

Hashfunktion: $h(k) = k \bmod 7$

Hashtabelle:

0	<div>14</div> <div>21</div>
1	
2	<div>2</div> <div>23</div>
3	<div>10</div>
4	
5	<div>12</div> <div>33</div>
6	<div>6</div>

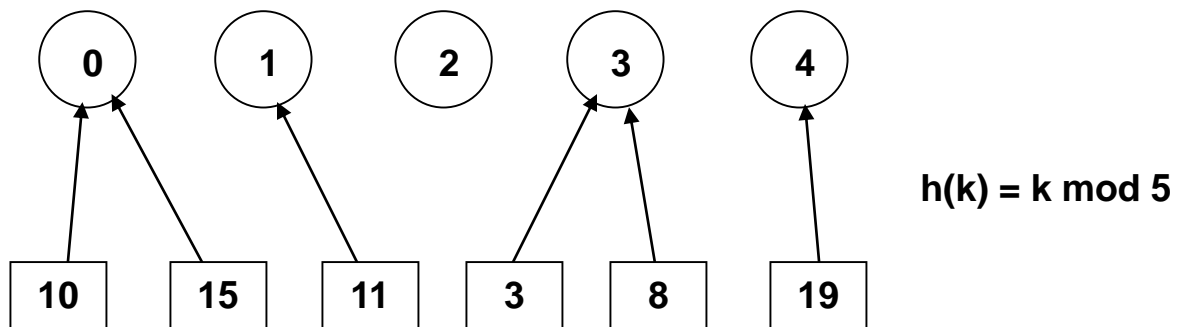
↑
Hashwerte

↑
zugeordnete Daten

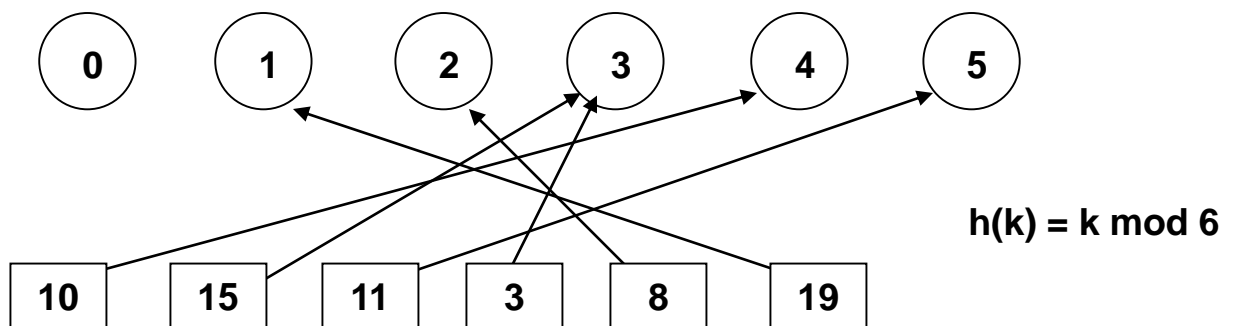
k

 = Daten mit Schlüssel k

Hashfunktion, Hashtabelle



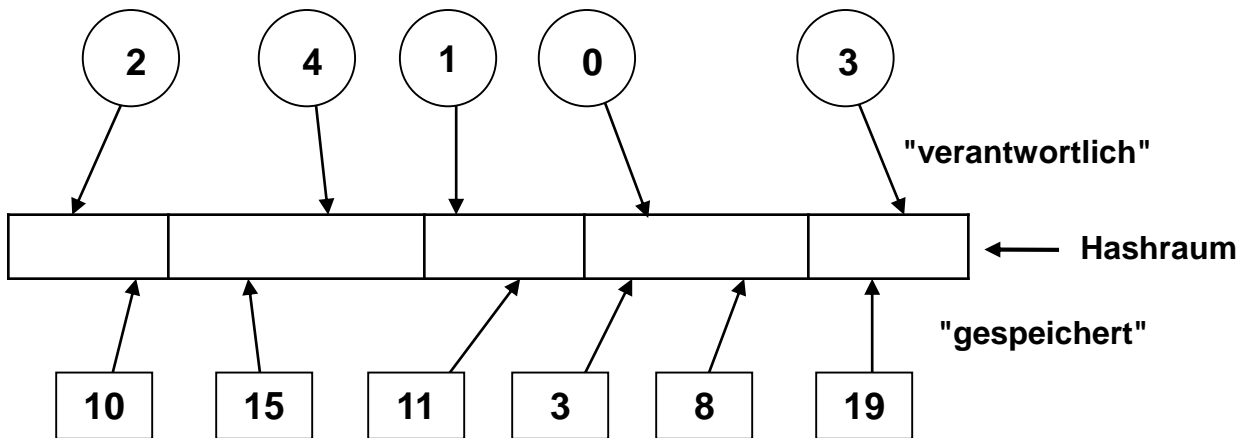
Hinzufügen eines neuen Servers mit Nummer 5:



Neue Hashfunktion: Aufwendiges Umspeichern notwendig!

