

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده مهندسی برق و رباتیک

پایان نامه کارشناسی ارشد مهندسی مخابرات سیستم

ارائه روش نوین کدگذاری در جهت بهبود عملکرد سیستم‌های ذخیره‌سازی ابری

نگارنده: نسترن چکنی

استاد راهنما

دکتر سید مسعود میررضایی

استاد مشاور

دکتر قوشه عابد هدتنی

شهریورماه ۱۳۹۷

شماره: ۱۵۹۰
تاریخ: ۹۷/۱/۱۰

باسمه تعالی



مدیریت تحصیلات تکمیلی

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای نسترن چکنی با شماره دانشجویی ۹۴۰۵۴۳۴ رشته مهندسی برق-مخابرات گرایش سیستم تحت عنوان: ارائه روش نوین کدگذاری در جهت بهبود عملکرد سیستم های ذخیره سازی ابری که در تاریخ ۱۳۹۷/۰۶/۱۰ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

قبول (با درجه: خوب) مردود
 نوع تحقیق: نظری عملی

عضو هیأت داوران	نام و نام خانوادگی	مرتبه علمی	امضاء
۱- استاد راهنمای اول	سید محمد میرزایی	استاد	
۲- استاد راهنمای دوم	—	—	—
۳- استاد مشاور	—	—	—
۴- نماینده تحصیلات تکمیلی	هادی کرایی	استاد	
۵- استاد ممتحن اول	سید علی حسینی	دانشیار	
۶- استاد ممتحن دوم	ایرینا سررخی	استاد	

نام و نام خانوادگی رئیس دانشکده: محمد صالحی
 وزارت علوم، تحقیقات و فناوری
 تاریخ و امضاء و مهر دانشکده:

تبصره: در صورتی که کسی مردود شود حداکثر یکبار دیگر (معمولاً در صورتی که در مقطع تحصیلی) می تواند از پایان نامه خود دفاع نماید (دفاع مجدد نباید زودتر از ۴ ماه برگزار شود).

تقدیم

به مادرم که همیشه همراه و مشوقم در علم آموزی بود
و تمام کسانی که عشق به یادگیری را در من زنده نگه داشته‌اند
و علم را راهی برای رشد برگزیده‌اند

سپاسگزاری

در این مسیر بسیار خرسندم که در کنار استاد بزرگوارم دکتر میررضایی کام به کام بردانایی خود افزودم و سپاسگزار ایشان هستم که لطف و دانش خود را همراه و به کام من کردند.

همچنین از استاد گرانقدر دکتر مهدی که بهره‌مندی از علم و تجربه ایشان برای من افتخاری بزرگ بود و مشاوره‌های ارزشمند خود را در پیچ‌زمینه‌ای از من دریغ نکردند، کمال تشکر را دارم.

و از استاد گرامی دکتر خسروی که با لطف و همراهی بی‌درنشان مسیر پایان نامه را برایم هموار کردند نهایت سپاس را دارم.

و در پایان از خانواده‌ام، که همیشه حامی و همراه من بودند و هستند قدر دانی می‌کنم.

تهیه‌نامه

اینجانب **نسترن چکنی** دانشجوی دوره کارشناسی ارشد (دکتری) رشته برق-مخابرات دانشکده مهندسی برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان‌نامه ارائه روش نوین کدگذاری در جهت بهبود عملکرد سیستم‌های ذخیره‌سازی ابری تحت راهنمایی دکتر میررضایی متعهد می

شوم

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ‌جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می‌باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به‌دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آن‌ها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود. استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

حکیده

در سال‌های اخیر با توجه به رشد چشم‌گیر داده و پیش‌بینی‌ها از تداوم این روند، نیاز به ارائه راهکارهای مناسب برای ذخیره‌سازی داده بیش از گذشته احساس می‌شود. با شروع به کار سیستم‌های ذخیره‌سازی ابری از سال ۲۰۰۶، توسعه و بهبود این سیستم‌ها در راستای پاسخگویی به این نیاز رو به رشد بسیار مورد توجه قرار گرفته است. در این پایان‌نامه با ارائه روش‌هایی عملکرد سیستم‌های ذخیره‌سازی ابری را بهبود می‌بخشیم.

با توجه به الگوی «پرداخت به‌ازای مصرف» در ذخیره‌سازی ابری و محدودیت‌ها، ابتدا با استفاده از معیار جدید «میزان اهمیت پیام» که در تشخیص زیرمجموعه اقلیت به کار می‌رود، به فشردن داده پیش از برون‌سپاری روی ابر می‌پردازیم. با مقایسه این روش با فشردن‌سازی هافمن، شانون و شانون فانو الیاس نشان می‌دهیم که در مواردی که بخش مهم پیام مورد نظر است، می‌توان فضای مورد نیاز برای ذخیره‌سازی را تا ۳۰ درصد کاهش داد. همچنین اثر آستانه حذف نماد در این روش فشردن‌سازی را مورد بررسی قرار می‌دهیم.

در ادامه پایان‌نامه، روش ذخیره‌سازی مبتنی بر کدگذاری را با کدهای LT پیاده‌سازی کرده و به منظور بهبود عملکرد این سیستم‌ها در بازیابی داده، سه توزیع شکرالهی، PRSD و CPRSD را علاوه بر توزیع RSD به کار می‌بریم. نشان می‌دهیم که با استفاده از توزیع‌های جدید می‌توان به‌ازای سربار کمتر، احتمال موفقیت بازیابی را افزایش داد و همچنین فضای مورد نیاز برای ذخیره‌سازی نسبت به ذخیره‌سازی مبتنی بر تکرار کاهش چشم‌گیری دارد. علاوه بر این، با استفاده از الگوریتمی بر پایه میانگین و انحراف معیار تعداد نمادهای کدگذاری مورد نیاز برای کدگشایی موفق، زمان بازیابی داده را کاهش می‌دهیم.

کلمات کلیدی: ذخیره‌سازی ابری، میزان اهمیت پیام، ذخیره‌سازی مبتنی بر کدگذاری، کدهای LT، توزیع درجه

فهرست مطالب

ی	فهرست جداول	۱
ک	فهرست اشکال	۱
م	فهرست اختصارات	۱
۱	فصل ۱: مقدمه	۱
۲	۱-۱ مقدمه	۲
۳	۲-۱ بیان مسئله و اهداف پایان نامه	۳
۴	۳-۱ ساختار پایان نامه	۴
۷	فصل ۲: مروری بر کارهای پیشین	۷
۱۹	فصل ۳ مفاهیم پایه	۱۹
۲۲	۱-۳ مقدمه	۲۲
۲۰	۲-۳ سیستم ذخیره سازی ابری	۲۰
۲۱	۱-۲-۳ ذخیره سازی تک منطقه ای	۲۱
۲۱	۲-۲-۳ ذخیره سازی چند منطقه ای	۲۱
۲۲	۳-۲-۳ مبانی صف M/G/1	۲۲
۲۲	۳-۳ کدگذاری منبع	۲۲
۲۳	۱-۳-۳ کدگذاری مبتنی بر میزان اهمیت پیام	۲۳
۲۵	۲-۳-۳ کدگذاری شانون	۲۵
۲۵	۳-۳-۳ کدگذاری شانون فانو الیاس	۲۵
۲۵	۴-۳-۳ کدگذاری هافمن	۲۵
۲۶	۴-۳ کدگذاری کانال	۲۶

۲۸.....	۱-۴-۳ کدهای LT
۳۷.....	فصل ۴: نوآوری و نتایج شبیه‌سازی
۳۷.....	۱-۴ مقدمه
۳۸.....	۲-۴ کدگذاری منبع بر پایه MIM
۴۲.....	۳-۴ سیستم ذخیره‌سازی ابری پیشنهادی
۴۴.....	۴-۴ ذخیره‌سازی بر پایه تکرار
۴۵.....	۵-۴ ذخیره‌سازی بر پایه کدهای LT
۴۶.....	۱-۵-۴ انتخاب n
۴۷.....	۲-۵-۴ انتخاب مؤلفه‌ها
۵۰.....	۳-۵-۴ بررسی احتمال موفقیت بازیابی
۵۴.....	۶-۴ بهبود زمان بازیابی
۶۱.....	۷-۴ نتیجه‌گیری
۶۳.....	۸-۴ پیشنهاد کارهای آینده

فهرست جداول

- جدول ۴-۱. مقایسه حجم ذخیره‌سازی حاصل از روش‌ها به‌ازای آستانه‌های مختلف ۴۲
- جدول ۴-۲. تعداد نماد کدگذاری با RSD ۴۷
- جدول ۴-۳. مقایسه حجم موردنیاز برای ذخیره‌سازی (مگابایت) برای دو حالت تکرار و کدگذاری .. ۵۴
- جدول ۴-۴. میانگین و انحراف معیار تعداد نمادهای کدگذاری پس از تطبیق با توزیع نرمال ۵۶
- جدول ۴-۵. مقایسه زمان بازیابی (ثانیه) در حالت اصلی (استفاده از k و n) و الگوریتم پیشنهادی .. ۵۹
- جدول ۵-۱. فاصله نواحی دسترسی تا مکان درخواست کاربر برای ذخیره و بازیابی داده ۶۴

فهرست اشکال

- شکل ۱-۳. فرایند کدگذاری LT ۲۹
- شکل ۲-۳. فرایند کدگشایی LT [۲۴] ۳۰
- شکل ۳-۳. توزیع RSD برای $k=100$, $c=0.08$ و $\delta=0.1$ ۳۲
- شکل ۴-۳. توزیع درجه‌های برای مقادیر k ، سربار ε و میانگین درجه k [۳۴] ۳۳
- شکل ۵-۳. توزیع PRSD برای $k=100$, $c=0.08$, $\delta=0.1$ و $\lambda=3.04$ ۳۵
- شکل ۶-۳. توزیع CPRSD برای $k=100$, $c=0.08$, $\delta=0.1$, $\lambda=3.04$ و $a=0.42$ ۳۶
- شکل ۱-۴. نرخ و آنتروپی اهمیتی ۳۹
- شکل ۲-۴. مقایسه نرخ در روش‌های فشرده‌سازی MIM، هافمن، شانون و شانون فانو الیاس ۳۹
- شکل ۳-۴. مقایسه افزونگی در روش‌های هافمن، MIM، شانون و شانون فانو الیاس ۴۰
- شکل ۴-۴. افزونگی برای آستانه‌های مختلف ۴۱
- شکل ۵-۴. ساختار کلی سیستم ذخیره‌سازی ابری مدنظر ۴۳
- شکل ۶-۴. ذخیره‌سازی و بازیابی داده در حالت ذخیره‌سازی برپایه تکرار ۴۴
- شکل ۷-۴. ذخیره‌سازی و بازیابی داده در حالت ذخیره‌سازی بر پایه کدگذاری LT ۴۶
- شکل ۸-۴. احتمال موفقیت به‌ازای تعداد نماد کدگذاری متفاوت [۲۸] ۴۶
- شکل ۹-۴. میزان موفقیت کدگشایی با RSD به‌ازای c و δ های مختلف برای $k=250$ ۴۷
- شکل ۱۰-۴. میزان موفقیت کدگشایی PRSD به‌ازای c و δ های مختلف برای $k=250$ ۴۸
- شکل ۱۱-۴. میزان موفقیت کدگشایی LT با توزیع CPRSD به‌ازای a و k های مختلف ۴۹
- شکل ۱۲-۴. میزان موفقیت کدگشایی LT با توزیع CPRSD برای $k=250$ به‌ازای a های مختلف ۴۹
- شکل ۱۳-۴. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های RSD، PRSD، CPRSD و شکرالهی برای $k=100$ بدون حذف مراکز داده ۵۰
- شکل ۱۴-۴. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های RSD، PRSD، CPRSD و شکرالهی برای $k=250$ بدون حذف مراکز داده ۵۱
- شکل ۱۵-۴. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های RSD، PRSD، CPRSD و شکرالهی برای $k=500$ بدون حذف مراکز داده ۵۱
- شکل ۱۶-۴. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف برای $k=100$ و حذف یک مرکز داده ۵۲
- شکل ۱۷-۴. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف برای $k=250$ و حذف یک مرکز داده ۵۲

