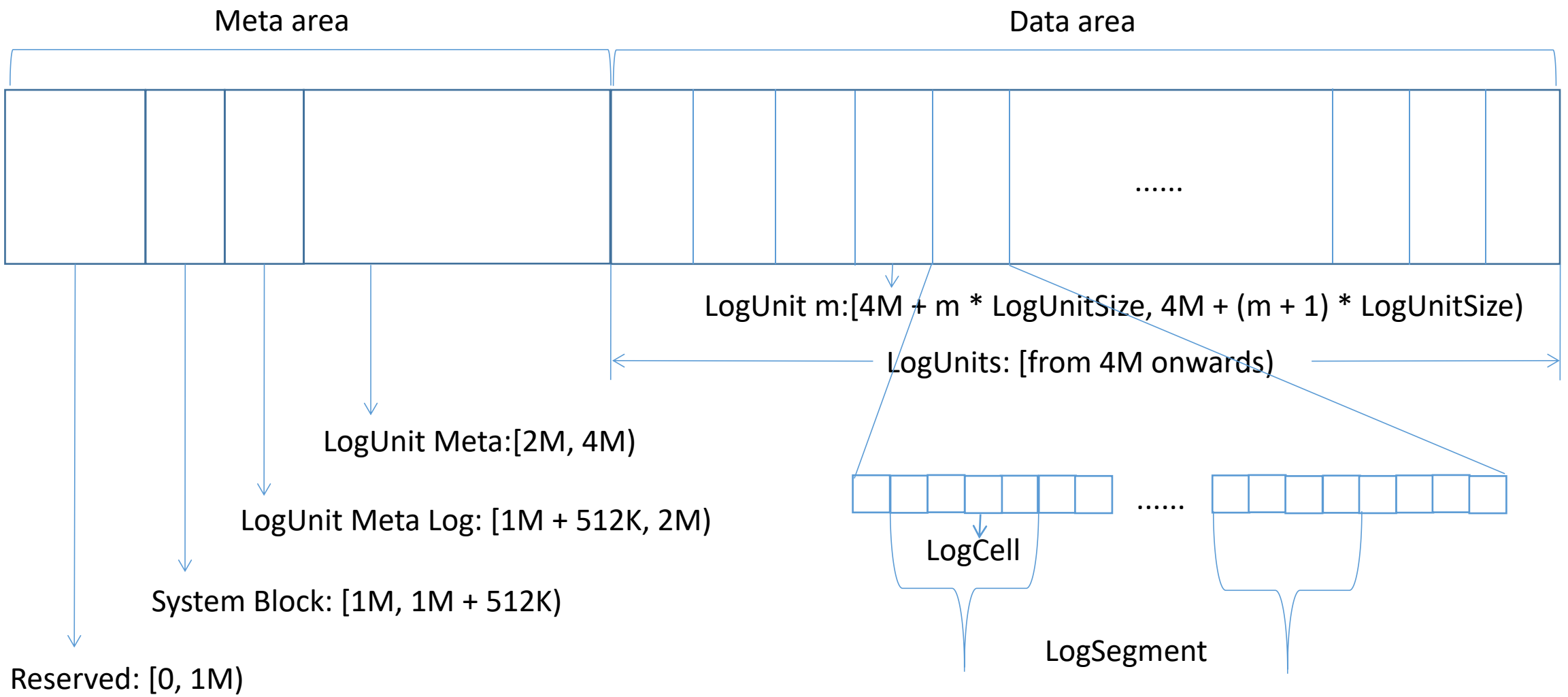
RawDeviceLog Logical View



Thread Model



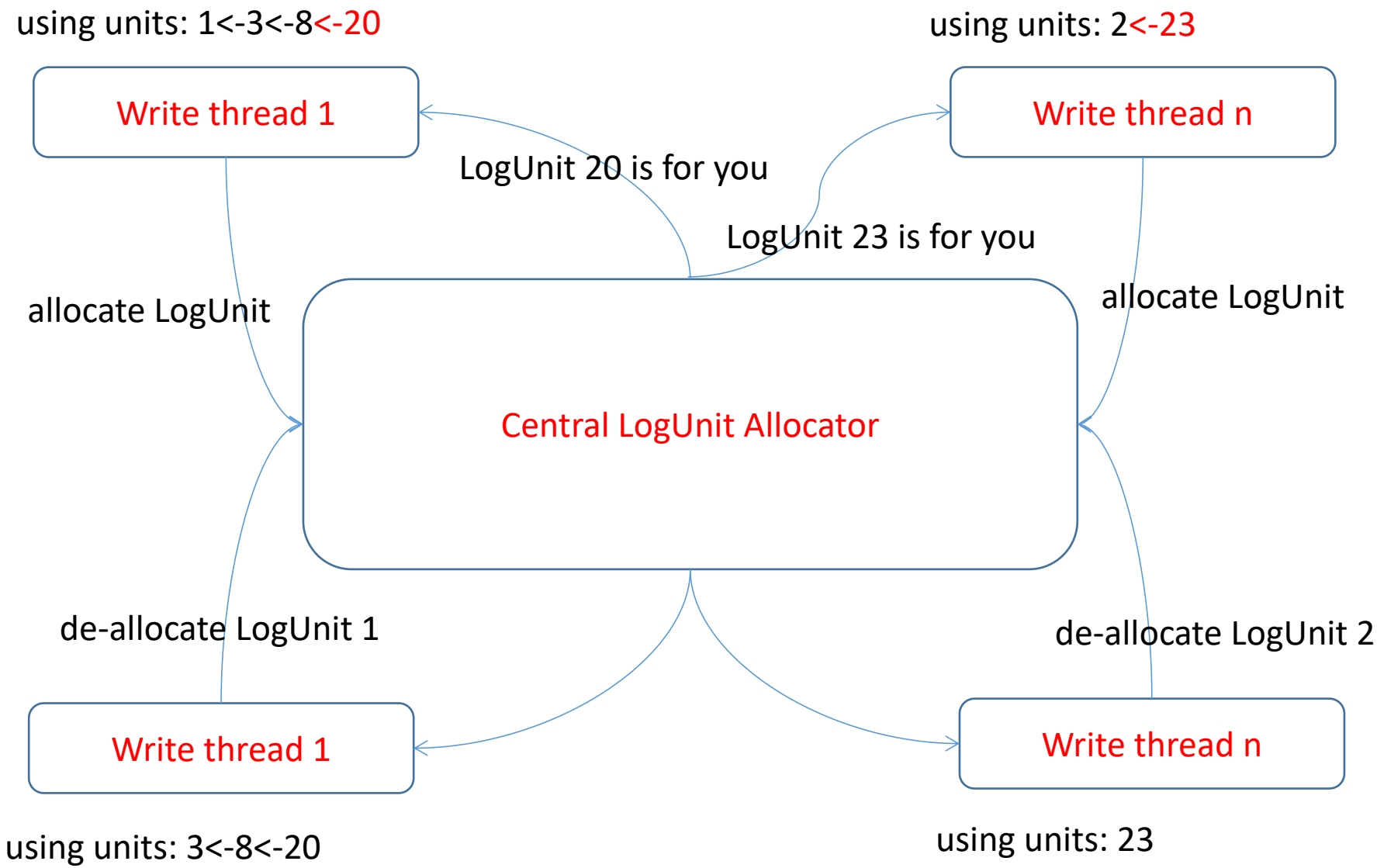
Disk Layout


Meta area
Data area
.....
LogUnit m:[4M + m * LogUnitSize, 4M + (m + 1) * LogUnitSize)
