CarrotSQL

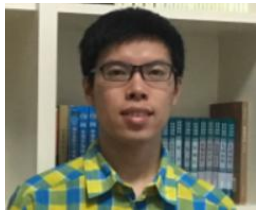
https://github.com/huangwentao0831/carrot

Team

Han Han

Guan Hua

Han Xueran



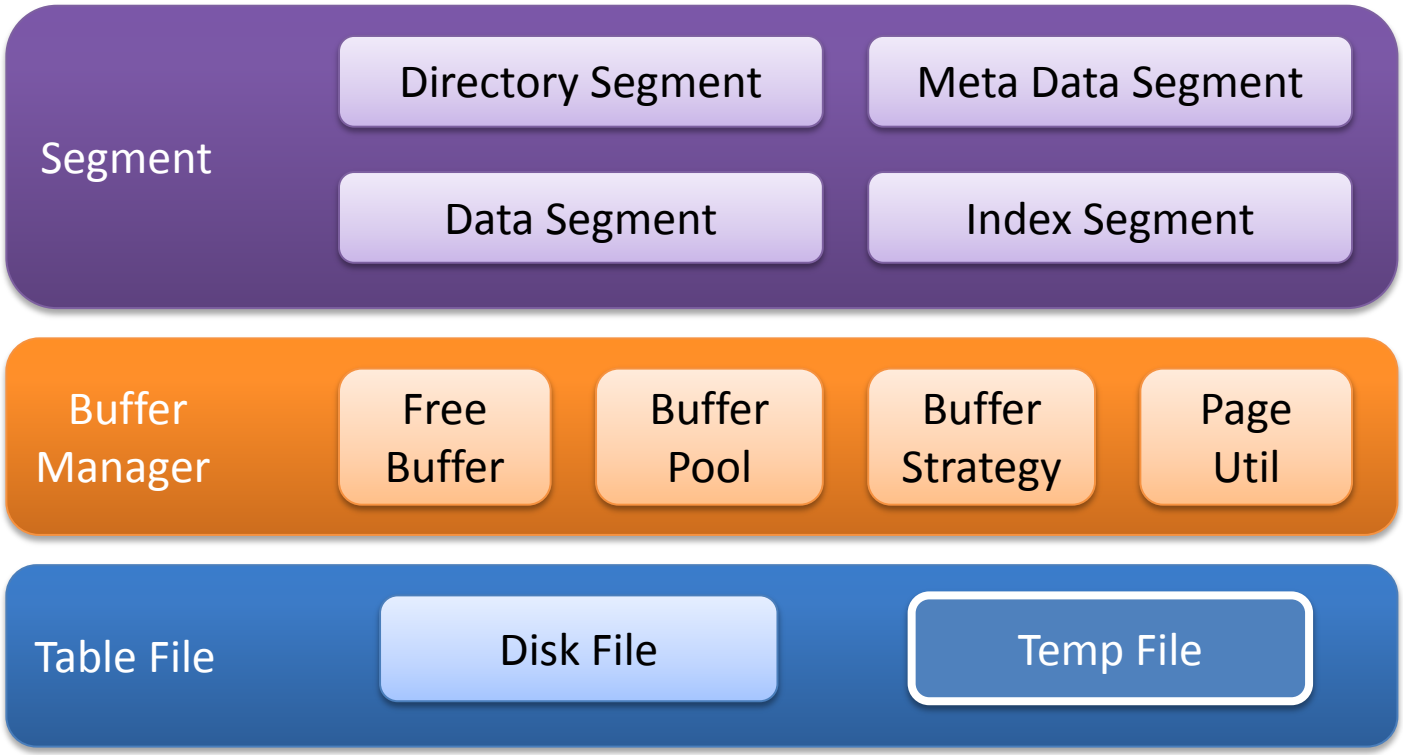
Huang Wentao

Shao Mingrui

Content

- Storage Management
- Index
- SPJ
- The Execution Demo

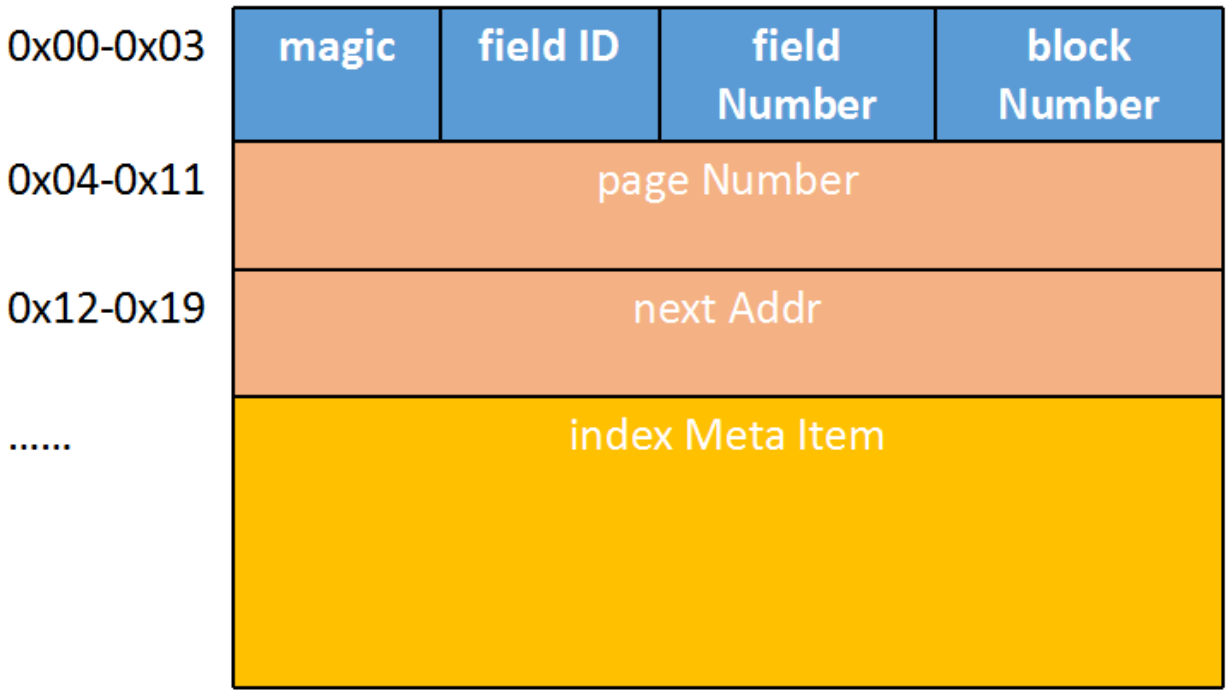
Storage Management



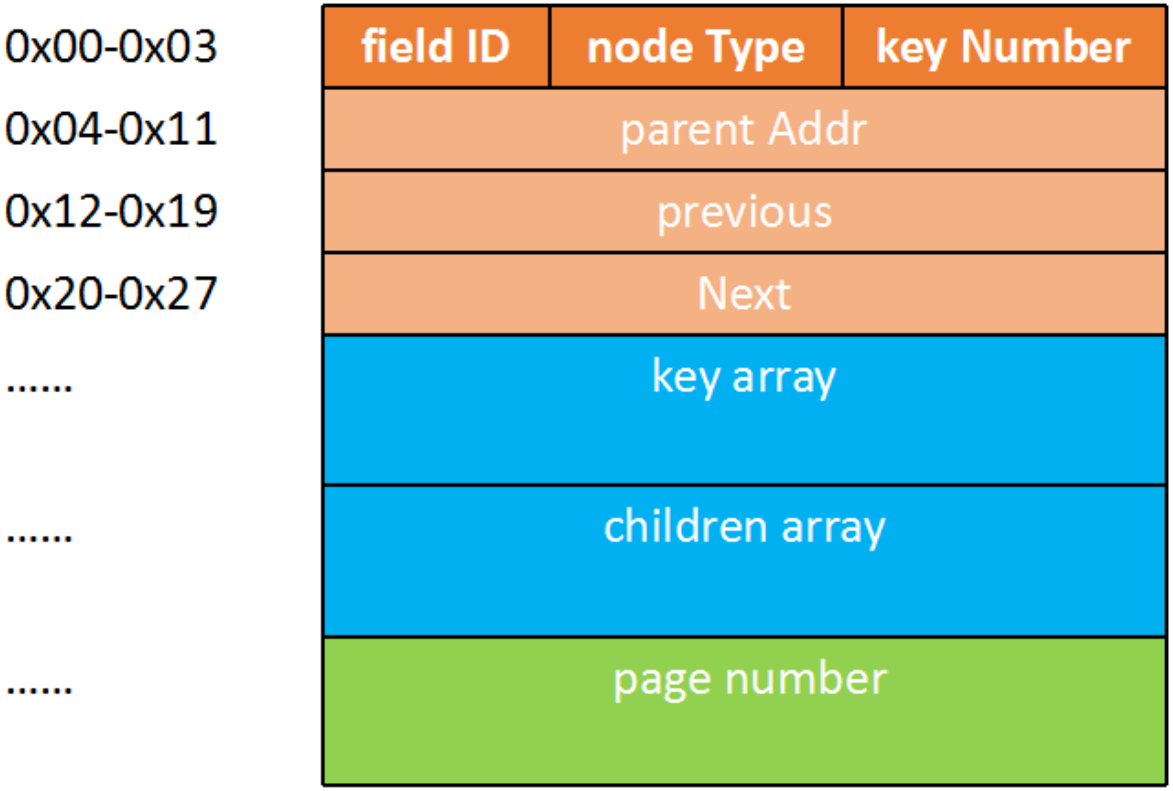
Content

- Storage Management
- Index
- SPJ
- The Execution Demo

Index


0x00-0x03
magic
field ID
field Number
block Number
0x04-0x11
page Number
0x12-0x19
next Addr
.....
index Meta Item

B+ tree Index

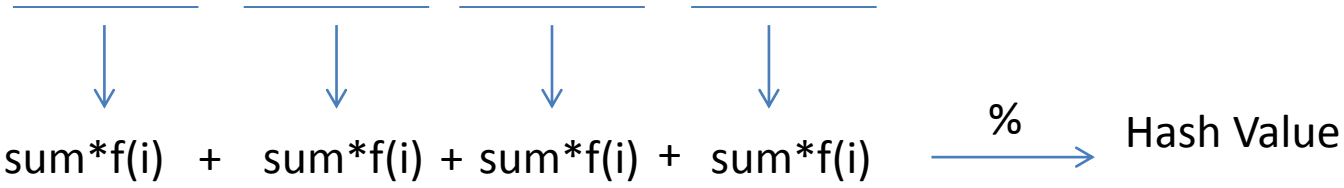

0x00-0x03
0x04-0x11
0x12-0x19
0x20-0x27
.....
.....
.....
field ID
node Type
key Number
parent Addr
previous
Next
key array
children array
page number

Linear Hash Index

- Expand the range of hash function

We give the factor for each sum of bytes, so that the range of hash function can expand from handreds to billions.

.... 0010 1001 0110 0001



$$\mathrm{sum} \times a^{i}, a = 3, i$$

- Extendible and shrinkable
  - The linear hash table can extend and shrink dynamically.

Linear Hash Index

- index structure

Bucket Size			Record Size
Split Rate	Key Size	Hash Size	Next Addr
Bucket Value ...			Bucket Addr ..