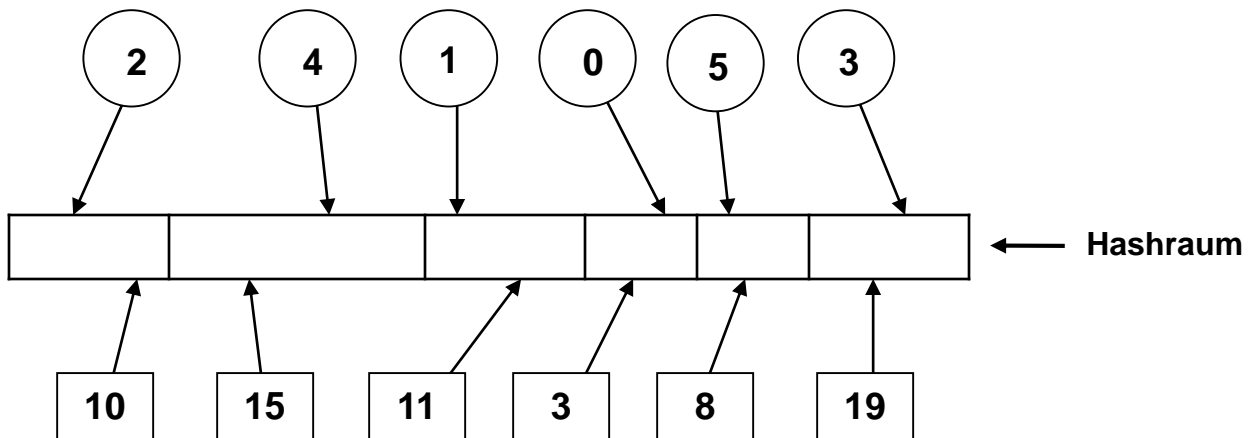


Verteilte Datenbank: Speicherung über Hashfunktion

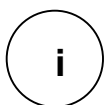


(Server 2 speichert Daten 10; ...; Server 0 speichert Daten 3,8; ...)

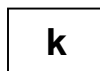
Hinzufügen eines Servers:



(Server 2 speichert Daten 10; Server 0 speichert Daten 3; ...; neuer Server 5 speichert Daten 8; ...)