- شکل ۴-۱۸. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف برای $k = 500$ و حذف یک مرکز داده. ۵۳
- شکل ۴-۱۹. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف برای $k = 250$ و حذف دو مرکز داده. ۵۳
- شکل ۴-۲۰. بررسی تفاوت دو مدل ذخیره‌سازی تک‌منطقه‌ای و چندمنطقه‌ای برای توزیع‌های مختلف و $k = 250$. ۵۴
- شکل ۴-۲۱. هیستوگرام نمادهای موردنیاز برای کدگشایی به‌ازای $k = 250$. ۵۶
- شکل ۴-۲۲. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف با استفاده از یک انحراف معیار از میانگین برای $k = 250$. ۵۷
- شکل ۴-۲۳. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف با استفاده از دو انحراف معیار از میانگین برای $k = 100$. ۵۷
- شکل ۴-۲۴. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف با استفاده از دو انحراف معیار از میانگین برای $k = 250$. ۵۸
- شکل ۴-۲۵. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف با استفاده از دو انحراف معیار از میانگین برای $k = 500$. ۵۸

فہرست اختصارات

LT	Luby Transform
MIM	Message Importance Measure
NMIM	Non-parametric Message Importance Measure
MID	Message Importance Divergence
BSC	Binary Symmetric Channel
MEC	Mobile Edge Computing
KL	Kullback Leibler
DMIM	Differential Message Importance Measure
MTTF	Mean Time to Failure
R-S	Reed-Solomon
LRC	Local Reconstruction Codes
WAS	Windows Azure Storage
BP	Belief Propagation
BLCS	Block Level Cloud Storage
OFG	On The Fly Gaussian Elimination
MDS	Maximum Distance Separable
AWS	Amazon Web Service
DWDM	Dense Wavelength Division Multiplexing
BEC	Binary Erasure Channel
IPD	Improved Poisson Distribution
RSD	Robust Soliton Distribution
PRSD	Poisson Robust Soliton Distribution
CPRSD	Combined Poisson Robust Soliton Distribution

فصل ۱: مقدمه

۱-۱ مقدمه

در سال‌های اخیر، رشد فزاینده داده باعث ایجاد تحولات چشم‌گیری در دنیای دیجیتال شده است. بر اساس گزارش شرکت بین‌المللی داده (IDC)، از سال ۲۰۰۵ تا ۲۰۲۰ دنیای دیجیتال از ۱۳۰ اگزابایت^۱ به ۴۰۰۰۰ اگزابایت (بیش از ۲۵۰۰ گیگابایت برای هر نفر در ۲۰۲۰) رشد خواهد داشت [۱]. با توجه به این روند، نیاز به ارائه راهکارهای مؤثری در پردازش و ذخیره‌سازی داده محسوس است. پیدایش مفاهیم اساسی رایانش ابری به دهه ۱۹۶۰ بازمی‌گردد. زمانی که جان مک‌کارتی اظهار داشت که «پردازش ممکن است روزی به‌عنوان یکی از صنایع همگانی سازمان‌دهی شود». در سال ۱۹۶۶ پارک هیل داگلاس تقریباً تمام ویژگی‌های امروز رایانش ابری را به‌همراه مقایسه‌ای با صنعت برق و شکل‌های مصرف عمومی، خصوصی، دولتی و انجمنی در کتاب «مشکل صنعت همگانی رایانه» مورد بررسی قرار داد.

سایت آمازون با نوسازی مرکز داده خود نقش مهمی در گسترش رایانش ابری ایفا کرد. آن‌ها دریافتند با تغییر فناوری مورد استفاده در مراکز داده خود به معماری ابر می‌توانند بازده داخلی خود را بهبود بخشند. آمازون از سال ۲۰۰۶ امکان دسترسی به سامانه خود را از طریق سرویس‌های وب آمازون و بر پایه پردازش همگانی فراهم کرد. در سال ۲۰۰۷، گوگل و آی‌بی‌ام به‌همراه چند دانشگاه، پروژه‌ای تحقیقاتی در مقیاسی بزرگ در زمینه رایانش ابری آغاز نمودند [۲].

ذخیره‌سازی حجم زیادی داده برای سازمان‌ها و کاربران خانگی مستلزم منابع سخت‌افزاریست که خود محدودیت و هزینه زیادی به‌دنبال خواهد داشت. ذخیره‌سازی ابری در مقایسه با ذخیره‌سازی سنتی دارای مزایای بسیاری است که می‌توان به موارد زیر اشاره کرد. کاربران می‌توانند از طریق اینترنت در هر زمان و در هر مکانی بدون دردسر حمل وسایل ذخیره‌سازی فیزیکی، به داده خود دسترسی داشته باشند. همچنین دیگر دغدغه تأمین و نگهداری سخت‌افزار و بازیابی اطلاعات از دست‌رفته را ندارند. علاوه‌براین کاربران دیگر نیز می‌توانند از داده‌های ذخیره‌شده در فضای ابری با

۱. Exabyte = 10^{18} Byte

اجازه مالک داده استفاده کنند.

بر اساس گزارش سیسکو، داده ذخیره شده در مراکز داده تا سال ۲۰۲۱ به ۱٫۳ زتابایت^۱ خواهد رسید، در حالی که این میزان ۲۸۶ اگزابایت در سال ۲۰۱۶ بوده است. همچنین ترافیک IP ابری جهانی به سالانه ۱۹٫۵ زتابایت تا سال ۲۰۲۱ می‌رسد [۳].

با توجه به آینده رایانش و ذخیره‌سازی ابری پرداختن به بخش‌های مختلف این حوزه از اهمیت بسیاری برخوردار است. امروزه نیز شرکت‌هایی بزرگی همچون آمازون، گوگل و مایکروسافت در ارائه هرچه بهتر سرویس‌های ابری به مشتریان خود به رقابت می‌پردازند.

۲-۱ بیان مسئله و اهداف پایان‌نامه

در بررسی عملکرد سیستم‌های ذخیره‌سازی ابری مؤلفه‌های بسیار زیادی از جمله قابلیت دسترسی به داده، قابلیت بازیابی داده، تأخیر، هزینه، امنیت و یکپارچگی داده مطرح است. در این سیستم‌ها، توزیع داده بر روی مراکز داده با روش‌های تکرار و کدگذاری صورت می‌گیرد. اگرچه روش تکرار داده روشی بسیار ساده است اما به دلیل مزایای روش کدگذاری، سیستم‌های ذخیره‌سازی ابری از هر دو روش بهره می‌برند.

کدگذاری به‌طور کلی به دو نوع کدگذاری منبع و کدگذاری کانال تقسیم می‌گردد. کدگذاری منبع با هدف بیان اطلاعات در کمترین تعداد بیت ممکن، به فشرده‌سازی اطلاعات می‌پردازد در حالی که کدگذاری کانال با اضافه کردن افزونگی به اطلاعات، آن را در برابر خطا محافظت می‌کند. در ذخیره سازی ابری با توجه به الگوی «پرداخت به‌ازای مصرف»، فشرده‌سازی اطلاعات می‌تواند به کاهش هزینه‌های کاربر کمک کند. کدگذاری کانال نیز در این سیستم‌ها با اهدافی همچون افزایش قابلیت دسترسی و قابلیت اطمینان داده و کاهش حجم داده ذخیره‌شده نسبت به حالت تکرار داده صورت می‌گیرد.

در این پایان‌نامه که شامل دو بخش کدگذاری منبع و کدگذاری کانال در سیستم‌های ذخیره

۱. Zettabyte = 10^{21} Byte

سازی ابری است، ابتدا به فشردن سازی داده با استفاده از «میزان اهمیت پیام» که در تشخیص زیرمجموعه‌های اقلیت کاربرد دارد پرداخته و سپس نرخ و افزونگی این روش را با روش‌های فشردن سازی هافمن، شانون و شانون فانو الیاس مقایسه خواهیم کرد. در ادامه کدگذاری منبع، اثر انتخاب آستانه در فشردن سازی مبتنی بر میزان اهمیت پیام^۱ (MIM) را بررسی کرده و نشان خواهیم داد که با به‌کارگیری این روش، می‌توان فضای موردنیاز برای نگهداری داده را کاهش داد و در نهایت کاربر هزینه کمتری را پرداخت خواهد کرد.

سپس در بخش کدگذاری کانال، ابتدا داده را با روش تکرار و بر اساس دو حالت تک‌منطقه‌ای و چندمنطقه‌ای روی سیستم موردنظر ذخیره می‌کنیم، در ادامه با استفاده از کدهای LT^2 و توزیع درجه‌های جدید برای این کد، داده را کدگذاری کرده و بر روی سیستم ذخیره‌سازی ابری بر اساس دو مدل تک‌منطقه‌ای و چندمنطقه‌ای توزیع خواهیم کرد. نشان می‌دهیم که با به‌کارگیری این روش، قابلیت کدگشایی موفق داده در ذخیره‌سازی ابری بهبود می‌یابد. همچنین کدگشایی با سربار کمتر انجام شده و در نتیجه به زمان کمتر برای بازیابی داده منجر خواهد شد.

در ادامه، فضای ذخیره‌سازی موردنیاز در دو روش تکرار و کدگذاری را مقایسه خواهیم کرد و سپس با ارائه روشی بر اساس خاصیت تعداد نمادهای کدگذاری موردنیاز برای بازیابی، زمان بازیابی را کاهش خواهیم داد. در نتیجه عملکرد سیستم‌های ذخیره‌سازی ابری از نظر قابلیت بازیابی داده، زمان، فضای موردنیاز برای ذخیره‌سازی و در نهایت هزینه بهبود خواهد یافت.