Linear Hash Index

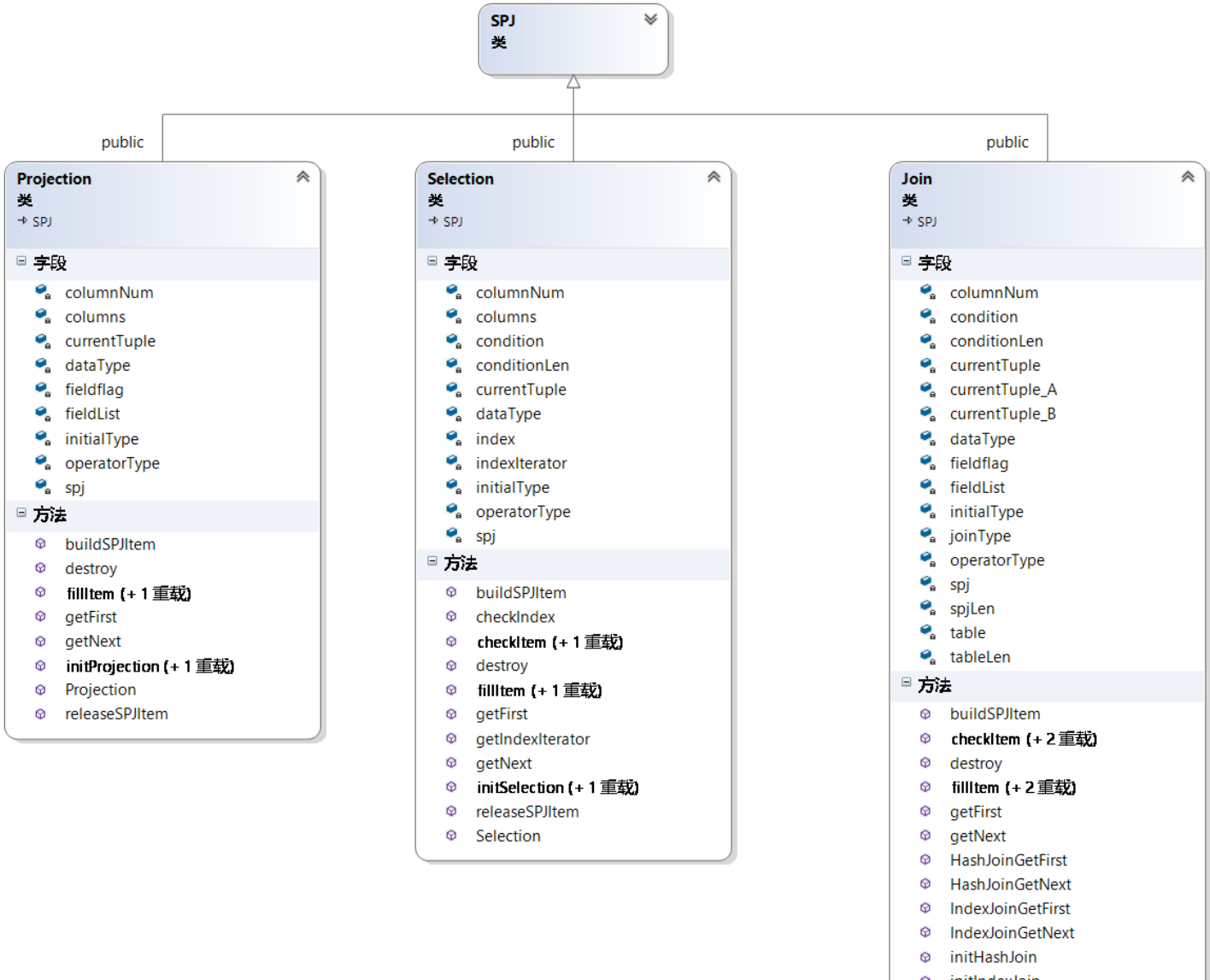
- bucket structure

Key num	Next Addr
Key value ...	Tuple Addr ..

Content

- Storage Management
- Index
- SPJ
- The Execution Demo

SPJ


SPJ
类
public
public
public
Projection
类
SPJ
字段
columnNum
columns
currentTuple
dataType
fieldflag
fieldList
initialType
operatorType
spj
方法
buildSPJItem
destroy
fillItem (+ 1 重载)
getFirst
getNext
initProjection (+ 1 重载)
Projection
releaseSPJItem
Selection
类
SPJ
字段
columnNum
columns
condition
conditionLen
currentTuple
dataType
index
indexIterator
initialType
operatorType
spj
方法
buildSPJItem
checkIndex
checkItem (+ 1 重载)
destroy
fillItem (+ 1 重载)
getFirst
getIndexIterator
getNext
initSelection (+ 1 重载)
releaseSPJItem
Selection
Join
类
SPJ
字段
columnNum
condition
conditionLen
currentTuple
currentTuple_A
currentTuple_B
dataType
fieldflag
fieldList
initialType
joinType
operatorType
spj
spjLen
table
tableLen
方法
buildSPJItem
checkItem (+ 2 重载)
destroy
fillItem (+ 2 重载)
getFirst
getNext
HashJoinGetFirst
HashJoinGetNext
IndexJoinGetFirst
IndexJoinGetNext
initHashJoin

SPJ

- Iterator

  iterator->getFirst(item);

  iterator->getNext(item);
- initial

  only table

  only iterator

  hybrid

SPJ

- Focus on Nest loop join and index join
  - Nest loop join based on tuple
  - Index join based on B+ tree and linear hash index
- Multiple tables join
  - Due to the join designed on iterators, so multiple tables join is very to realise and number of tables can be increased with no contrains.

Content

- Storage Management
- Index
- SPJ
- The Execution Demo

Thank you!

Conclusion

- Creation

  We focus on designing details, not perfect ,but better.

- Team Work

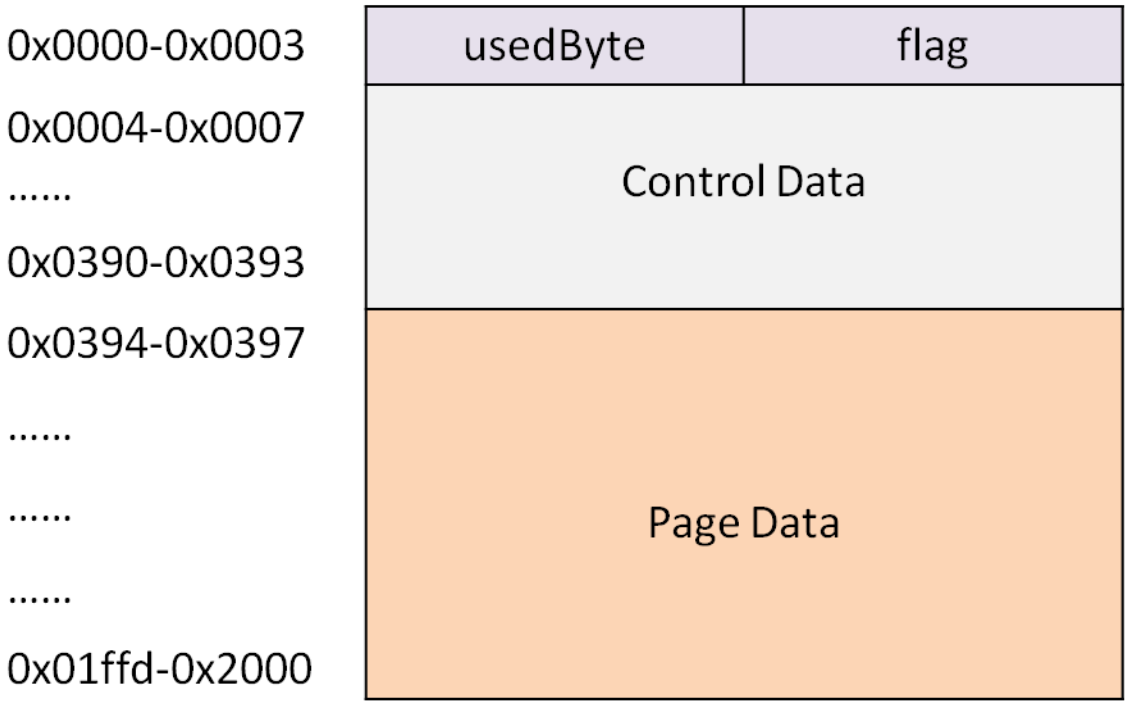
  One people ,One experiment ?