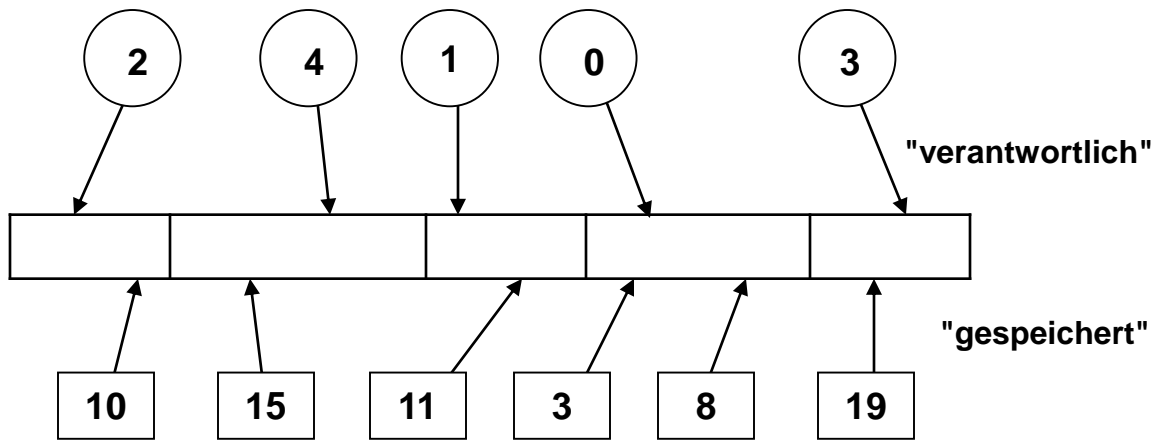


Server i



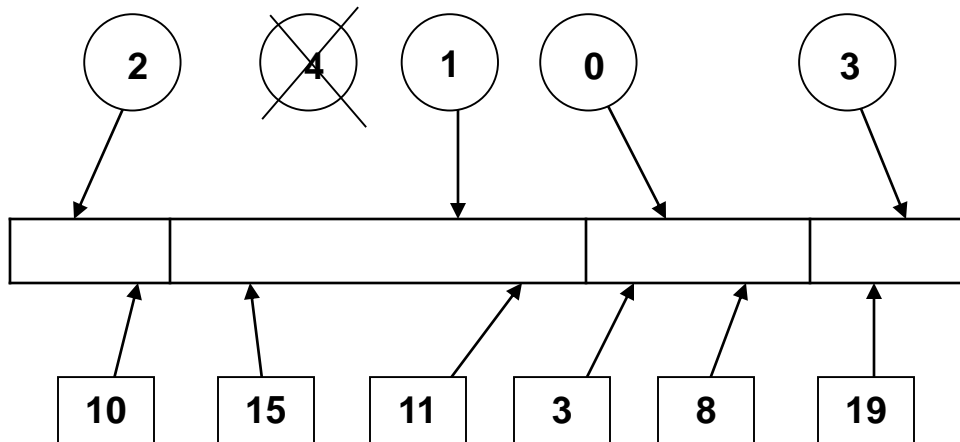
Daten mit Schlüssel k

Verteilte Datenbank:
Konsistentes Hashing, Server hinzufügen VS 9.3

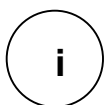


(Server 2 speichert Daten 10; Server 4 speichert Daten 15; ...)

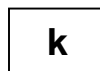
Entfernen eines Servers:



(Server 2 speichert Daten 10; Server 1 speichert Daten 15,11; ...)