LogUnits: [from 4M onwards)
LogUnit Meta:[2M, 4M)
LogUnit Meta Log: [1M + 512K, 2M)
System Block: [1M, 1M + 512K)
Reserved: [0, 1M)
.....
LogCell
LogSegment

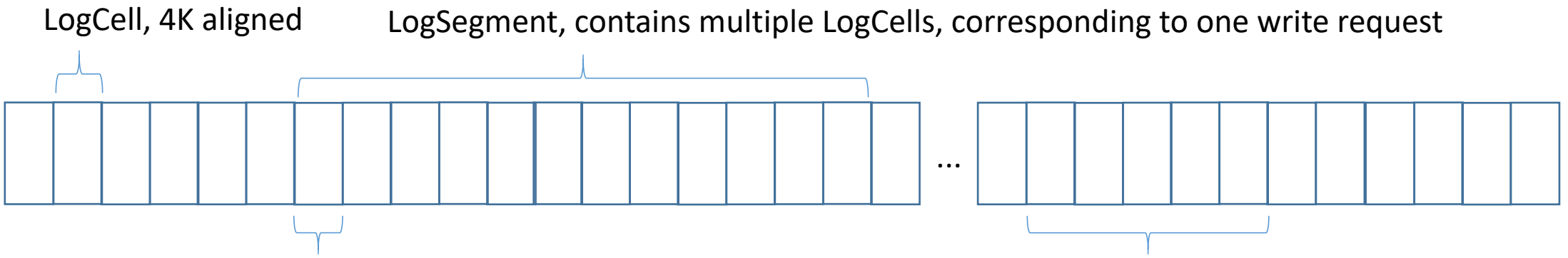
Disk Layout(Cont 1)