- Future work

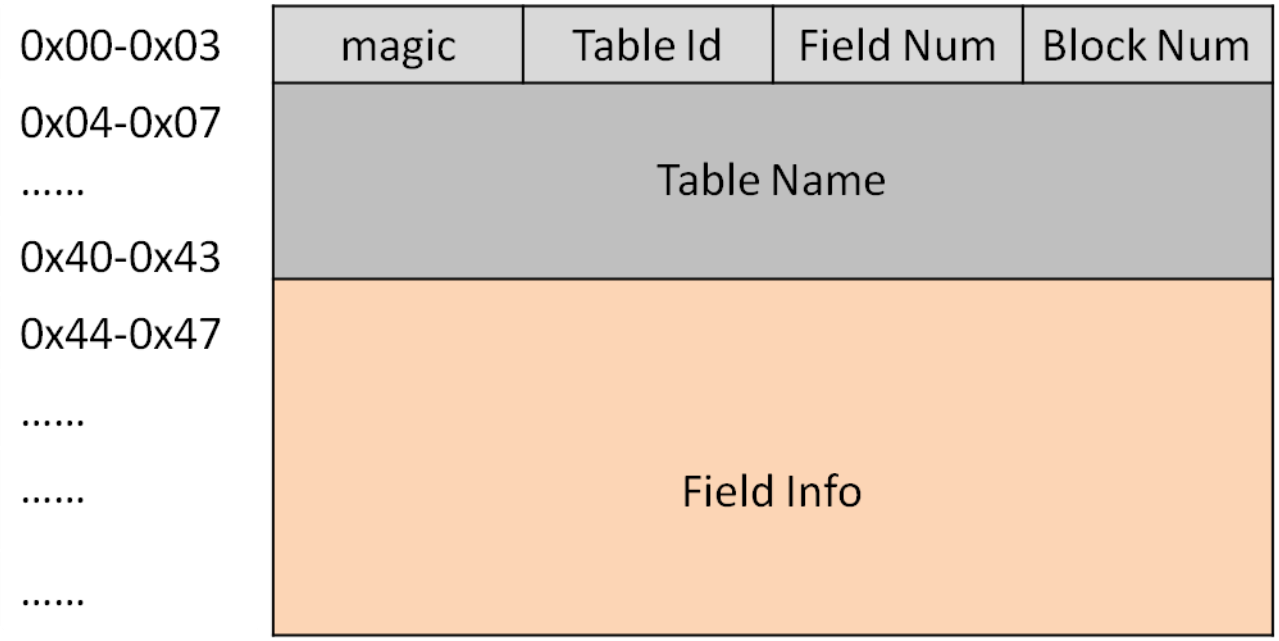
  Work harder, work better

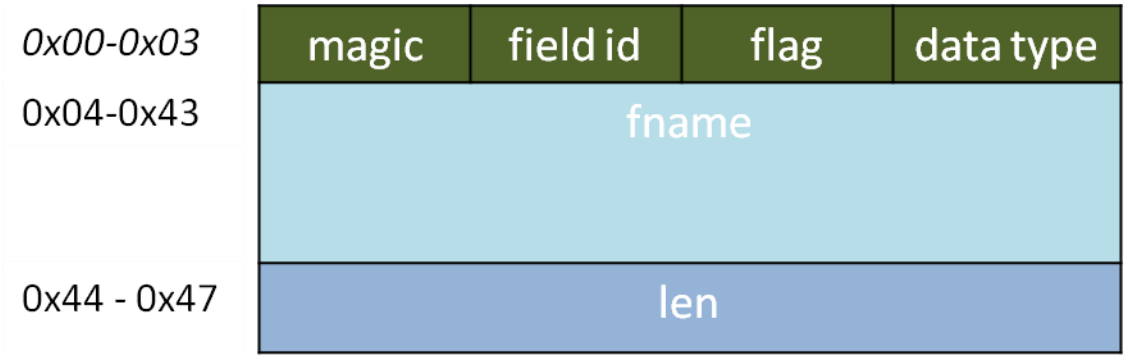
  Coding&Debug(Segment Fault)