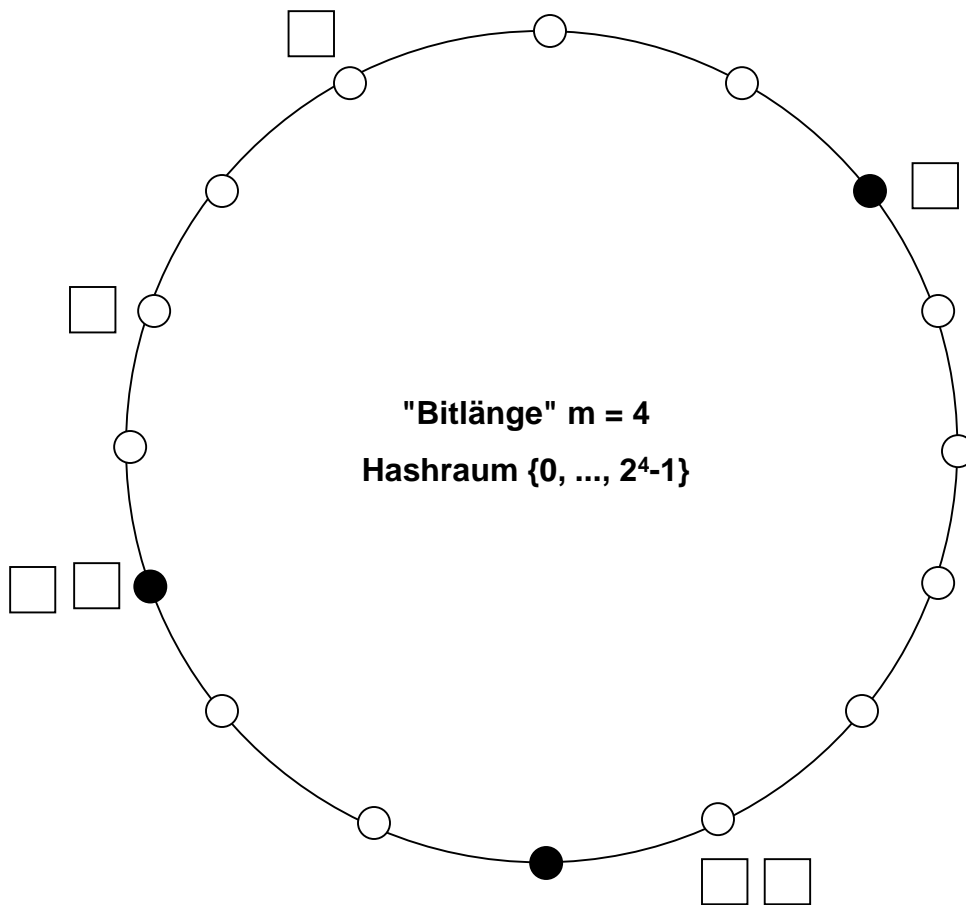


Server i



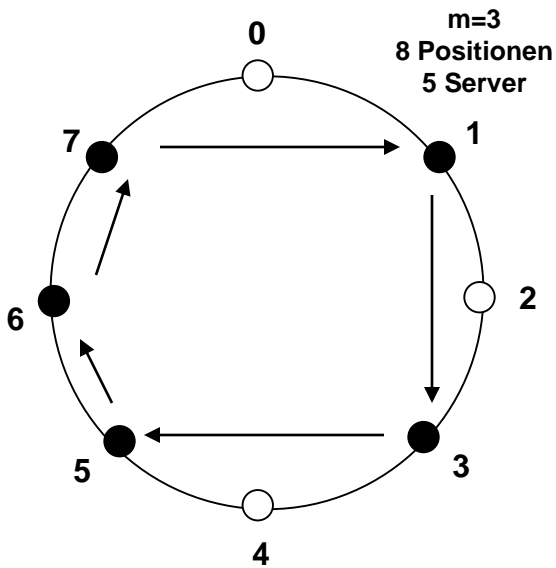
Daten mit Schlüssel k

**Verteilte Datenbank:
Konsistentes Hashing, Server entfernen**

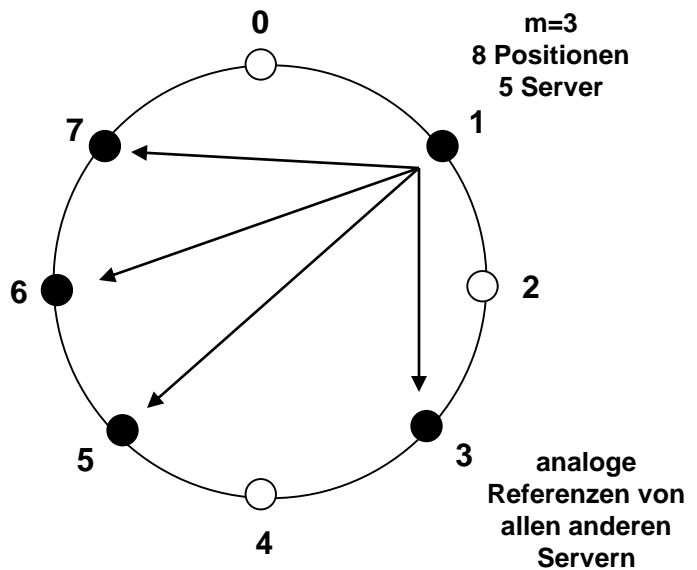


- Positionen
- Server = "Knoten"
- Daten = "Ressourcen"

