

# Example: Overlapping Insert

Friday, September 29, 2017

10:52 AM



example

example  
insert local text cat pos 0 cli A ref seq 0 Client A inserts locally "cat" at 0  
Node (len 15) p len (0) with 2 live segments:  
min: 11; sc: 1;  
cli: A seq: -1 ← unassigned seq #  
cat  
cli: original seq: 0 ← original text seq # 0  
on the mat.

merge tree for A

```

graph TD
    A[cat] --- B[on the mat.]
  
```

root node  
children of root (leaf nodes)

insert local text big pos 0 cli B ref seq 0 Client B inserts locally "big" at 0  
Node (len 15) p len (0) with 2 live segments:  
min: 11; sc: 1;  
cli: B seq: -1  
big  
cli: original seq: 0  
on the mat.

B's tree

```

graph TD
    B[big] --- C[on the mat.]
  
```

insert local text furry pos 4 cli B ref seq 0 B local insert of "furry" at 4 ref seq 0  
Node (len 21) p len (0) with 3 live segments:  
min: 11; sc: 1;  
cli: B seq: -1  
big  
cli: B seq: -1  
furry  
cli: original seq: 0  
on the mat.

after "big"

```

graph TD
    B[big] --- C[furry] --- D[on the mat.]
  
```

B's tree

@cli A ack seq # 1 server acks A's insert of "cat" & assigning seq # 1  
Node (len 15) p len (0) with 2 live segments:  
min: 11; sc: 2; (1,4) Client A[(1,4)] ← partial length info seq 1 → length + 4 (and client A added this)  
cli: A seq: 1 ← seq # changed to 1  
cat  
cli: original seq: 0  
on the mat.

A's tree

```

graph TD
    A[cat] --- B[on the mat.]
  
```

@cli B text cat seq 1 insert remote pos 0 refseq 0 cli A server tells B of A's insert of "cat" 4  
Node (len 25) p len (0) with 4 live segments:  
min: 11; sc: 4; (1,4) Client A[(1,4)] ← now B has 0 len info.  
cli: B seq: -1  
big  
cli: B seq: -1  
furry  
cli: A seq: 1  
cat  
cli: original seq: 0  
on the mat.

A & B agree on min  
B's tree

```

graph TD
    B[big] --- C[furry] --- D[cat] --- E[on the mat.]
    B --- F[-1]
    C --- G[-1]
    D --- H[unassigned seq #]
  
```

tie at position 0 but big & furry shifted left because unassigned seq # is later than seq # 1

@cli B ack seq # 2 server ack B "big" seq #  
Node (len 25) p len (0) with 4 live segments:

min: 11; sc: 4; (1,4) (2,8) Client B[(2,4)] Client A[(1,4)]  
 cli: B seq: 2  
 big  
 cli: B seq: -1  
 furry  
 cli: A seq: 1  
 cat  
 cli: original seq: 0  
 on the mat.

*seq #1 + 4*  
*seq #2 + 8 (cumulative)*  
*example*  
*more interesting len info on the mat.*  
*big furry cat*

*B's tree*

@cli A text big seq 2 insert remote pos 0 refseq 0 cli B  
 Node (len 19) p len (0) with 3 live segments:  
 min: 11; sc: 3; (1,4) (2,8) Client A[(1,4)] Client B[(2,4)]  
 cli: B seq: 2  
 big  
 cli: A seq: 1  
 cat  
 cli: original seq: 0  
 on the mat.

*server notifies A of B's "big", assigning seq #2*  
*now A has len info same as B*  
*big cat on the mat.*

*A's tree*

@cli B ack seq # 3  
 Node (len 25) p len (0) with 4 live segments:  
 min: 11; sc: 4; (1,4) (2,8) (3,14) Client B[(2,4)(3,10)] Client A[(1,4)]  
 cli: B seq: 2  
 big  
 cli: B seq: 3  
 furry  
 cli: A seq: 1  
 cat  
 cli: original seq: 0  
 on the mat.