۱-۳ ساختار پایان‌نامه

در ساختار این پایان‌نامه، در فصل دوم به مروری بر کارهای پیشین در حوزه‌های مطرح شده پرداخته می‌شود. مفاهیم اولیه مربوط به کدگذاری منبع از جمله کدگذاری شانون، شانون فانو الیاس، هافمن و همچنین مفهوم جدید میزان اهمیت پیام، محاسبه نرخ و افزونگی در فصل سوم مطرح می‌گردد و

۱. Message Importance Measure

۲. Luby Transform

مبانی نظری مربوط به کدگذاری کانال از جمله کدهای LT به همراه بیان توزیع درجه‌های مختلف در ادامه این فصل بررسی خواهد شد. در فصل چهارم روش پیشنهادی به همراه ساختار سیستم ذخیره سازی ابری در نظر گرفته شده و نتایج حاصل از شبیه‌سازی‌ها آورده خواهد شد. نتیجه‌گیری و پیشنهاد کارهای آینده در فصل پنجم مطرح خواهد گردید.

فصل ۲ مروری بر کارهای پیشین

۲-۱ مقدمه

در این فصل به مروری بر کارهای پیشین در زمینه کدگذاری منبع و معیار جدید «میزان اهمیت پیام» خواهیم پرداخت. در ادامه فصل روند استفاده از روش‌های ذخیره‌سازی در سیستم‌های توزیع شده و ابری را توضیح خواهیم داد که شامل کارهای مرتبط با تکرار، کدگذاری پاک‌شونده، کدگذاری شبکه و کدگذاری بدون نرخ است.

۲-۲ کدگذاری منبع

امروزه فشرده‌سازی اطلاعات بخش تفکیک‌ناپذیر فناوری اطلاعات محسوب می‌شود که نقش حیاتی در ذخیره‌سازی و ارتباطات دارد. فشرده‌سازی با اهداف صرفه‌جویی در ذخیره‌سازی، استفاده کارآمد از پهنای باند و کاهش زمان ارسال صورت می‌گیرد. کلود شانون پایه‌گذار مفاهیم اساسی تئوری اطلاعات، در سال ۱۹۴۸ قضیه اساسی کدگذاری منبع و مفهوم آنتروپی را به‌عنوان حد فشرده‌سازی داده مطرح کرد [۴]. روش‌های فشرده‌سازی بسیاری از جمله کدگذاری شانون، شانون فانو الیاس، هافمن، حسابی و لمپل زیو ابداع شده و برای انواع داده توسعه یافته‌اند.

کدگذاری شانون روشی بود که در اثبات قضیه کدگذاری بدون نویز [۴] مورد استفاده قرار گرفت، این کدگذاری کدهای آنی تولید می‌کند اما کدهای شانون بهینه نیستند. کدگذاری شانون فانو الیاس [۵] که با استفاده از توزیع احتمال تجمعی کار می‌کند، یک روش پایه برای بسیاری از استانداردها از جمله JPEG و دورنگار بوده و کدگذاری حسابی نیز توسعه‌ای از این روش است. در سال ۱۹۵۲ الگوریتم کدگذاری هافمن توسط دیوید هافمن کشف شد [۶]. کدهای تولیدشده توسط این کدگذاری آنی و بهینه هستند.

۲-۳ میزان اهمیت پیام

در موارد بسیاری و با توجه به این که رویدادها با احتمال کم و نادر حاوی اطلاعات مهمی هستند می‌توان از معیار جدید میزان اهمیت پیام^۱ (MIM) برای فشرده‌سازی استفاده کرد. این معیار برای تشخیص زیرمجموعه‌های اقلیت در کلان داده^۲ مطرح شده است. تشخیص رویدادهای غیرعادی در هر زمینه‌ای که با حجم عظیمی از داده‌های علمی، پزشکی یا تجاری مواجه هستیم بسیار کاربرد دارد. به‌عنوان مثال در علم نجوم، مطالعه غیرمعمول‌ها در عکس‌های آسمان می‌تواند منجر به دستاوردهای جدید گردد [۷] یا تشخیص مسائل غیرعادی در داده‌های حسابداری و انتقالات مالی می‌تواند در جلوگیری از فعالیت‌های جعلی کمک‌کننده باشد [۸] و یا این که تنها اشخاص کمی در سیستم‌های ضدتروریست باید از نزدیک موردبررسی قرار گیرند [۹].

آنترپبی شانون و رنی معیارهای متداول برای اندازه‌گیری اطلاعات هستند. در [۱۰] مفهوم میزان اهمیت پیام با ساختاری مشابه، آنترپبی‌های شانون و رنی پیشنهاد شده است. میزان اهمیت پیام با تمرکز بر تشخیص زیرمجموعه‌های اقلیت در کلان داده، اهمیت رویدادهایی که احتمال رخ دادن آن‌ها کم است را پررنگ می‌کند. این مقاله به بیان تعریف MIM برحسب توزیع احتمال پرداخته و ویژگی‌های آن از جمله غیرمنفی بودن، اثر تجزیه و ادغام رویدادها، کران بالا و پایین MIM و مقدار آن برای توزیع یکنواخت را مطرح کرده، سپس با استفاده از MIM، روشی برای تعیین احتمالات پیشین در آزمایش فرضیه^۳ ارائه کرده است.

در [۱۱] میزان اهمیت پیام با تمرکز بر روی یک مؤلفه احتمال موردبررسی قرار گرفته و روشی عملی برای انتخاب مؤلفه پیشنهاد شده است. در این مقاله تعریف اولیه MIM با تغییر ضریب اهمیت بر اساس مؤلفه احتمال موردنظر تغییر داده می‌شود، سپس خواصی از جمله مؤلفه اصلی، قاعده زنجیره ای MIM، کران پایین برای توزیع محدود و تمرکز بر کمترین احتمال بررسی گردیده است. انتخاب

۱. Message Importance Measure

۲. Big data

۳. Hypothesis testing

مؤلفه با استفاده از احتمال پیشین از طریق بهینه‌سازی به‌دست‌آمده و در ادامه قابلیت دسترسی MIM مؤلفه‌ای در کلان داده برای تشخیص زیرمجموعه اقلیت مورد تحقیق قرار گرفته است.

میزان اهمیت پیام مؤلفه‌ای فراگیر نیست زیرا بهینه‌سازی مؤلفه‌ها نیاز به دانسته‌های قبلی دارد و منابع محاسباتی زیادی مورد نیاز است. در [۱۲] میزان اهمیت پیام غیرمؤلفه‌ای^۱ (NMIM) تعریف شده است. این تعریف نیز همانند آنتروپی شانون تنها به توزیع احتمال وابسته است. در این مقاله خواص یکنواختی، تجزیه و ادغام رویدادها، مرکز هندسی، کران پایین NMIM، توزیع‌های احتمال وابسته، بسط تیلور و احتمال کمینه مطرح و به اثبات رسیده است. سپس مدلی از ذخیره‌سازی با NMIM فشرده می‌گردد به این صورت که مسئله‌ای برای کاهش اتلاف اهمیت NMIM بهینه‌سازی می‌گردد. با توجه به این که خطای بازسازی نسبت به طول کد به‌طور یکنواخت کاهش می‌یابد، در مسئله بهینه‌سازی دو مدل خطای بازسازی معکوس و نمایی به‌همراه NMIM به‌عنوان عامل وزن‌دهی در نظر گرفته شده است. بنابر نتایج، استفاده از بخش کوچکی از ذخیره‌سازی در حالی که مهم‌ترین اطلاعات حفظ می‌گردند امکان‌پذیر است. در ادامه مقاله، تغییر NMIM در حین ارسال پیام و تعریف تابع اعوجاج اتلاف NMIM به‌منظور توصیف مصالحه بین اتلاف NMIM و اعوجاج بیان گردیده است. نواحی رشد و اشباع برای مسئله بیشینه ارسال به‌دست‌آمده است که می‌تواند به بهبود طراحی سیستم‌های ارتباطی عملی کمک کند.

در [۱۳] دیورژانس اهمیت پیام^۲ (MID) برای دو احتمال گسسته تعریف می‌گردد که می‌توان از این مفهوم برای اندازه‌گیری تغییرات آنها به انتها در یک فرایند انتقال اطلاعات استفاده کرد. سپس ظرفیت انتقال اهمیت پیام که توسط MID اندازه‌گیری می‌شود تعریف شده و انتقال اطلاعات روی کانال متقارن باینری^۳ (BSC) مورد بررسی قرار گرفته است. کاربرد این مفهوم در پردازش در لبه شبکه

۱. Non-parametric Message Importance Measure

۲. Message Importance Divergence

۳. Binary Symmetric Channel

موبایل^۱ (MEC) با در نظر گرفتن صف مطرح شده و نشان داده شده است که درمقایسه با دیورژانس KL^۲، می‌تواند کران پایین‌تری برای انتخاب اندازه ذخیره‌سازی برای MEC فراهم آورد. همچنین MID در حالت پیوسته توسعه داده‌شده تا همبستگی اهمیت پیام در فرایند انتقال اطلاعات موردتحقیق قرار گیرد.

در [۱۴]، میزان اهمیت پیام دیفرانسیلی^۳ (DMIM) به‌عنوان میزانی از اطلاعات برای متغیر تصادفی پیوسته پیشنهاد شده است که فرایند جمع‌آوری اطلاعات را توصیف می‌کند. همچنین DMIM ارتباط بین واریانس یک متغیر تصادفی و تخمین خطای توزیع تجربی را نشان می‌دهد. DMIM توزیع‌های نرمال، یکنواخت و نمایی منفی و همچنین خواص آن موردبررسی قرار گرفته است. در این مقاله ثابت می‌شود که دیورژانس بین مدل تجربی توزیع و مدل واقعی آن توسط انحراف DMIM کنترل می‌گردد.

۲-۴ انواع روش‌های ذخیره‌سازی ابری

به‌طورکلی به‌منظور تضمین قابلیت اطمینان داده، افزونگی به داده اضافه می‌شود. برای ذخیره‌سازی داده در سیستم‌های ذخیره‌سازی توزیع‌شده و ابری چهار روش مطرح شده است که عبارت‌اند از ذخیره‌سازی بر پایه تکرار^۴، ذخیره‌سازی بر پایه کدهای پاک‌شونده^۵، ذخیره‌سازی بر پایه کدگذاری شبکه^۶ و در سال‌های اخیر ذخیره‌سازی بر پایه کدهای بدون نرخ^۷ موردبررسی قرار گرفته است.