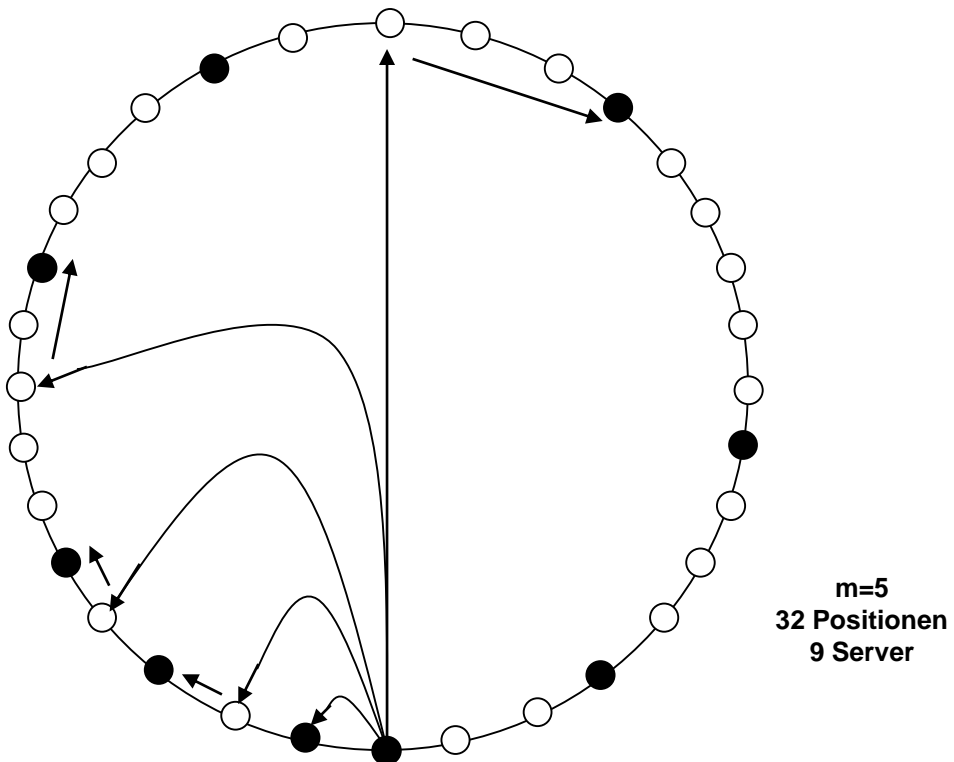
Chord: Server und Daten



jeder Server kennt
nur unmittelbaren
seinen Nachfolger



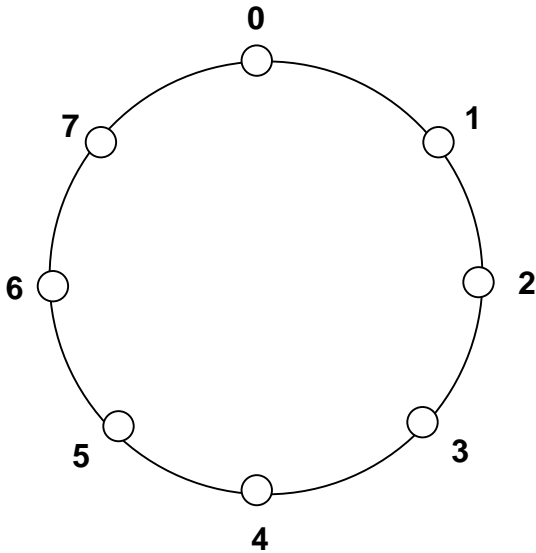
jeder Server kennt
alle anderen Server



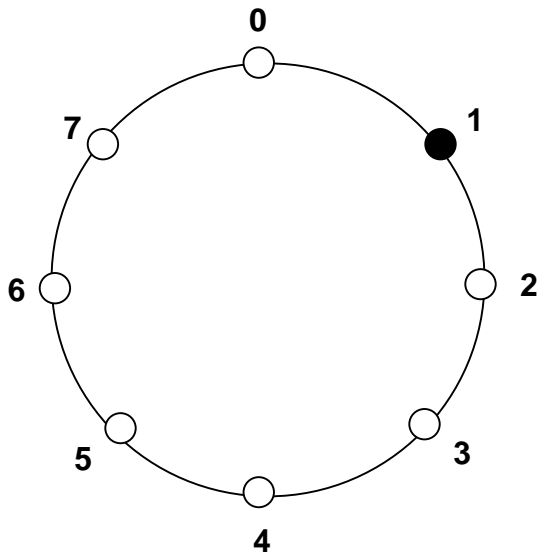
jeder Server kennt die Nachfolger der Positionen:
(eigene Position $+2^i$, $i = 0, 1, 2, \dots$)