*server acks B's insert of furry*  
*2 len info*  
*2 segs from B; still one seg from A*  
*big furry cat on the mat.*

*B's tree*

@cli A text furry seq 3 insert remote pos 4 refseq 0 cli B  
 Node (len 25) p len (0) with 4 live segments:  
 min: 11; sc: 4; (1,4) (2,8) (3,14) Client A[(1,4)] Client B[(2,4)(3,10)]  
 cli: B seq: 2  
 big  
 cli: B seq: 3  
 furry  
 cli: A seq: 1  
 cat  
 cli: original seq: 0  
 on the mat.

*server informs A of "furry" from B*  
*B did not see "cat"*  
*big ignored seq #2 > up seq #6*  
*furry added*  
*cat ignored*  
*on the mat. original*

*A's tree*

B sees the cat!  
 insert local text cat pos 0 cli A ref seq 0  
 Node (len 15) p len (0) with 2 live segments:  
 min: 11; sc: 1;

*Same example but B inserts after cat, referencing seq #1;*  
*will annotate differences*

example

```
cli: A seq: -1
cat
cli: original seq: 0
on the mat.
```

*same this page*

```
insert local text big pos 0 cli B ref seq 0
Node (len 15) p len (0) with 2 live segments:
min: 11; sc: 1;
cli: B seq: -1
big
cli: original seq: 0
on the mat.
```

```
@cli A ack seq # 1
Node (len 15) p len (0) with 2 live segments:
min: 11; sc: 2;(1,4) Client A[(1,4)]
cli: A seq: 1
cat
cli: original seq: 0
on the mat.
```

```
@cli B text cat seq 1 insert remote pos 0 refseq 0 cli A
Node (len 19) p len (0) with 3 live segments:
min: 11; sc: 3;(1,4) Client A[(1,4)]
cli: B seq: -1
big
cli: A seq: 1
cat
cli: original seq: 0
on the mat.
```

```
@cli B ack seq # 2
Node (len 19) p len (0) with 3 live segments:
min: 11; sc: 3;(1,4) (2,8) Client B[(2,4)]Client A[(1,4)]
cli: B seq: 2
big
cli: A seq: 1
cat
cli: original seq: 0
on the mat.
```

```
@cli A text big seq 2 insert remote pos 0 refseq 0 cli B
Node (len 19) p len (0) with 3 live segments:
min: 11; sc: 3;(1,4) (2,8) Client A[(1,4)]Client B[(2,4)]
cli: B seq: 2
big
cli: A seq: 1
cat
```

Page 3

# example

cli: original seq: 0  
on the mat.

insert local text furry pos 8 cli B ref seq 2  
Node (len 25) p len (0) with 4 live segments:  
min: 11; sc: 3;(1,4) (2,8) Client B[(2,4)]Client A[(1,4)]  
cli: B seq: 2

B inserts "furry" referencing  
seq #2  
(goes "cat")

B's tree

big  
cli: A seq: 1  
cat  
cli: B seq: -1  
furry  
cli: original seq: 0  
on the mat.



@cli B ack seq # 3 server acks B  
Node (len 25) p len (0) with 4 live segments:  
min: 11; sc: 4;(1,4) (2,8) (3,14) Client B[(2,4)(3,10)]Client A[(1,4)]  
cli: B seq: 2

B's tree

big  
cli: A seq: 1  
cat  
cli: B seq: 3  
furry  
cli: original seq: 0  
on the mat.

@cli A text furry seq 3 insert remote pos 8 refseq cli B  
Node (len 25) p len (0) with 4 live segments:  
min: 11; sc: 4;(1,4) (2,8) (3,14) Client A[(1,4)]Client B[(2,4)(3,10)]  
cli: B seq: 2

2 ← bpo from earlier  
prntout  
server informs A of  
"furry" from B

A's tree

big  
cli: A seq: 1  
cat  
cli: B seq: 3  
furry  
cli: original seq: 0  
on the mat.

