//Olubukola Akintoroye

#include “rc4test.h”

........

.........

int FdEntity::Load(off\_t start, size\_t size)

{

S3FS\_PRN\_DBG("[path=%s][fd=%d][offset=%jd][size=%jd]", path.c\_str(), fd, (intmax\_t)start, (intmax\_t)size);

//encryp(fd);

if(-1 == fd){

return -EBADF;

}

AutoLock auto\_lock(&fdent\_lock);

int result = 0;

// check loaded area & load

fdpage\_list\_t unloaded\_list;

if(0 < pagelist.GetUnloadedPages(unloaded\_list, start, size)){

for(fdpage\_list\_t::iterator iter = unloaded\_list.begin(); iter != unloaded\_list.end(); ++iter){

if(0 != size && static\_cast<size\_t>(start + size) <= static\_cast<size\_t>((\*iter)->offset)){

// reached end

break;

}

// check loading size

size\_t need\_load\_size = 0;

if(static\_cast<size\_t>((\*iter)->offset) < size\_orgmeta){

// original file size(on S3) is smaller than request.

need\_load\_size = (static\_cast<size\_t>((\*iter)->next()) <= size\_orgmeta ? (\*iter)->bytes : (size\_orgmeta - (\*iter)->offset));

}

size\_t over\_size = (\*iter)->bytes - need\_load\_size;

// download

if(static\_cast<size\_t>(2 \* S3fsCurl::GetMultipartSize()) <= need\_load\_size && !nomultipart){ // default 20MB

// parallel request

// Additional time is needed for large files

time\_t backup = 0;

if(120 > S3fsCurl::GetReadwriteTimeout()){

backup = S3fsCurl::SetReadwriteTimeout(120);

}

result = S3fsCurl::ParallelGetObjectRequest(path.c\_str(), fd, (\*iter)->offset, need\_load\_size);

if(0 != backup){

S3fsCurl::SetReadwriteTimeout(backup);

}

}else{

// single request

if(0 < need\_load\_size){

S3fsCurl s3fscurl;

result = s3fscurl.GetObjectRequest(path.c\_str(), fd, (\*iter)->offset, need\_load\_size);

}else{

result = 0;

}

}

if(0 != result){

break;

}

encryp(fd);

....

....

}

int FdEntity::RowFlush(const char\* tpath, bool force\_sync)

{

int result;

S3FS\_PRN\_INFO3("[tpath=%s][path=%s][fd=%d]", SAFESTRPTR(tpath), path.c\_str(), fd);

// encryp(fd);

if(-1 == fd){

return -EBADF;

}

AutoLock auto\_lock(&fdent\_lock);

if(!force\_sync && !is\_modify){

// nothing to update.

return 0;

}

encryp(fd);

// If there is no loading all of the area, loading all area.

size\_t restsize = pagelist.GetTotalUnloadedPageSize();

......

......

}