۲-۴-۱ ذخیره‌سازی بر پایه تکرار

به‌منظور محافظت از داده در برابر خرابی‌ها، امروزه فراهم‌آورندگان ذخیره‌سازی ابری به‌سادگی

۱. Mobile Edge Computing

۲. Kullback Leibler

۳. Differential Message Importance Measure

۴. Replication-based cloud storage

۵. Erasure codes-based cloud storage

۶. Network coding-based cloud storage

۷. Rate less codes-based cloud storage

محتوا را درون شبکه ذخیره‌سازی روی چندین مرکز داده تکرار می‌کنند. علاوه بر تحمل خطا، تکرار باعث می‌گردد محتوا به سرعت قابل‌دستیابی باشد؛ از آنجایی که چندین کاربر یک محتوا را درخواست می‌کنند و یا درخواست کاربر از نقاط مختلف جغرافیایی صورت می‌گیرد، کاربر می‌تواند به رونوشت‌های مختلف در دسترس هدایت شود. ذخیره‌سازی بر پایه تکرار ساده‌ترین روش توزیع داده روی سیستم‌های ذخیره‌سازی ابری است که با توجه به نوع ذخیره‌سازی انتخاب‌شده از سوی کاربر، با یک ضریب تکرار انجام می‌شود.

اگرچه تکرار در ابعاد وسیع دارای توانایی افزایش قابلیت دسترسی و ماندگاری است، دو چالش بزرگ به معماران سیستم معرفی می‌کند. اولین چالش این است که معماران سیستم باید تعداد تکرارها را به‌منظور دستیابی به ماندگاری بالا برای سیستم‌های بزرگ افزایش دهند. چالش دیگر این است که افزایش تعداد تکرارها، پهنای باند و ملزومات ذخیره‌سازی سیستم را زیاد می‌کند [۱۵]. استفاده بیشتر از فضای ذخیره‌سازی بر میزان مصرف برق، هزینه نگهداری و تعمیر بیشتر در مراکز داده دلالت دارد.

از دیگر معایب روش تکرار این است که در سرویس‌های توزیع‌شده که تعداد زیادی از کاربران نیاز به دسترسی همزمان دارند، نمی‌تواند کارایی بالا موردنیاز را برآورده کند. همچنین برای بازیابی داده باید به مراکز داده معینی دسترسی پیدا کند درحالی‌که در سیستم‌های بر پایه کدگذاری دریافت تعداد کافی بسته از مراکز در دسترس، کفایت می‌کند.

۲-۴-۲ ذخیره‌سازی بر پایه کدگذاری

کدگذاری که در اصل به‌منظور ارتباطات قابل‌اطمینان در حضور نویز توسعه یافت، یک روش کارآمدتر برای ذخیره داده در سیستم‌های توزیع‌شده ارائه می‌کند. ایده اصلی پشت کدگذاری این است که با اضافه کردن افزونگی، محتوایی که در مجموعه‌ای از دیسک‌ها ذخیره شده است با خواندن زیرمجموعه‌ای از این دیسک‌ها بتواند بازسازی شود.

با سطح افزونگی یکسان، سیستم ذخیره‌سازی بر پایه کدهای پاک‌شونده ده‌ها برابر قابل‌اطمینان‌تر از سیستم بر پایه تکرار است. مقاله [۱۵] به مقایسه کمی دو روش تکرار و کد پاک‌شونده در سیستم‌های نظیر به نظیر پرداخته است. قابلیت دسترسی برای سیستم بر پایه کد پاک‌شونده را به دست آورده و نشان می‌دهد که با میانگین زمان خرابی^۱ (MTTF) مشابه، این سیستم دارای ذخیره‌سازی و پهنای باند بسیار کمتری نسبت به سیستم بر پایه تکرار است. همچنین MTTF یک سیستم با کد پاک‌شونده، ده‌ها برابر بزرگ‌تر از MTTF یک سیستم با تکرار است در شرایطی که هر دو سربار ذخیره‌سازی و دوره اصلاح یکسانی دارند.

با استفاده از تقسیم‌بندی محتوا و کدگذاری آن‌ها می‌توان زمان بارگیری را به‌طور چشم‌گیری کاهش داد. در [۱۶] دو مدل آبخاری و صف انشعاب-اتصال^۲ برای دستیابی به محتوا ارائه شده است که در مدل اول زمان بارگیری به‌صورت صریح به دست آمده است اما در مدل دوم با استفاده از کران پایین و بالا بیان گردیده است. با به‌کارگیری کدهای ماکزیمم فاصله تفکیک‌پذیر^۳ (MDS)، زمان بارگیری محتوا بسیار کاهش یافته است.

در [۱۷] نشان داده شده است که با استفاده از کدگذاری علاوه‌بر فراهم آوردن قابلیت اطمینان در برابر خطا، می‌توان زمان بارگیری مورد انتظار را نیز کاهش داد به این دلیل که خواندن تنها زیرمجموعه‌ای از محتوای کدگذاری شده کفایت می‌کند. یک سیستم انشعاب-اتصال در تئوری صف پیشنهاد شده و کران‌های آن بر روی میانگین زمان پاسخگویی مورد بررسی قرار گرفته است. تحلیل زمان بارگیری محتوا به مصالحه‌ای بین زمان بارگیری و فضای ذخیره‌سازی منجر شده است که می‌تواند در طراحی سطح بهینه افزونگی در یک سیستم ذخیره‌سازی توزیع‌شده به کار رود.

در [۱۸] بهره‌وری انرژی سیستم‌های ذخیره‌سازی توزیع‌شده ناهمگون بر پایه کدهای پاک‌شونده مورد بررسی قرار گرفته است، به این صورت که داده‌ها در رده‌های مجزا دسته‌بندی می‌شوند و نشان

۱. Mean Time to Failure

۲. Fork-join

۳. Maximum Distance Separable Codes

داده شده است که بهره‌وری انرژی ارتباط نزدیکی با میانگین تأخیر دارد. به دلیل تحلیل پیچیده صف انشعاب-اتصال، کران پایین میانگین تأخیر با در نظر گرفتن صف M/M/1 به دست آمده و کران بالا نیز با استفاده از صف M/G/1 در سیستم ترک-ادغام^۱ حاصل شده است. بهره‌وری انرژی این سیستم تحت سیاست‌های زمان‌بندی گوناگون و عوامل مختلف مورد تحقیق قرار گرفته است.

به طور معمول کدهای رید-سالمون^۲ (R-S) [۱۹] برای کدگذاری پاک‌شونده در سیستم‌های ذخیره سازی ابری به کار می‌روند [۲۰, ۲۱]. در [۲۱] کدهای بازسازی محلی^۳ (LRC) برای ذخیره‌سازی WAS^۴ پیشنهاد شده است که این کدها علاوه بر تولید توازن‌های سراسری مانند کدهای R-S، از توازن‌های محلی نیز بهره می‌برند. کدهای LRC به منظور کاهش سربار و تعداد عملیات ورودی/خروجی برای بازسازی معرفی شده‌اند. تعداد کمتر قطعات مورد نیاز برای بازسازی، به کاهش زمان و هزینه بازسازی منجر شده است.

برای یک سیستم مبتنی بر کدهای پاک‌شونده در صورت خرابی تک‌گره باید کل داده اصلی بازسازی شود تا تنها یک بسته کدگذاری شده تولید گردد؛ این فرایند بهینه نیست و هزینه ارتباطات بالایی به دنبال دارد. با توجه به حجم زیاد داده برون‌سپاری شده در سیستم‌های توزیع‌شده، بازسازی کل داده گران است و از جذابیت این روش می‌کاهد.

کدگذاری شبکه تعمیمی از مسیریابی سنتی ست که در آن هر گره میانی در شبکه اطلاعات دریافتی را به سادگی ذخیره و ارسال می‌کند در حالی که کدگذاری شبکه به گره‌های میانی اجازه می‌دهد تا داده خروجی را با استفاده از کدگذاری داده دریافتی تولید کند.

در [۲۲] به حل مشکل اصلاح در سیستم‌های توزیع‌شده با استفاده از کدگذاری شبکه پرداخته است. کدهای بازتولید^۵ به یک گره جدید اجازه می‌دهند تا با توابعی از داده‌های ذخیره‌شده در گره

۱. Split-Merge

۲. Reed-Solomon

۳. Local Reconstruction Codes

۴. Windows Azure Storage

۵. Regenerating Codes

های باقیمانده ارتباط برقرار کنند. در این مقاله نشان داده شده است که این کدها می‌توانند هزینه ارتباطات اصلاح را به کمینه نظریه اطلاعات برسانند. همچنین مصالحه بین پهنای باند اصلاح و ذخیره سازی با بررسی دو نوع کد بازتولید کمینه ذخیره‌سازی و کمینه پهنای باند مشخص شده است. استفاده از اصلاح تابعی^۱ که داده نه به شکل کاملاً دقیق بلکه به صورت درست بازیابی می‌شود، منجر به چندپخشی روی یک گراف جریان اطلاعات می‌گردد. گراف جریان اطلاعات، سیر اطلاعات را در هنگام اتصال یا ترک گره‌ها از شبکه نشان می‌دهد. همچنین در اصلاح تابعی نیاز است تا کد به شکل سازمان‌یافته باشد، بنابراین از نظر عملی زیاد موردعلاقه نیست. در [۲۳] از اصلاح دقیق به جای اصلاح تابعی استفاده شده و تحلیل برش در گراف جریان اطلاعات موردبررسی قرار گرفته است؛ بنابراین با استفاده از اصلاح دقیق، مسئله چندپخشی یک منبع به تعداد نامحدود گیرنده روی یک گراف بی‌کران برطرف شده و به منظور بررسی مصالحه بهینه بین بهترین بهره‌وری ذخیره‌سازی و کمینه پهنای باند اصلاح، مشابه مقاله قبلی دو نوع کد بازتولید کمینه ذخیره‌سازی و کمینه پهنای باند با اصلاح دقیق موردبررسی قرار گرفته است.

با این وجود، کدگذاری شبکه به علت به کار بردن روش حذف گوسی برای کدگشایی، بسیار گران‌تر از روش کدگذاری پاک‌شونده از نظر هزینه محاسباتی است. سیستم‌های ذخیره‌سازی توزیع‌شده بر اساس کدهای پاک‌شونده و کدگذاری شبکه دارای هزینه بالای محاسبات کدگشایی برای کاربران داده هستند و مسئولیت اصلاح داده و برخط بودن مالکان داده را نیز به دنبال دارند.

در [۲۴] یک معماری ذخیره‌سازی پیشنهاد شده است که از ترکیب کدهای $LT[۲۵]$ و سازوکارهای دستیابی نظری برای نوشتن و خواندن موازی در محیط‌های توزیع‌شده بهره می‌برد. استفاده از کدهای نزدیک به بهینه LT انعطاف‌پذیری بالاتری در خواندن داده نسبت به تکرار فراهم می‌آورد. عملکرد بالا و پایدار این سیستم نسبت به سیستم‌های ذخیره‌سازی موازی سنتی با نتایج شبیه‌سازی به تأیید رسیده است.

