HBase Table layout (v0.2) Grouperfish clustering engine

documents

	row key	utf8	f_1(namespace, collection_key, id)*
main	**namespace**	utf8	namespace
main	**collection_key**	utf8	name of the collection
main	**id**	utf8	document id (given by the application)
main	**text**	utf8	document text
main	**member_of**	utf8	last assigned cluster label
*processing***	***id***	*utf8*	*document id (given by the application), reduncant for scans*
*processing***	***vector_idf***	*byte[]*	*serialization of a mahout sparse feature vector (matching the collection dictionary)*
*processing***	***vector_tfidf***	*byte[]*	*corresponding tf/idf vector*

clusters

	row key	utf8	f_2(namespace, collection_key, $conf, timestamp, label)**
	namespace	utf8	namespace
	collection_key	utf8	name of the collection
	timestamp	utf8	When this cluster was fully rebuilt. Matches the configuration:$conf:rebuilt item of this conf.
	label	utf8	numeric id or (better) descriptive text label
	size	utf8	current size of the cluster
	$documentid	utf8	the score for the given document in the cluster

collections

	row key	utf8	f_3(namespace, collection_key)
main	**namespace**	utf8	namespace
main	**collection_key**	utf8	name of the collection
main	**size**	utf8	base 10 integer
main	**modified**	utf8	base 10 unix timestamp / last document addition time
main	**configuration:$conf:rebuilt**	utf8	base10 unix ts / when this configuration was fully rebuilt
main	**configuration:$conf:processed**	utf8	base10 unix ts / when this configuration was updated
processing	***dictionary***	*byte[]*	*dictionary descibing the document's (TF) IDF vector features*
processing	***index***	*byte[]*	*inverted index (for similarity based algorithms)*

* The values of $f_1(\ldots)$ for the same collection are all prefixed with the same $f'_1(namespace, collection\_key)$. This allows to quickly scan for all documents of a collection.

** The values of $f_2(\ldots)$ for the same cluster are all prefixed with the same $f'_2(namespace, collection\_key, \$conf, ts)$. This allows to quickly get all documents in a cluster.

Note that for each collection we can store a clustering for each (conf, ts) combination. This allows to store a new clustering while serving the previous one.

*** The *processing* family is not used yet (only full rebuilds are supported).