Appendix: Page

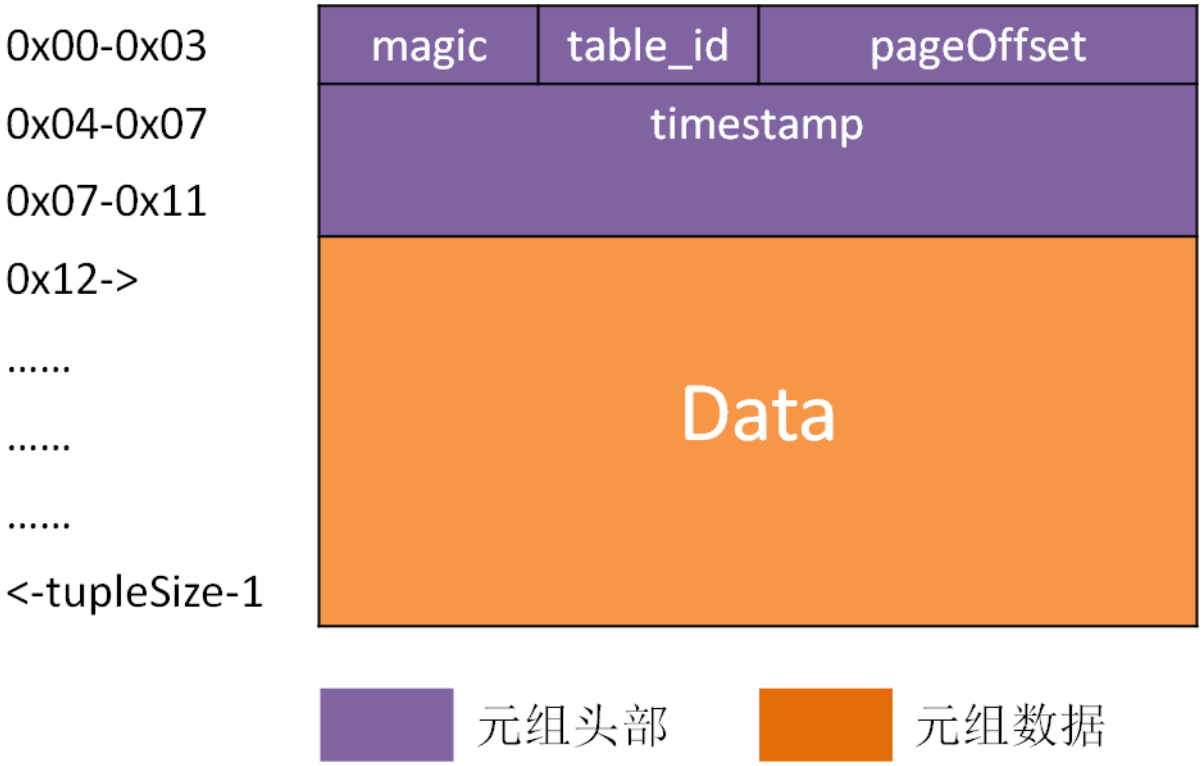


Appendix: Meta Data

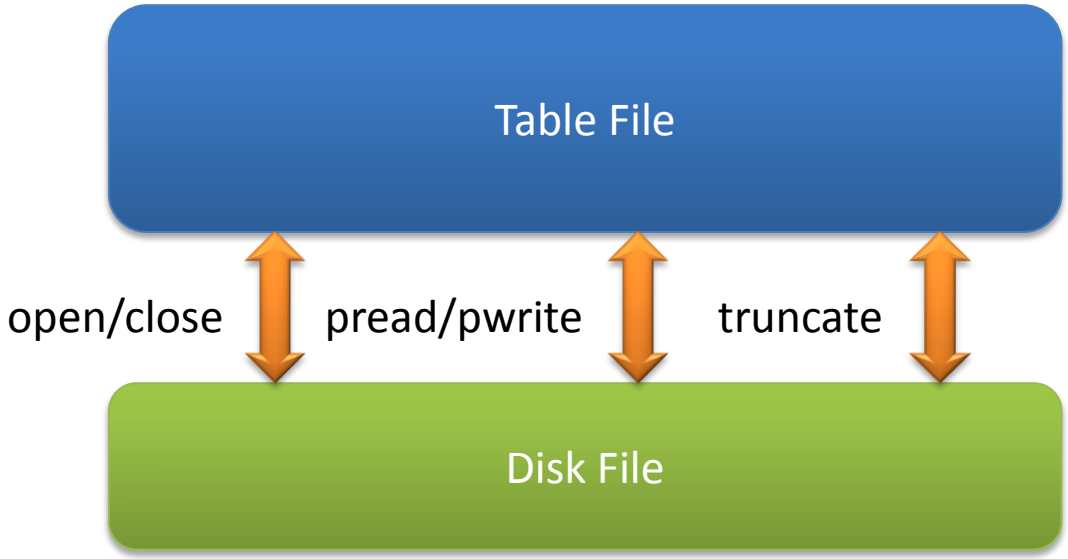




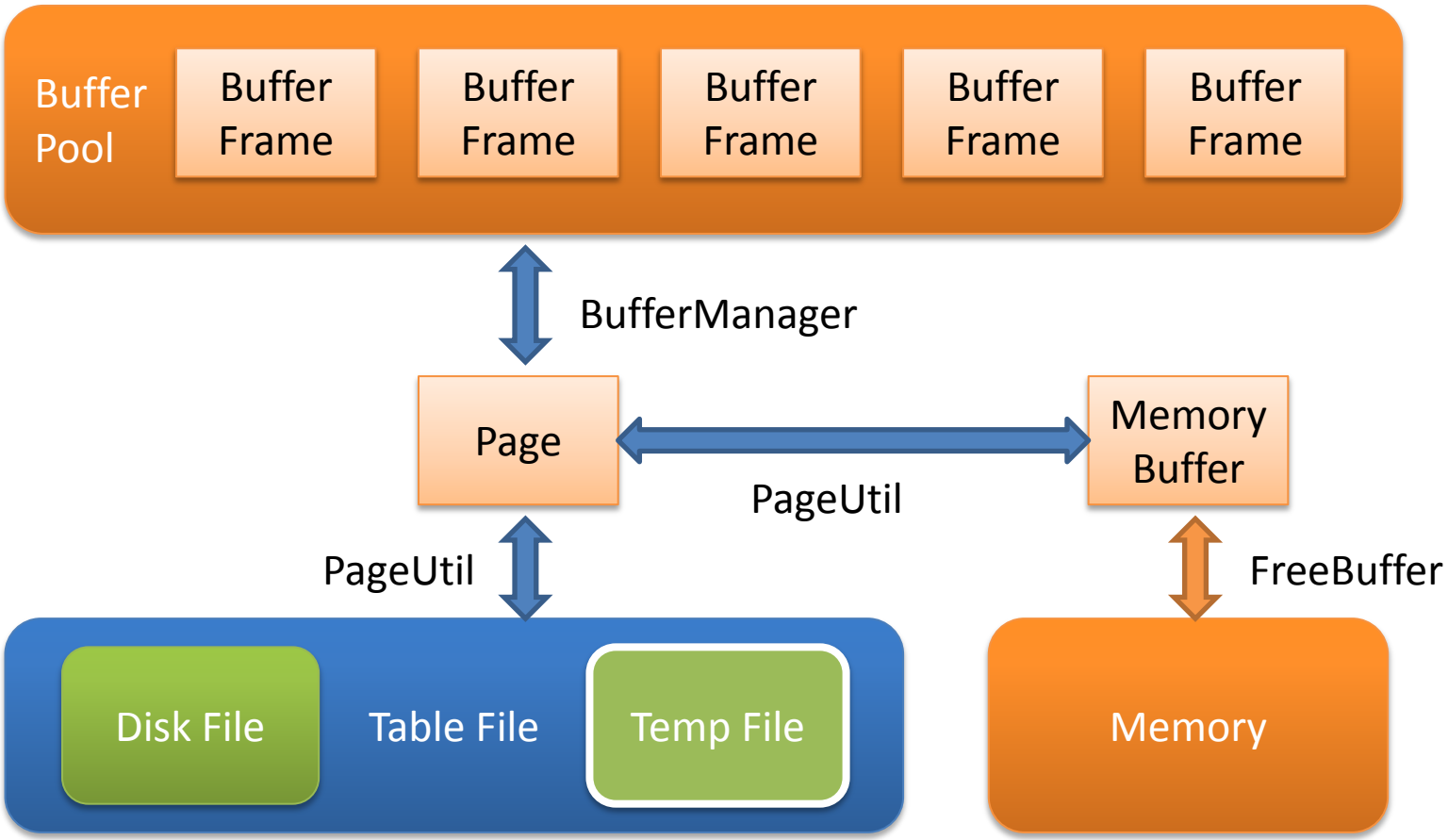