۱. Functional repair

در [۲۶] یک سرویس ذخیره‌سازی ابری امن بر پایه کدهای LT با توجه کارآمد به بازیابی و اصلاح داده طراحی شده که به دلیل استفاده از کدگشایی انتشار باور^۱ (BP) در طول بازیابی داده، دارای کدگشایی سریع‌تری نسبت به راهکارهای موجود است. در این سیستم امن و قابل اطمینان علاوه بر بردار کدگذاری، برچسب‌های بازیابی و بازبینی نیز به ابتدای بسته‌های کدگذاری شده اضافه می‌شوند و یک سرور مجزا برای بررسی یکپارچگی استفاده می‌گردد. با وجود اینکه تعداد بسته‌های کدگذاری در سیستم‌های ذخیره‌سازی ابری بر پایه کدهای LT بسیار بزرگ‌تر از سرویس‌های ذخیره سازی مبتنی بر کدهای پاک‌شونده و مبتنی بر کدگذاری شبکه است، هزینه محاسبات کدگذاری بسیار کمی دارد. علت این مزیت بزرگ استفاده از عملیات XOR در کدگذاری است در حالی که ترکیب های خطی در کدگذاری شبکه و کدگذاری بر پایه R-S، درگیر عملیات ضرب همراه با ضرایب هستند. هزینه ذخیره‌سازی در این سیستم بیشتر از سیستم مبتنی بر R-S است زیرا کدهای LT، به لحاظ مصالحه افزونگی-قابلیت اطمینان نزدیک به بهینه‌اند. اگرچه تعداد کل بسته‌های اصلی در دو سیستم پیشین کمتر از سیستم جدید است اما فرایند کدگشایی برای کاربر داده حداقل دو برابر سریع‌تر است. این فرایند کدگشایی در مقایسه با سایر سرویس‌های ذخیره‌سازی مبتنی بر افزونگی بسیار مناسب‌تر است زیرا در الگوی ذخیره‌سازی ابری، بازیابی داده یک روال عادی محسوب می‌شود.

در [۲۷]، یک معماری برای خادم^۲ یک سیستم ذخیره‌سازی در سطح بلوکی^۳ (BLCS) ارائه شده است که قادر است به سطوح مختلف از عملکرد، قابلیت دسترسی، یکپارچگی و امنیت با استفاده از کدهای LT دست یابد. کدهای LT با نیازهای سیستم BLCS مطابقت دارد زیرا دارای پیچیدگی کمتر و قابلیت تنظیم افزونگی در زمان اجرا است. استفاده از این کدها منجر به زمان دسترسی کمتر به بخش‌ها و کارایی انتقال بالا در حین اجرا می‌گردد. قطعات کدشده روی منابع ذخیره‌سازی مختلف مستقل ذخیره می‌گردند. قطعات گم‌شده می‌توانند دوباره بدون محدودیت تولید گردند. سطوح

۱. Belief Propagation

۲. Back-end

۳. Block Level Cloud Storage

مناسب محرمانگی می‌تواند توسط پنهان نگه‌داشتن فرایند تصادفی تولید و آدرس‌های منابع ذخیره سازی حاصل گردد. در این سیستم به دلیل استفاده از تعداد نمادهای اصلی کوچک، روش کدگشایی حذف گوسی در حین اجرا^۱ افزایشی برای حذف اثرات نامطلوب سر بار زیاد و احتمال کدگشایی ناموفق به کار گرفته شده است. برای اهداف مقیاس‌پذیری و به منظور دستیابی به یک توازن بار خوب، گره های ذخیره‌سازی در ساختار سلسه‌مراتبی خوشه‌بندی شده‌اند.

برخلاف کدهای MDS از جمله کد R-S که به تعداد قطعی بسته‌های کدشده اضافی برای کدگشایی نیاز دارند، در سیستم ذخیره‌سازی بر پایه LT، کدگشا به تعداد نامعینی از قطعه‌های کد گذاری شده برای کدگشایی موفق نیاز دارد. این تعداد نامعین از بسته‌های کدگذاری شده، تأخیری در بازیابی اضافه می‌کند؛ بنابراین یک مصالحه بین احتمال کدگشایی موفق و تأخیر بازیابی مطرح است. در [۲۸] نشان داده شده است که بازیابی چندمرحله‌ای قطعه‌ها می‌تواند در کاهش تأخیر بازیابی فایل مؤثر باشد. در مقایسه با سیستم‌های ذخیره‌سازی ابری بر پایه کدهای پاک‌شونده، سیستم‌های ذخیره سازی ابری بر پایه کدهای LT جای‌گذاری محتوا و مشکلات بازیابی را به قیمت تأخیر بازیابی طولانی تر، تسهیل می‌کنند. با ارائه یک طرح درخواست چندمرحله‌ای و حل مسئله بهینه‌سازی برای تأخیر بازیابی با در نظر گرفتن صف M/G/1، مشاهده می‌شود که طرح درخواست دومرحله‌ای، میانگین تأخیر را به طور چشم‌گیری کاهش می‌دهد. بر اساس مصالحه تأخیر-قابلیت کدگشایی گزارش شده، کاربران می‌توانند برای کاربردهایی حساس به تأخیر مانند دنباله ویدئویی، یک احتمال کدگشایی کمتر به کار گرفته تا در آن کاربرد متوسط تأخیر بازیابی زیر یک آستانه معین نگه داشته شود.

۱. On The Fly Gaussian Elimination

فصل ۳ مفاهیم پایه

۳-۱ مقدمه

در این فصل ابتدا به بیان ساختار کلی سیستم‌های ذخیره‌سازی ابری و مدل‌های مختلف ذخیره‌سازی در این سیستم‌ها خواهیم پرداخت. در ادامه، مفاهیم و روابط کدگذاری منبع از جمله روابط MIM را بیان کرده و مروری بر روش‌های فشرده‌سازی شانون، شانون فانو الیاس و هافمن خواهیم داشت. سپس به مبانی کدگذاری کانال با معرفی کدهای LT و انواع توزیع درجه برای آن‌ها خواهیم پرداخت.

۳-۲ سیستم ذخیره‌سازی ابری

زیر ساختار ابری شرکت‌های بزرگی مانند آمازون، گوگل و مایکروسافت از مناطق و نواحی دسترسی تشکیل شده است. منطقه^۱، یک موقعیت فیزیکی در دنیا است که دارای چندین ناحیه دسترسی^۲ است. نواحی دسترسی از یک یا چند مرکز داده مجزا تشکیل شده‌اند که هر کدام با تغذیه، شبکه و ارتباطات در امکانات جداگانه قرار گرفته‌اند. نواحی دسترسی به صورت مجزا و مستقل از یکدیگر طراحی شده‌اند تا بروز خطا و خرابی در یکی از نواحی، اثری بر عملکرد سایر نواحی نداشته باشد. نواحی دسترسی توسط شبکه فیبر نوری خصوصی سریع با یکدیگر در ارتباط هستند. به عنوان مثال، ابر سرویس وب آمازون^۳ (AWS) ۵۴ ناحیه دسترسی درون ۱۸ منطقه جغرافیایی را در اطراف دنیا مدیریت می‌کند [۲۹]. در AWS، ارتباط بین نواحی دسترسی از طریق فناوری تقسیم چندگانه طول موج متراکم^۴ (DWDM) صورت می‌گیرد. هر ناحیه دسترسی یک یا چند مرکز داده دارد، برخی از نواحی دسترسی آمازون تا ۶ مرکز داده نیز دارند که هر یک دارای بیش از ۵۰۰۰۰ سرور هستند. ابر آزور^۵ با بیشترین مناطق جهانی دارای ۵۰ منطقه و ۱۴۰ ناحیه دسترسی است.

به منظور دستیابی به سرویس‌های مقاوم در برابر خطا با قابلیت دسترسی بالا، توصیه می‌شود که

۱. Region

۲. Availability Zone

۳. Amazon Web Service

۴. Dense Wavelength Division Multiplexing

۵. Azure

داده روی چندین ناحیه دسترسی در یک منطقه یا چند منطقه توزیع گردد. هر یک از سرویس دهندگان فضای ذخیره‌سازی ابری بنابر معیارهایی از جمله قابلیت دسترسی، نرخ دستیابی به داده، تأخیر و هزینه، قابلیت‌ها و امکانات متفاوتی در اختیار کاربران قرار می‌دهند. در نهایت، کاربر از بین انتخاب‌هایی که سرویس‌دهندگان ابری در اختیار او قرار می‌دهند، سرویسی را متناسب با کاربرد و هزینه موردنظر خود انتخاب می‌نماید.

بین انواع سرویس‌های ذخیره‌سازی که این شرکت‌ها پیشنهاد می‌دهند دو نوع تک‌منطقه‌ای و چندمنطقه‌ای مشترک بوده و به‌طور گسترده مورد استفاده قرار می‌گیرد.

۳-۲-۱ ذخیره‌سازی تک‌منطقه‌ای

این نوع ذخیره‌سازی کاربر را قادر می‌سازد تا داده خود را با هزینه پایین‌تر ذخیره نماید در حالی که مصالحه نگهداری داده در یک منطقه مشخص به‌جای توزیع آن روی یک سطح جغرافیایی بزرگ را می‌پذیرد. در حالت تکرار در ذخیره‌سازی تک‌منطقه‌ای، سه رونوشت از داده روی سه ناحیه درون یک منطقه توزیع می‌گردد.

۳-۲-۲ ذخیره‌سازی چندمنطقه‌ای

این نوع ذخیره‌سازی برای داده‌هایی که اغلب در دسترس هستند مطلوب است از جمله خدمات‌دهی به محتوا وب‌گاه، کارهای تعاملی یا پشتیبانی داده‌های مرتبط با برنامه‌های کاربردی تلفن همراه و بازی. ذخیره‌سازی چندمنطقه‌ای بیشترین قابلیت دسترسی را در میان انواع ذخیره‌سازی دارد. این نوع ذخیره‌سازی، داده کاربر را در حداقل دو منطقه جداگانه با حداقل ۱۰۰ مایل فاصله از یکدیگر ذخیره می‌کند. این امر بیشینه دسترسی را حتی در صورت بروز اختلالات در مقیاس بزرگ مانند بلایای طبیعی به‌دنبال خواهد داشت.

علاوه بر این، شرکت گوگل دو نوع ذخیره‌سازی کم‌هزینه دیگر را به کاربران خود ارائه می‌دهد:

ذخیره‌سازی برای داده‌هایی که به‌طور متوسط یک‌بار در ماه یا کمتر به آن‌ها مراجعه می‌شود^۱ و نوعی دیگر یک سرویس ذخیره‌سازی بسیار ارزان برای داده‌هایی است که کاربر قصد دارد یک‌بار در سال به آن‌ها دسترسی یابد^۲ [۳۰].

ذخیره‌سازی آזור شرکت مایکروسافت علاوه‌بر دو نوع ذخیره‌سازی تک‌منطقه‌ای و چندمنطقه‌ای، ذخیره‌سازی افزونه محلی که داده را سه‌بار درون یک مرکز داده تکرار می‌کند نیز ارائه می‌دهد [۳۱].

۳-۲-۳ مبانی صف M/G/1

هر یک از مراکز داده دارای یک صف هستند که درخواست‌های بازیابی داده از سوی کاربران در آن قرار می‌گیرد. از آنجایی که در بسیاری از طرح‌های عملی توزیع سرویس‌دهی ناشناخته است، صف M/G/1 برای هر یک از مراکز داده در نظر گرفته می‌شود [۱۷].