Server-Verweise auf dem logischen Ring

$8 = 2^3$ Positionen, d.h. $m=3$



**Anfangszustand
keine Position 0,...,7
durch Server besetzt**

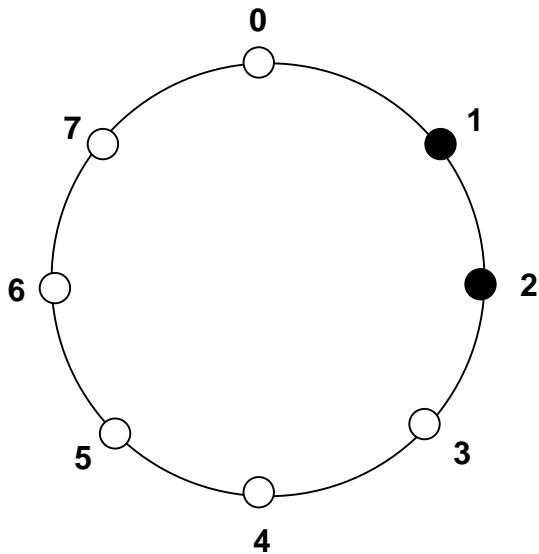


**Server mit id=1 im
Ring an Position 1
⇒ Fingertabelle:**

Server mit id = 1

i = index	s = (id+2ⁱ) mod 2^m	n = Nach- folger
0	2	1
1	3	1
2	5	1

Chord: Server im logischen Ring I



**Neuer Server mit id=2
im Ring an Position 2
⇒ Fingertabellen:**

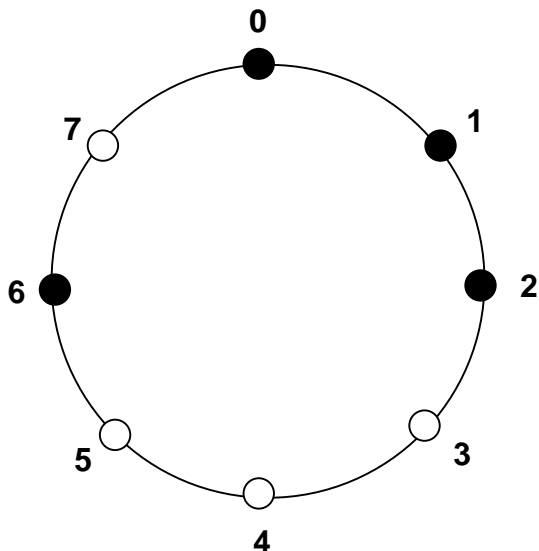
Server mit Id = 2

i	s	n
0	3	1
1	4	1
2	6	1

Server mit Id = 1

i	s	n
0	2	2
1	3	1
2	5	1

**Neue Server mit id=0, id=6 im
Ring an den Positionen 0,6
⇒ Fingertabellen:**



Server mit Id = 2

i	s	n
0	3	6
1	4	6
2	6	6

Server mit Id = 1

i	s	n
0	2	2
1	3	6
2	5	6

Server mit Id = 0