Appendix: Tuple



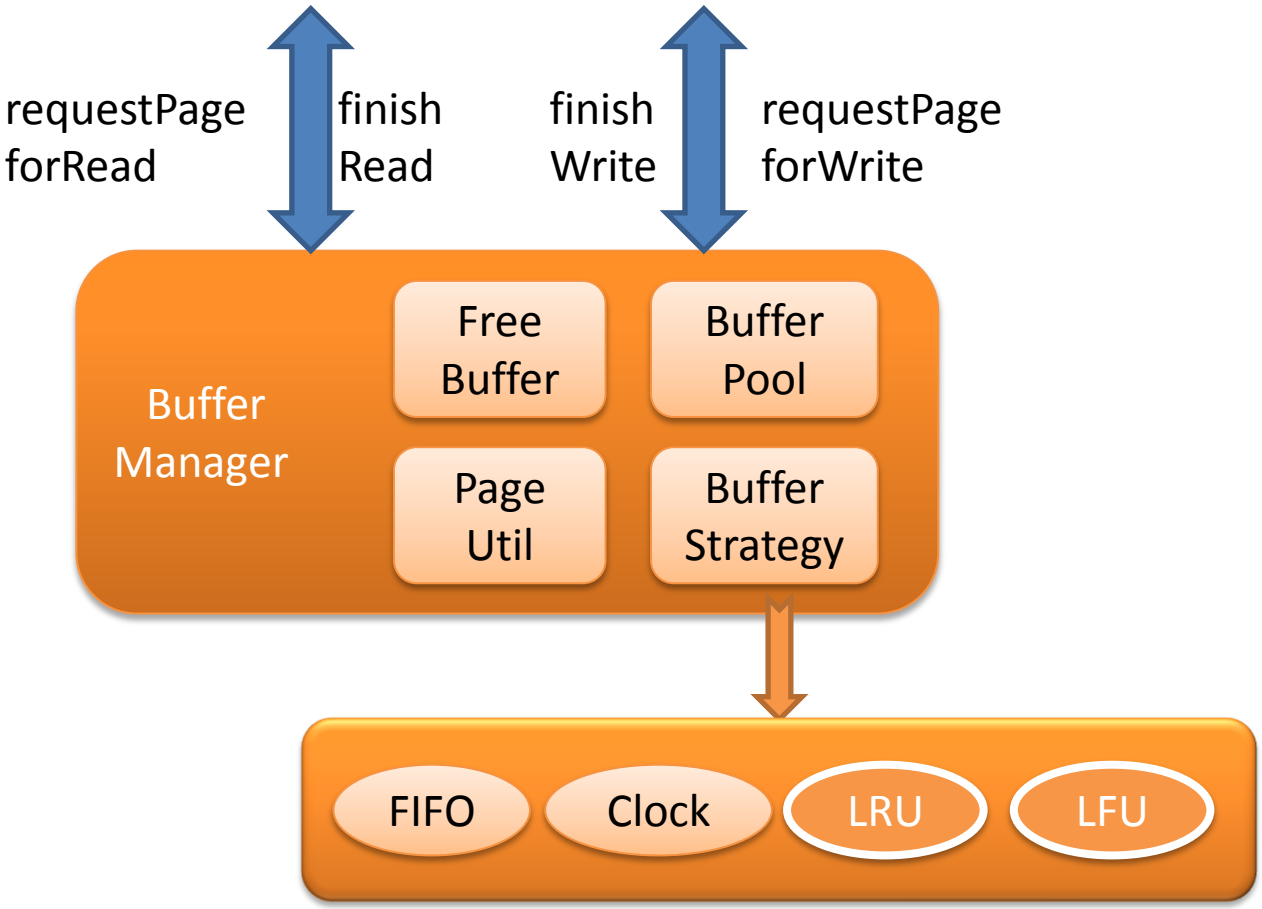
Storage: IO



Storage: Buffer



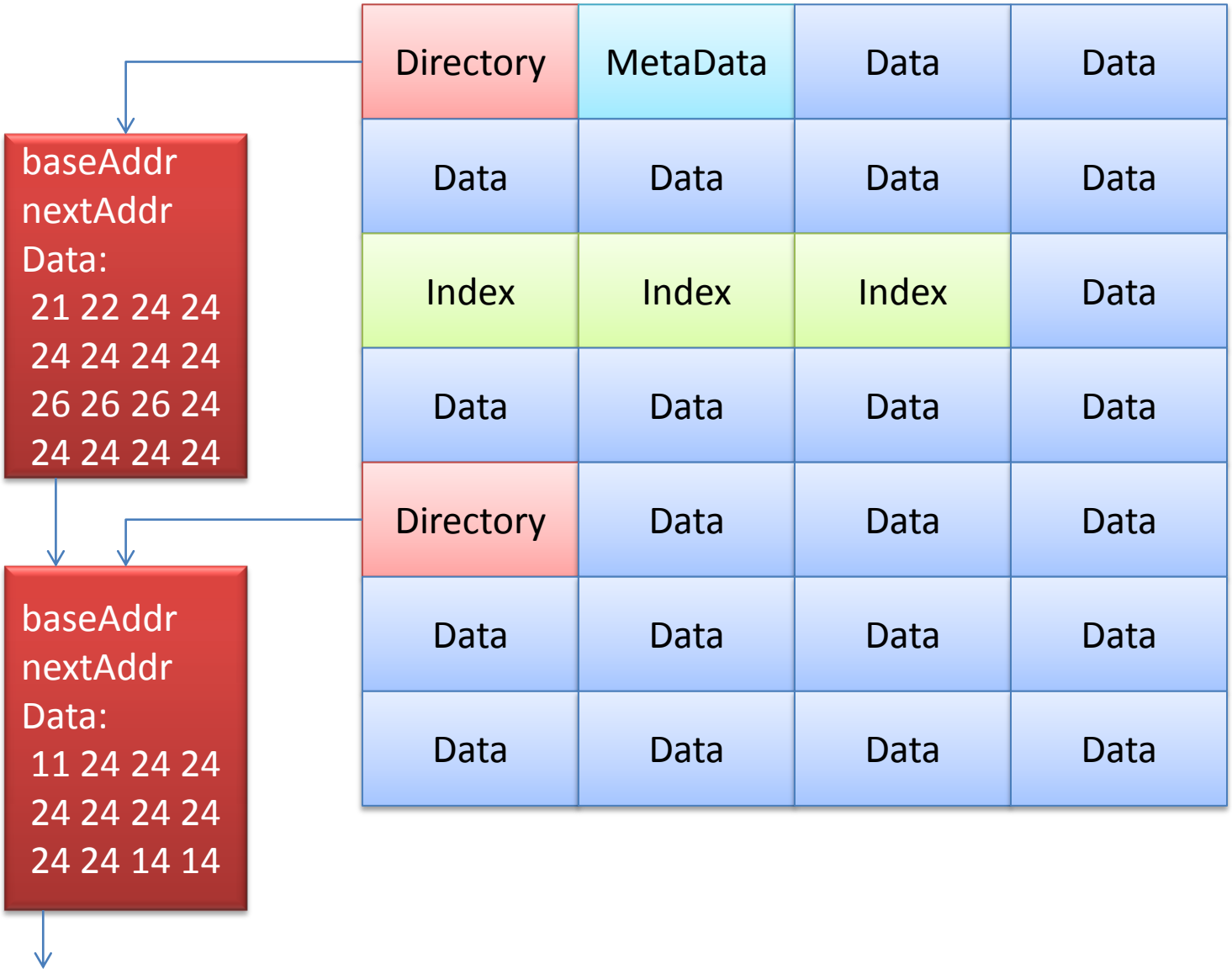
Storage: Buffer



Storage: Segment