در صف M/G/1، نرخ ورود به صف از توزیع نمایی با میانگین λ تبعیت می‌کند. توزیع سرویس‌دهی در این صف با میانگین $E[X]$ و واریانس $\sigma^2 [X]$ توصیف شده و زمان انتظار در صف M/G/1 با رابطه پولاک-کنچین^۳ به‌صورت زیر بیان می‌گردد [۳۲].

$$W = \frac{\lambda(\sigma_x^2 + \bar{x}^2)}{2(1 - \lambda \bar{x})} \quad (1-3)$$

۳-۳ کدگذاری منبع

همان‌طور که در فصل قبل بیان شد، کدگذاری منبع یا فشرده‌سازی با اهداف صرفه‌جویی در ذخیره‌سازی، استفاده کارآمد از پهنای باند و کاهش زمان ارسال صورت می‌گیرد. از مفهوم میزان اهمیت پیام با ساختاری مشابه آنتروپی‌های شانون و رنی در فشرده‌سازی استفاده خواهیم کرد. از فشرده

۱. Nearline

۲. Coldline

۳. Pollaczek-Khinchine

سازی بر اساس MIM در کاربردهایی که بخش‌های مهم پیام مدنظر است می‌توان بهره برد. درحقیقت نیز بسیاری از افراد به بخش‌های مهم پیام نه کل پیام اهمیت می‌دهند. به‌عنوان مثال برای فشرده سازی یک متن با استفاده از MIM به دلیل حذف نمادها با احتمال بالا، متن ناخوانا می‌شود اما در مواردی به دلیل محدودیت، ضرورت و یا انتخاب کاربر، می‌توان تنها عناوین، اعداد، بخش‌های پررنگ و مورب متن را نمایش داد. همچنین موارد دیگری که فقط رویدادها با احتمال پایین در یک فایل مد نظر هستند.

با توجه به «الگوی پرداخت به‌ازای مصرف» در سیستم‌های ذخیره‌سازی ابری این روش می‌تواند در کاهش هزینه کاربر کارآمد باشد.

۳-۳-۱ کدگذاری مبتنی بر میزان اهمیت پیام

در این بخش به بیان تعاریف میزان اهمیت پیام مؤلفه‌ای، میزان اهمیت پیام غیرمؤلفه‌ای، آنتروپی بر اساس MIM، نرخ بر اساس MIM می‌پردازیم. همچنین مفاهیم کدگذاری شانون، شانون فانو الیاس و هافمن که برای مقایسه با فشرده‌سازی بر اساس MIM به کاررفته‌اند، یادآوری می‌شوند.

تعریفی که از این معیار ارائه می‌گردد برخلاف آنتروپی شانون که از عملگر لگاریتم و آنتروپی رنی که از عملگر چندجمله‌ای استفاده می‌کند، از صورت نمایی بهره می‌برد زیرا عملگر نمایی می‌تواند به بزرگ‌نمایی زیرمجموعه اقلیت کمک می‌کند.

تعریف ۳-۱. میزان اهمیت پیام: برای یک توزیع احتمال معین $p = (p_1, p_2, \dots, p_n)$ با الفبای

محدود، میزان اهمیت پیام با مؤلفه ϖ به صورت زیر تعریف می‌گردد

$$L_j(p, \varpi) = \log\left(\sum_{i=1}^n p_i e^{\varpi(1-p_i)}\right) \quad (2-3)$$

که در آن ϖ ضریب اهمیت است [۱۰].

تعریف ۳-۲. میزان اهمیت پیام مؤلفه‌ای: در MIM مؤلفه‌ای که بر روی یک مؤلفه احتمال تمرکز

دارد، ضریب اهمیت $\varpi = \frac{1}{p_j}$ است و تعریف MIM به صورت زیر بیان می شود [۱۱]

$$L_j(p, \varpi_j) = \log\left(\sum_{i=1}^n p_i e^{\frac{1}{p_j}(1-p_i)}\right) \quad (۳-۳)$$

تعریف ۳-۳. میزان اهمیت پیام غیر مؤلفه‌ای: برای یک توزیع احتمال معین $p = (p_1, p_2, \dots, p_n)$ با الفبای محدود، MIM غیر مؤلفه‌ای به صورت زیر تعریف می گردد

$$\mathcal{L}(p) = \log \sum_{i=1}^n p_i e^{\frac{1-p_i}{p_i}} \quad (۴-۳)$$

تعریف ۳-۴. اهمیت یک رویداد با احتمال p_i با رابطه زیر به دست می آید

$$M(p_i) = p_i e^{\frac{1-p_i}{p_i}} \quad (۵-۳)$$

بنابراین رابطه NMIM نیز به صورت زیر بازنویسی می شود [۱۲]

$$\mathcal{L}(p) = \log \sum_{i=1}^n M(p_i) \quad (۶-۳)$$

تعریف ۳-۵. آنتروپی اهمیتی: آنتروپی اهمیتی که با H^* نمایش داده خواهد شد به صورت زیر تعریف می گردد

$$H^*(\chi) = \log\left(-\sum_{i=1}^n (p_i e^{\frac{1-p_i}{p_i}}) \log_2(p_i)\right) \quad (۷-۳)$$

تعریف ۳-۶. نرخ اهمیتی: نرخ اهمیتی که با R^* نمایش داده خواهد شد به صورت زیر تعریف می شود

$$R^*(\chi) = \log\left(\sum_{i=1}^n (p_i e^{\frac{1-p_i}{p_i}}) n_i\right) \quad (۸-۳)$$

که در آن n_i طول نماد کدگذاری متناظر با p_i است.

۲-۳-۳ کدگذاری شانون

شانون یک سیستم کدگذاری معرفی کرد که طول کد را بر اساس احتمال آن به دست آورده و میانگین رقم باینری برای هر نماد برابر با آنتروپی آن منبع است [۴]. کدهای شانون کدهای آنی^۱ هستند به این معنا که هیچ کلمه کدی پیشوند کلمه کد دیگر نیست. اگر احتمال رخداد هر نماد p_i باشد آنگاه طول کد برابر است با

$$l_i = \lceil -\log p_i \rceil \quad (۹-۳)$$

برای کدگذاری شانون از تابع توزیع تجمعی استفاده می شود و کدها از بسط باینری تابع توزیع تجمعی با توجه به طول کد به دست می آیند. کدهای شانون بهینه نیستند.

۳-۳-۳ کدگذاری شانون فانو الیاس

در این کدگذاری نیز از تابع توزیع تجمعی برای اختصاص کد کلمه ها استفاده می گردد. برای نمادهای منبع، تابع توزیع اصلاح شده به صورت زیر بیان می شود

$$\bar{F}(x) = \sum_{a \leq x} p(a) + \frac{1}{2} p(x) \quad (۱۰-۳)$$

به طور کلی $\bar{F}(x)$ یک عدد حقیقی است که تنها به وسیله تعداد بی نهایت بیت قابل بیان است. از آنجایی که استفاده از مقدار دقیق $\bar{F}(x)$ کارآمد نیست، طول کد کلمه ها با استفاده از رابطه زیر به دست می آید

$$l_i = \lceil -\log p_i \rceil + 1 \quad (۱۱-۳)$$

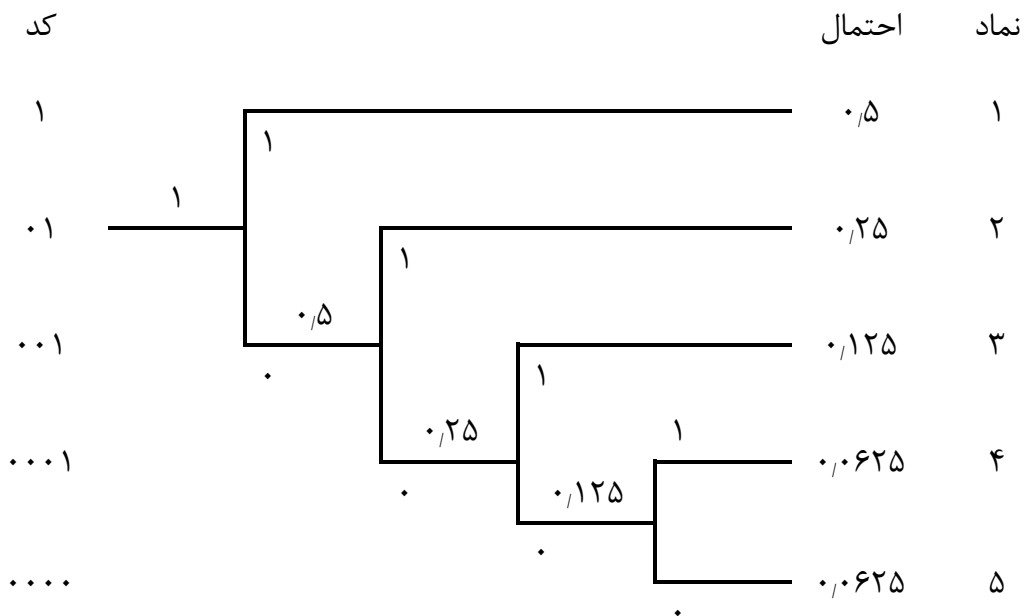
کدگذاری شانون فانو الیاس با استفاده از بسط باینری $\bar{F}(x)$ حاصل می شود [۵].

۴-۳-۳ کدگذاری هافمن

یک کد بهینه آنی برای یک توزیع معین می تواند توسط یک الگوریتم ساده که توسط هافمن کشف

۱. Instantaneous codes or prefix-free codes

شد ساخته شود. کدگذاری هافمن کمترین طول مورد انتظار کد را در بین روش‌های کدگذاری دارد. روش کدگذاری هافمن را با ارائه مثالی یادآوری می‌کنیم [۳۳].



۳-۴ کدگذاری کانال

کانال پاک‌شونده باینری^۱ (BEC) توسط الیاس در ۱۹۵۵ معرفی شد اما تا زمان به‌کارگیری اینترنت در مقیاس گسترده در حدود ۴۰ سال اخیر، بیشتر به‌عنوان یک مدل کانال نظری در نظر گرفته می‌شد. اینترنت یک مدل بسیار خوب واقعی از BEC است.

هنگامی که داده روی شبکه بر پایه IP ارسال می‌گردد، به‌طور معمول به بخش‌هایی با طول یکسان تقسیم شده و در بسته‌ها قرار می‌گیرد. هر پروتکل انتقالی باید دریافت تمام بسته‌ها توسط گیرنده را تضمین دهد. به این دلیل است که به‌عنوان مثال گیرنده TCP هر بسته دریافت شده را تصدیق می‌کند و یک فرستنده TCP بسته‌های تصدیق نشده را دوباره ارسال می‌کند. این روش‌ها در بسیاری از موارد رفتار ضعیفی از خود نشان می‌دهند، از جمله ارسال داده از یک سرور به چندین گیرنده، ارسال داده روی کانال‌های به‌شدت ضعیف از جمله لینک‌های ضعیف بی‌سیم یا ماهواره، ارسال روی شبکه‌هایی با

۱. Binary Erasure Channel

تاخیرهای زیاد و همچنین شبکه‌هایی با نرخ اتلاف بالا. علاوه بر این، پروتکل‌های بر پایه تصدیق مانند TCP هنگامی که فاصله بین فرستنده و گیرنده زیاد باشد به صورت ضعیف عمل می‌کنند. به این دلایل راه‌حل‌های ارسال دیگری از جمله کدگذاری پیشنهاد شده‌اند.