i	s	n
0	1	1
1	2	2
2	4	6

Server mit Id = 6

i	s	n
0	7	0
1	0	0
2	2	2

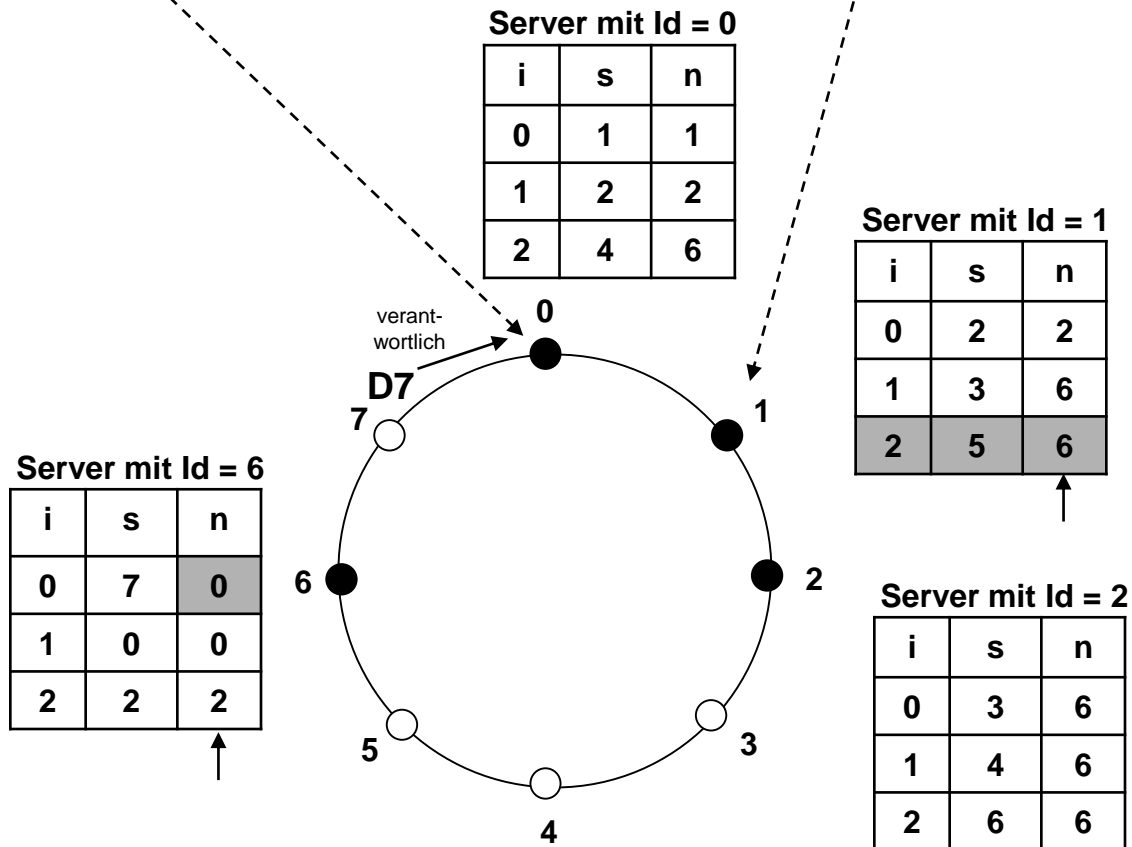
$8 = 2^3$ Positionen, d.h. $m=3$

i =index, $s = (id+2^i) \bmod 2^m$, n = nächster Server hinter s

Chord: Server im logischen Ring II

auf diesem Server ist ein Datenelement mit Hashwert=7 gespeichert

von diesem Server aus erfolgt die Anfrage



Anfrage nach Datenelement mit Hashwert=7 an Server mit id=1:

Server mit id=1 sucht Datenelement mit Hashwert=7

//gespeichert auf Server mit Id=0:

Selbst: Datenelement ist nicht bei ihm lokal gespeichert,

Nachfolger: Server mit id=2 ist nicht verantwortlich für das Datenelement,

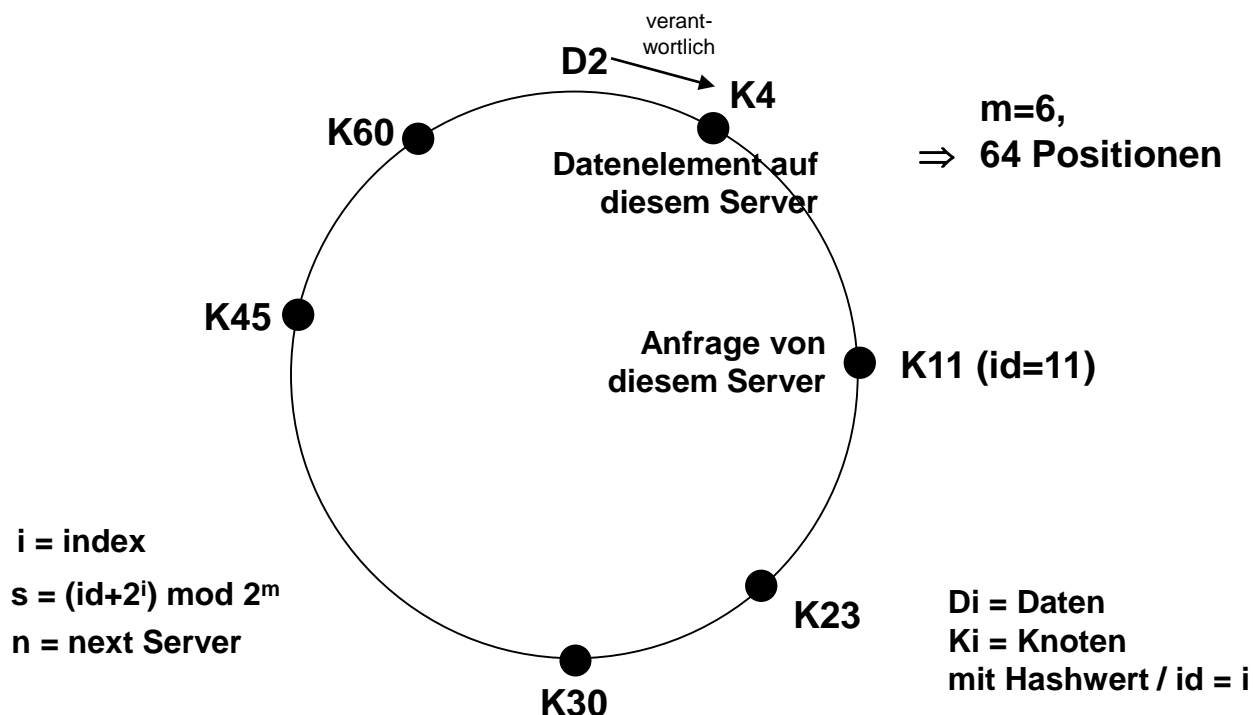
Fingertabelle: Such-Request wird an Server mit id=6 weitergeleitet.