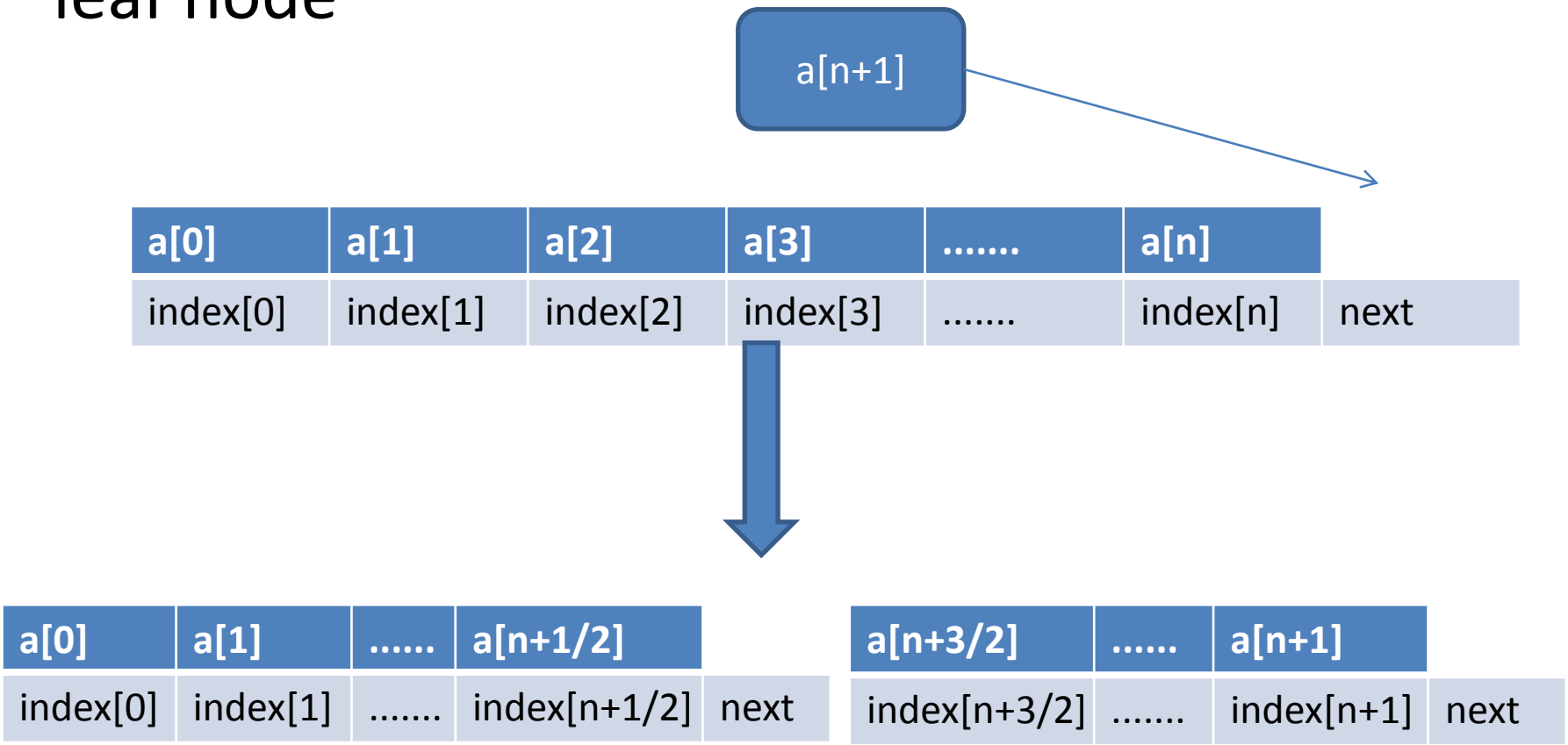
- How to manage pages that store different data?
- Can one page store both a tuple and a index structure(eg. a tree node)?