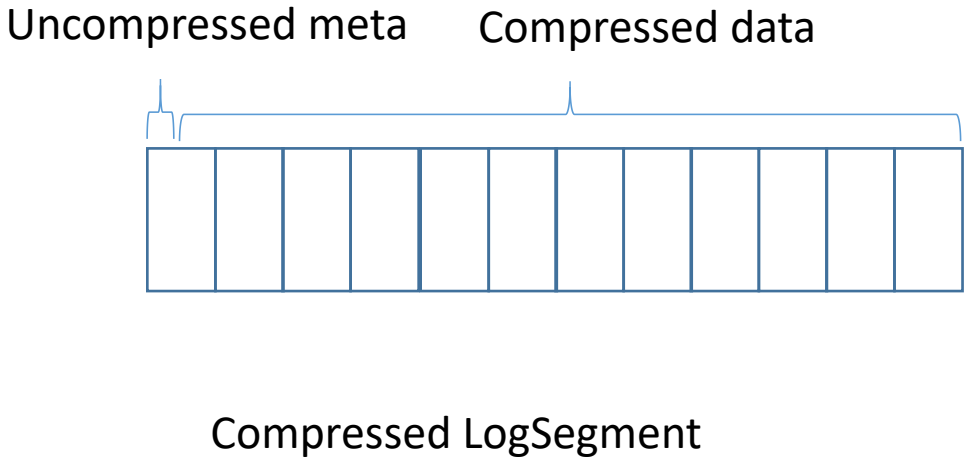
- System Block
  - systemid, deviceid, magic ...
- LogUnit Meta Log
  - act as write buffer of LogUnit Meta update
- LogUnit Meta
  - LogUnits' allocation/deallocation
  - LogUnits' ownership
  - LogUnits' time serial view
- LogUnit Data
  - composed of multiple LogUnits
  - all LogUnits are same in size
  - one LogUnit is exclusively accessed by one backend disk before its ownership is released
  - LogUnit's allocation is controlled by central UnitAllocator