Also: Server mit id=6 sucht Datenelement mit Hashwert=7:

Selbst: Datenelement ist nicht bei ihm lokal gespeichert,

Nachfolger: Server mit id=0 ist verantwortlich, dort ist das Datenelement also gespeichert. Ende.

Chord: Suche im logischen Ring



Anfrage nach Datenelement mit Hashwert=2 an Server mit id=11:

Fingertabelle K11

i	s	n
0	12	23
1	13	23
2	15	23
3	19	23
4	27	30
5	43	45

↑
 nächste
 Fingertabelle:
 K45

Fingertabelle K45

i	s	n
0	46	60
1	47	60
2	49	60
3	53	60
4	61	4
5	13	23

↑
 nächste
 Fingertabelle:
 K60

Fingertabelle K60

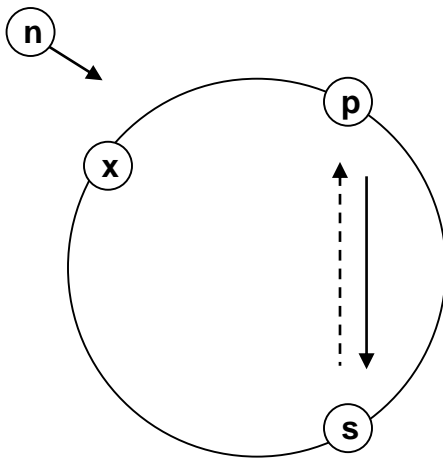
i	s	n
0	61	4
1	62	4
2	0	4
3	4	4
4	12	23
5	28	30

←
 Nachfolger im
 Ring

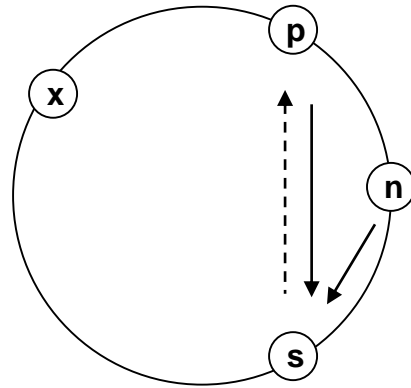
D2 hat eine Position im Ring zwischen K60 und K4, wird daher von K4 gespeichert und bei einer Suche dort gefunden.

(weiterentwickeltes Beispiel aus Martinovsky, Wagner: Einführung in DHTs)

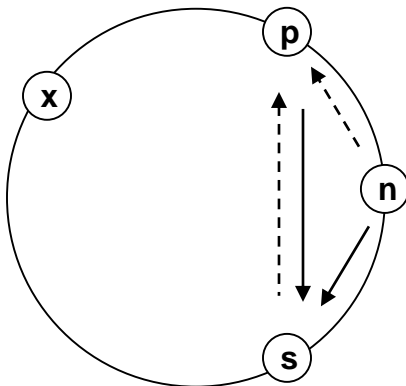
Chord: Lookup-Beispiel



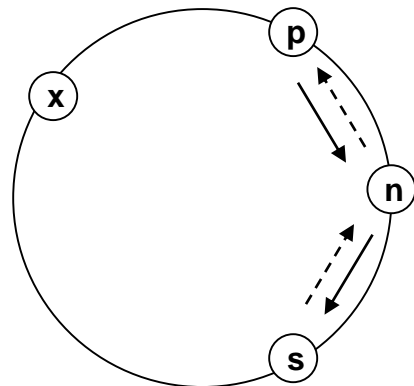
neuer Server n kennt x,
x sucht Platz für n



x findet s mit Lookup-Prozedur,
n positioniert sich zwischen p und s
n kennt jetzt s



n erfährt p von s



n teilt p bzw. s mit,
dass er Nachfolger
bzw. Vorgänger ist

x = beliebiger, dem neuen Server vorher bekannter Server
s=Successor Server, n=neuer Server, p=Predecessor Server

————→ Nachfolgerzeiger
-----→ Vorgängerzeiger

Chord: Einfügen eines neuen Servers