در کدگذاری، داده اصلی با استفاده از برخی کدهای تصحیح خطا پاک‌شونده خطی کدگذاری می‌شود. اگر در حین ارسال قسمتی از داده گم شود آنگاه این امکان وجود دارد که داده گم‌شده با استفاده از الگوریتم‌های تصحیح پاک‌شونده بازیابی گردد. کدهای پاک‌شونده به منظور فراهم آوردن قابلیت اطمینان برای انواع کاربردهای تحویل داده مطرح شده‌اند. برای کاربردها بسیار مهم است که کدهای مورد استفاده قادر به تصحیح بسیاری از پاک‌شدگی‌ها تا حد امکان باشند و همچنین سرعت بالای الگوریتم‌های کدگذاری و کدگشایی برای این کدها حیاتی است.

زمان‌های کدگذاری و کدگشایی روش‌های کدگذاری مرسوم برای طراحی سیستم‌های ارسال داده مقیاس‌پذیر کفایت نمی‌کنند. علاوه بر این، کانال‌های پاک‌شونده از یک فرستنده به چندین گیرنده، بالقوه دارای احتمالات پاک‌شدگی متفاوت هستند. به طور معمول در کاربردها، فرستنده یا گیرنده ممکن است کانال‌های خود را بررسی کنند بنابراین فرستنده دارای یک حدس منطقی از احتمال پاک‌شدگی جاری کانال بوده و می‌تواند بر اساس آن نرخ کدگذاری را تنظیم کند؛ اما اگر تعداد گیرنده‌ها زیاد باشد، یا در موقعیت‌هایی از جمله ارسال ماهواره یا بی‌سیم که گیرنده‌ها تغییرات ناگهانی در مشخصات دریافت خود تجربه می‌کنند، فرض کردن و ردیابی نرخ‌های تلفات گیرنده‌ها به صورت تکی غیرواقعی می‌شود. آنگاه گیرنده مجبور است بدترین حالت نرخ اتلاف را برای تمامی گیرنده‌ها در نظر بگیرد. این نه تنها اگر نرخ اتلاف واقعی کوچک‌تر باشد بار غیرضروری به شبکه اعمال می‌کند، بلکه همچنین با ارسال مطمئن سازش می‌کند اگر نرخ اتلاف واقعی از آنچه برای آن تدارک دیده شده بزرگ‌تر باشد.

در نتیجه، برای ساختن طرح‌های ارسال قابل اطمینان و پایدار، دسته‌ای جدید از کدها مورد نیاز

است. کدهای آبشاری^۱ تمامی مسائل مطرح شده بالا را حل می کنند. کدهای آبشاری برای انتقال اطلاعات روی شبکه های کامپیوتری به صورت ایده آلی مناسب هستند. به منظور این که کدهای آبشاری در عمل کار کنند نیاز است که تضمین داده شود آن ها دارای یک کدگذار و کدگشای سریع هستند و این که کدگشا قادر است نمادهای اصلی را از هر مجموعه از نمادهای خروجی که اندازه آن نزدیک به بهینه است با احتمال بالا بازیابی کند. این گونه کدهای آبشاری را فراگیر^۲ نامیده می شود. اولین دسته از این کدهای آبشاری فراگیر، کدهای LT هستند [۲۵، ۳۴].

۳-۴-۱ کدهای LT

کدهای LT اولین تحقق کدهای آبشاری دیجیتال هستند که در سال ۲۰۰۲ توسط مایکل لابی ابداع شدند. کدهای LT در موارد بسیار متنوعی کاربرد دارند که می توان به موارد زیر اشاره کرد. پخش همگانی، شبکه رادیو شناختی، ذخیره سازی توزیع شده پایدار، تحویل جریانی از محتوا، تحویل محتوا به کاربران تلفن همراه در شبکه های بی سیم، کاربردهای نظیر به نظیر و شبکه های حسگر بی سیم. این کدها بدون نرخ هستند به این معنا که تعداد نمادهای کدگذاری شده توسط آن ها، به طور بالقوه نامحدود است. مدل اتلاف کانال پاک شونده هرچه باشد، این کدها قادرند تعداد مورد نیاز نماد کد شده را تولید کرده و روی کانال ارسال نمایند تا زمانی که تعداد کافی نماد کد شده برای بازیابی داده به کدگشا برسد. به این دلیل که کدگشا می تواند داده را از حداقل تعداد نماد کد شده ممکن بازیابی کند، این کدها نزدیک به بهینه هستند.

۳-۴-۱-۱ کدگذاری LT

فرایند کدگذاری LT را با یک گراف دو طرفه می توان نشان داد که در طرف چپ آن k نماد اصلی (گره های ورودی) و در سمت راست گراف n نماد کدگذاری (گره های خروجی) قرار دارد. مراحل کدگذاری

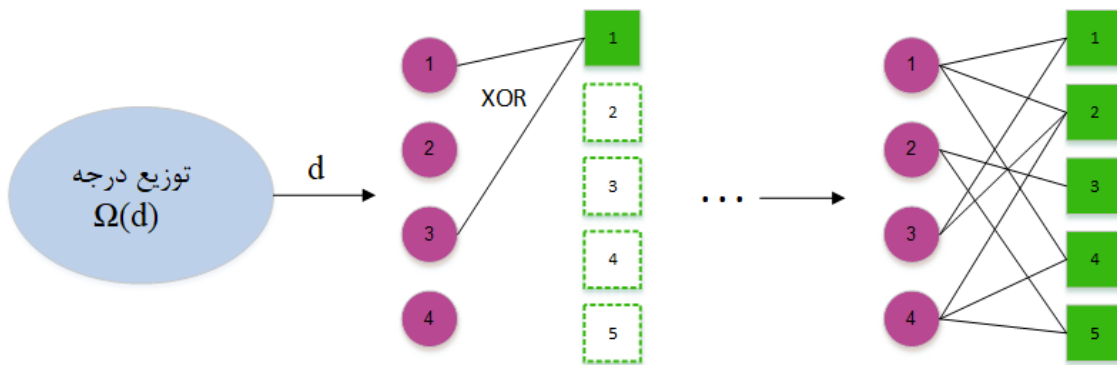
۱. Fountain codes

۲. Universal

به صورت زیر است:

- ◀ انتخاب درجه نماد کدگذاری d از یک توزیع درجه به صورت تصادفی
- ◀ انتخاب d نماد ورودی به صورت تصادفی و یکنواخت به عنوان همسایه‌های یک نماد کدگذاری
- ◀ محاسبه مقدار نماد کدگذاری که از عملیات XOR همسایه‌های آن به دست می‌آید.

در شکل ۱، روند کدگذاری LT برای کدی با $k = 4$ و $n = 5$ نشان داده شده است.



شکل ۱-۳. فرایند کدگذاری LT

۳-۴-۱-۲ کدگذاری LT

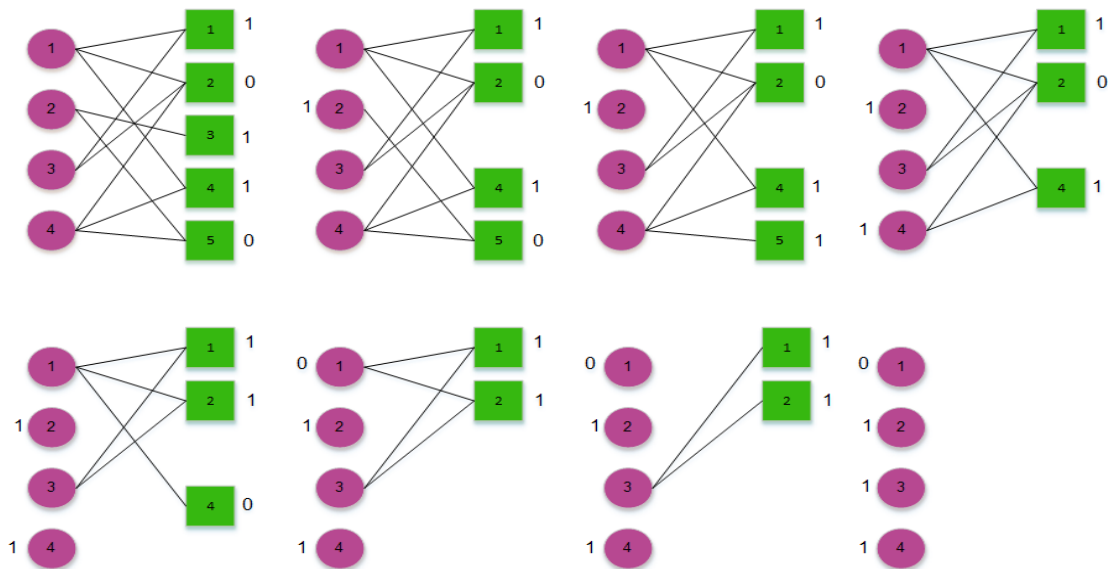
فرایند کدگذاری LT با روش BP صورت می‌گیرد. به منظور کدگذاری نمادهای کدگذاری، کدگشا نیاز دارد تا از درجه و مجموعه همسایه‌های هر کدام از آن‌ها مطلع باشد. فرایند کدگذاری دارای گام‌های زیر است:

- ◀ تمام نمادهای کدگذاری با یک همسایه (درجه یک) آزاد می‌شوند تا همسایه یکتای خود را آشکار کنند.
- ◀ مجموعه نمادهای ورودی آشکار شده که هنوز پردازش نشده‌اند، ریپل نامیده می‌شود و در این گام تمامی نمادهای ورودی آشکار شده در ریپل می‌باشند.

◀ یک نماد ورودی در ریپل پردازش می‌گردد: به‌عنوان یک همسایه از تمام نمادهای کدگذاری که در همسایگی آن‌هاست، حذف می‌شود و تمامی نمادهای کدگذاری دیگر که دارای دقیقاً یک همسایه هستند به همین صورت آزاد می‌گردند.

◀ فرایند کدگذاری موفق خواهد بود چنانچه همه نمادهای ورودی در پایان بازیابی شوند. اگر حداقل یک نماد ورودی آشکارنشده در انتها وجود داشته باشد، فرایند با شکست مواجه می‌گردد.

در ادامه با ارائه مثالی این الگوریتم را نشان می‌دهیم.



