Region Server Failure Recovery Time Comparison

Test Setup

Each test case has 41-43 WAL files are recovered. Each WAL file is about 6.6M size and single WAL edit value length is 200 bytes and total amount data recoverred are 270M - 280M

Envriroment

Hbase 0.96 Trunk
EC2 m1.large instance
5 Nodes Cluster with 4 data nodes and 4 region servers. Kill one region server and the rest 3 region servers are paticipating in recovery.

Current Implementation with creations of recovered edits files

Number of Regions Recovered	Log Splitting Time(ms)	Region Opening Time(ms)	Total Recovery Time For Writes(ms)	Total Recovery Time For Reads(ms)*
80	47629	22692	70321	70321
160	67712	30172	97884	97884
320	108357	53740	162097	162097

Proposed Implementation by direclty replaying WAL edits to fail over region servers

Number of Regions Recovered	Log Splitting Time(ms)	Region Opening Time(ms)	Total Recovery Time For Writes(ms)*	Total Recovery Time For Reads(ms)
80	37298	1156	1156	38454
160	40054	1630	1630	41684
320	51960	2834	2834	54794

Notes:

1) Some regions will be available sooner because once a region server finishes replaying all recovered.edits for a region, the region is fully recovered. The reovery time used here is the time spent till all regions are opened.
2) In new implementation, we allow writes while rejecting reads during recovering. (We could disable this by a configuation setting)

Summary

As we can see with the same amount of data when number of regions increases, existing log splitting performance is degrading nonlinearly. Because it needs write (sum of # of WAL files * # of regions inside the WAL) small files concurrently, It causes many random IOs and will be bounded by random disk IO thoughput of the underlying system. The new implemantation doesn't create intermediate small files so performance is much better and won't affect much by number of WAL or regions to be recovered.