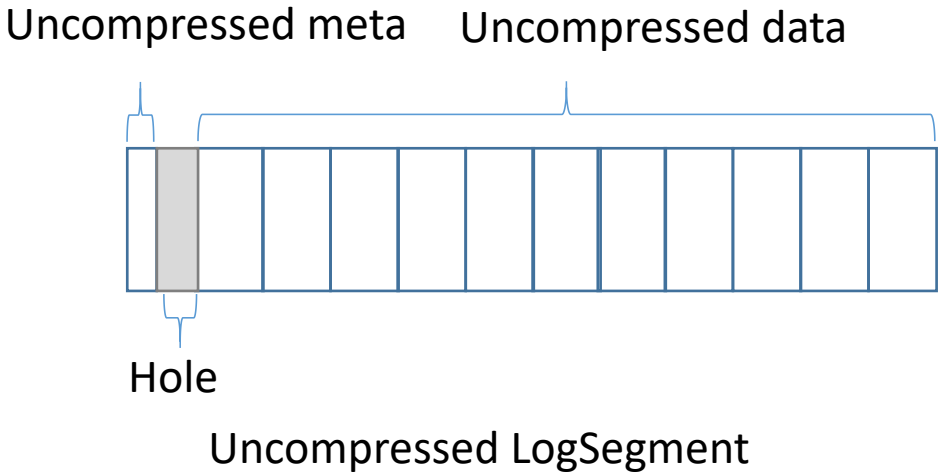
LogUnit Allocation



LogUnit

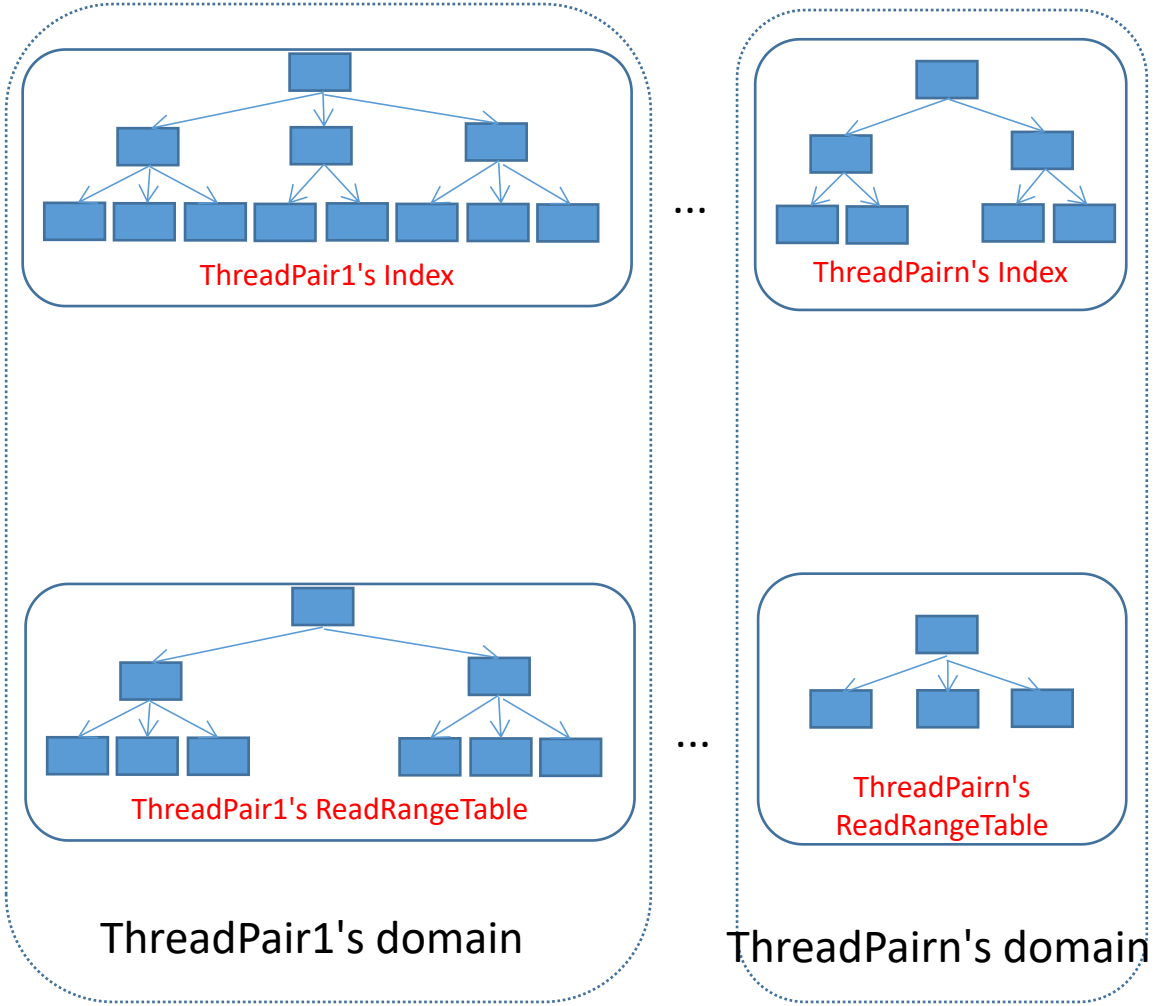

LogCell, 4K aligned
LogSegment, contains multiple LogCells, corresponding to one write request
...
Header LogCell of
this LogSegment
LogSegment's size is variable


Uncompressed meta
Compressed data
Compressed LogSegment


Uncompressed meta
Uncompressed data
Hole
Uncompressed LogSegment

RawDeviceLog Index Management

- **RawDeviceLog index is a memory based b* tree**
- **RawDeviceLog index is managed by thread pair**
- **Write/read/drain thread share the same RawDeviceLog index view managed by its owner thread pair**
- **Concurrent access to RawDeviceLog index is protected by RWLock**
- **ReadRangeTable is adopted for accelerating read**
  - record how much LogCells to be read(so that data can be correctly de-compressed)
  - managed similary to RawDeviceLog index



RawDeviceLog index management - case study

- Write {inode: 10, chunkidx: 3, startPage: 6, endPage: 8} @ {LogUnit: 23, LogCell: 105}, consume 3 LogCells totally
  - update {key: {10, 3, 5}, value: {23, 5}} in RawDeviceLog index
  - update {key: {10, 3, 6}, value: {23, 5}} in RawDeviceLog index
  - update {key: {10, 3, 7}, value: {23, 5}} in RawDeviceLog index
  - update {key: {10, 3, 8}, value: {23, 5}} in RawDeviceLog index
  - update {key: {23, 5}, value: 3} in ReadRangeTable
- Read {inode: 10, chunkidx: 3, startPage: 6, endPage: 7}
  - find {key: {10, 3, 6}, value: {23, 5}} from RawDeviceLog index
  - find {key: {10, 3, 7}, value: {23, 5}} from RawDeviceLog index
  - find {key: {23, 5}, value: 3} from ReadRangeTable
- Delete {inode: 10, chunkidx: 3, startPage: 6, endPage: 7}
  - delete {key: {10, 3, 6}, value: {23, 5}} from RawDeviceLog index
  - delete {key: {10, 3, 7}, value: {23, 5}} from RawDeviceLog index

Main Logic

- Write logic
  - pls refer to “logfs logic.pdf”
- Read logic
  - pls refer to “logfs logic.pdf”
- Drain/Redo logic
  - each ThreadPair has its own drain/redo thread
  - drain LogUnit one by one in allocation order
  - pls refer to “logfs logic.pdf”
- Recover logic
  - pls refer to “logfs logic.pdf”