Segment Design



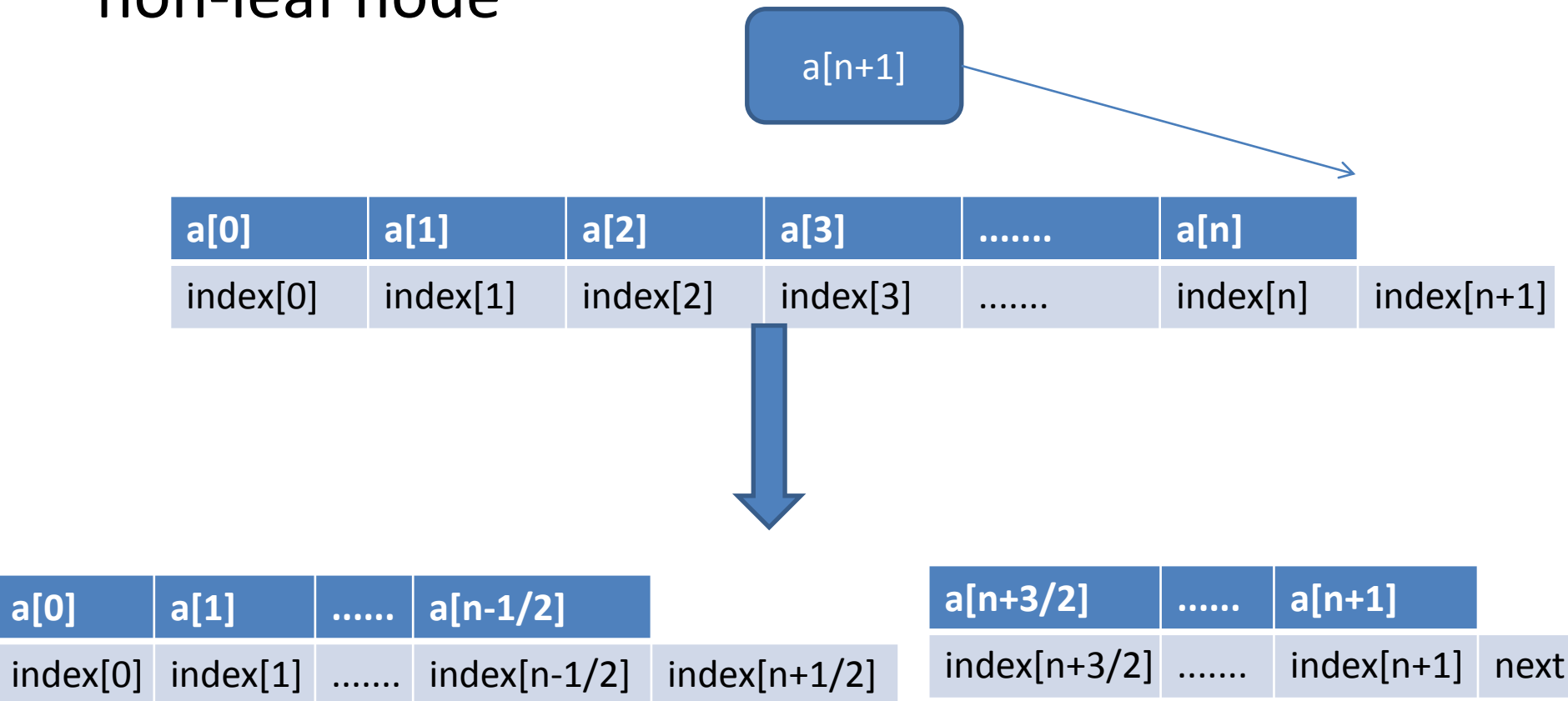
split of node

leaf node



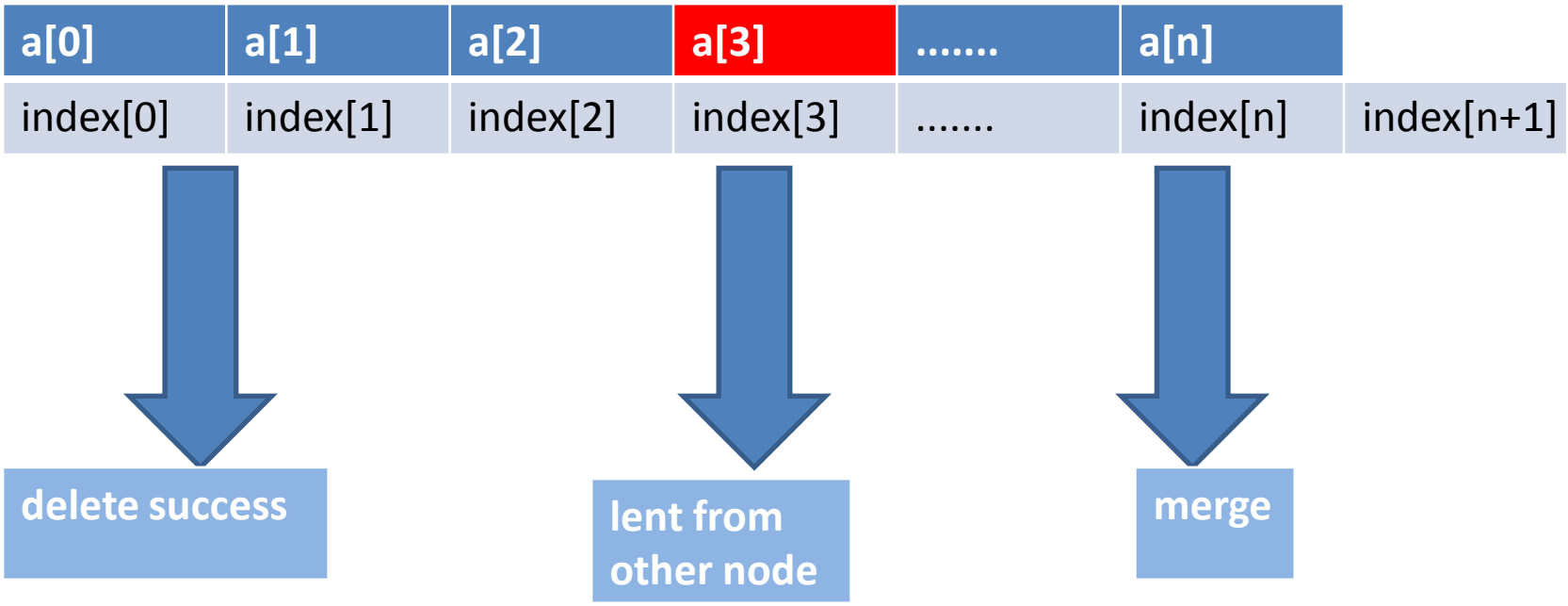
split of node

non-leaf node



merge of node

leaf node



search for node