شکل ۳-۲. فرایند کدگذاری LT [۲۴]

۳-۴-۱-۳ توزیع درجه‌ها کدهای LT

توزیع درجه برای عملکرد کدهای LT دارای اهمیت بسیاری است. هر نماد کدگذاری دارای یک درجه از یک توزیع درجه است که به‌صورت مستقل انتخاب می‌گردد. در طراحی یک توزیع درجه، دو هدف زیر دنبال می‌شود

(۱) میانگین تعداد نمادهای کدگذاری موردنیاز برای تضمین موفقیت کدگذاری تا حد امکان کم باشد.

(۲) میانگین درجه که بیانگر تعداد عملیات XOR برای تولید یک نماد کدگذاری است تا حد امکان کم باشد؛ بنابراین به کاهش پیچیدگی کدگذاری و کدگذاری کمک می‌کند.

ویژگی اساسی موردنیاز یک توزیع درجه خوب این است که نمادهای ورودی چنان که پردازش می‌گردند با یک نرخ یکسان به ریپل اضافه شوند. این ویژگی الهامبخش برای نام توزیع Soliton است. رفتار مطلوب این است که اندازه ریپل هرگز خیلی بزرگ یا خیلی کوچک نشود.

☼ توزیع Ideal Soliton

توزیع Ideal soliton (ISD) از نظر تعداد مورد انتظار نمادهای کدگذاری موردنیاز برای بازیابی داده، رفتار ایده‌آل از خود نشان می‌دهد. متأسفانه، مانند بیشتر موارد ایده‌آل، این توزیع عملکرد ضعیفی دارد بنابراین استفاده از آن در عمل بی‌فایده است. با این حال، پایه توزیع‌های پایدار است که در ادامه به توصیف آن‌ها پرداخته خواهد شد. ISD به صورت زیر تعریف می‌گردد

$$\rho(i) = \begin{cases} \rho(1) = \frac{1}{k} \\ \rho(i) = \frac{1}{i(i-1)} \quad i = 2, \dots, k. \end{cases} \quad (۱۲-۳)$$

☼ توزیع Robust Soliton

اگرچه ISD در عمل ضعیف کار می‌کند، بینشی به یک توزیع پایدار ارائه می‌دهد. مشکل ISD این است که اندازه ریپل مورد انتظار بسیار کوچک است بنابراین این احتمال وجود دارد که هرگونه تغییر در اندازه ریپل موجب ناپدید شدن آن گردد و آنگاه کل فرایند با شکست مواجه شود. توزیع Robust Soliton (RSD) تضمین می‌دهد که اندازه مورد انتظار ریپل به اندازه کافی بزرگ است به طوری که با احتمال زیاد ریپل هرگز به صورت کامل ناپدید نمی‌گردد. از طرف دیگر، به منظور کمینه کردن تعداد

کل نمادهای کدگذاری مورد استفاده، مهم است که اندازه مورد انتظار ریپل کمینه شود در نتیجه نباید تعداد زیادی از نمادهای کدگذاری آزاد شده، نمادهای ورودی موجود در ریپل را بیش از حد پوشش دهند.

فرض کنید δ احتمال شکست مجاز کدگشا برای بازبایی داده برای k نماد کدگذاری باشد. RSD که با $\mu(\cdot)$ نشان داده می شود، به صورت زیر تعریف می گردد. اندازه مورد انتظار ریپل را

$$R = c \cdot \ln(k/\delta) \sqrt{k} \quad \text{برای برخی ثابت } c > 0 \text{ در نظر بگیرید، داریم}$$

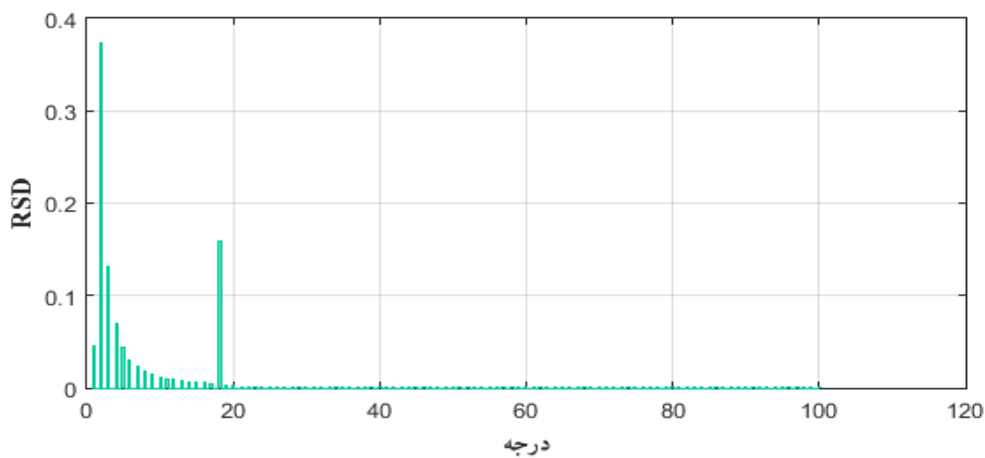
$$\tau(i) = \begin{cases} R/ik & i = 1, \dots, k/R - 1 \\ R \ln(R/\delta)/k & i = k/R \\ 0 & i = k/R + 1, \dots, k \end{cases} \quad (13-3)$$

با اضافه کردن تابع ISD $\rho(\cdot)$ به $\tau(\cdot)$ و نرمالیزه کردن، به توزیع $\mu(\cdot)$ دست می یابیم

$$\beta = \sum_{i=1}^k \rho(i) + \tau(i) \quad (14-3)$$

$$\mu(i) = (\rho(i) + \tau(i))/\beta \quad i = 1, \dots, k \quad (15-3)$$

هیستوگرام درجه ها در توزیع RSD به صورت زیر است.



شکل ۳-۳. توزیع RSD برای $k=100$ ، $c=0.08$ و $\delta=0.1$

قضیه ۳-۱: (تعداد نمادهای کدگذاری) تعداد نماد کدگذاری از رابطه زیر به دست می آید [۲۵]

$$n = k + O(\sqrt{k} \cdot \ln^2(k/\delta)) \quad (۱۶-۳)$$

اثبات:

$$\begin{aligned} n &= k \beta = k \cdot \left(\sum_i \rho(i) + \tau(i) \right) \\ &= k + \sum_{i=1}^{k/R} \frac{R}{i} + R \ln(R/\delta) \\ &\leq k + H(k/R) + R \cdot \ln(R/\delta) \end{aligned}$$

توزیع شکرالهی

در [۳۴]، شکرالهی چندین توزیع درجه بهینه شده با استفاده از یک روش تجربی برای مقادیر مختلف k ارائه کرده است. این روش بر این اساس است که توزیع درجه خروجی باید به گونه ای طراحی شود که بازیابی بخش بزرگی از k نماد ورودی را تضمین دهد و برای حل این مسئله از ایده ی لابی در [۳۵] استفاده شده است. توزیع های شکرالهی به صورت زیر هستند

k	65536	80000	100000	120000
Ω_1	0.007969	0.007544	0.006495	0.004807
Ω_2	0.493570	0.493610	0.495044	0.496472
Ω_3	0.166220	0.166458	0.168010	0.166912
Ω_4	0.072646	0.071243	0.067900	0.073374
Ω_5	0.082558	0.084913	0.089209	0.082206
Ω_8	0.056058		0.041731	0.057471
Ω_9	0.037229	0.043365	0.050162	0.035951
Ω_{18}				0.001167
Ω_{19}	0.055590	0.045231	0.038837	0.054305
Ω_{20}		0.010157	0.015537	
Ω_{65}	0.025023			0.018235
Ω_{66}	0.003135	0.010479	0.016298	0.009100
Ω_{67}		0.017365	0.010777	
ε	0.038	0.035	0.028	0.02
a	5.87	5.91	5.85	5.83

شکل ۳-۴. توزیع درجه های برای مقادیر k ، سربار ε و میانگین درجه k [۳۴].

توزیع Poisson Robust Soliton

در [۳۶] ویژگی‌های توزیع پواسون (PD) و RSD با یکدیگر ترکیب شده و PRSD پیشنهاد شده است که با انتخاب مناسب مؤلفه‌های c ، δ و λ ، این توزیع از توزیع RS بهتر عمل می‌کند. سهم درجه‌های معین در یک توزیع درجه، بر قابلیت کدگشایی کدهای LT حکم‌فرماست. یک ویژگی آن، بیشینه‌بودن سهم نمادهای کدشده با درجه ۲ در توزیع درجه است. در توزیع پواسون که به‌صورت زیر تعریف می‌گردد، با انتخاب مؤلفه مناسب می‌توان تعداد بیشتری درجه ۱ نسبت به RS تولید کرد.

$$P(d) = m^d e^{-m} / d! \quad d = 1, 2, \dots, k \quad (۱۷-۳)$$

توزیع پواسون به‌دلیل اندازه معقول ریپل مورد انتظار در گام‌های ابتدایی کدگشایی می‌تواند کدگشایی را در بازه‌ای معین بهبود دهد. بر اساس ویژگی توزیع درجه، توزیع اصلاح‌شده پواسون به‌صورت زیر تعریف می‌شود

$$\theta(d) = \begin{cases} \frac{1}{2} & d = 2 \\ \frac{\lambda^d e^{-\lambda}}{d!} & d = 1, 3, \dots, k \end{cases} \quad (۱۸-۳)$$

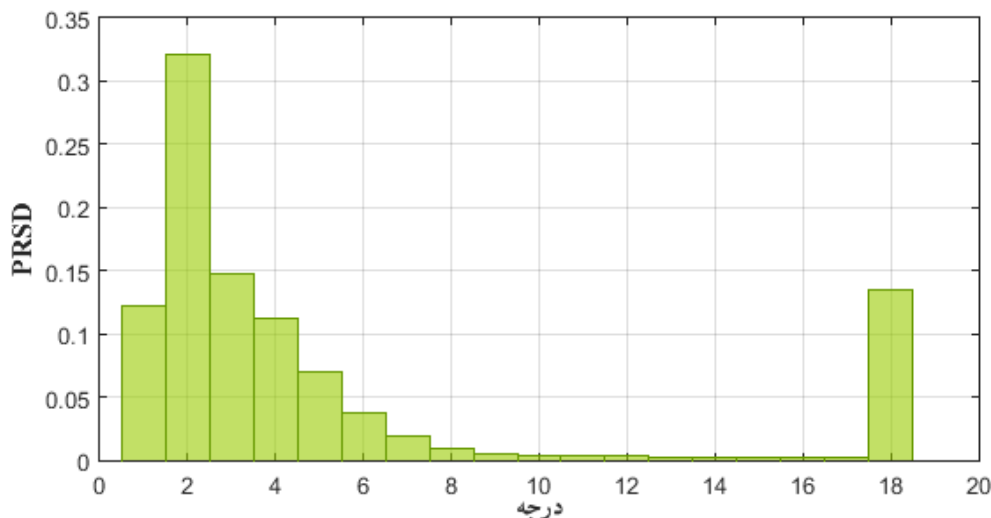
که λ یک ثابت مثبت است.

سپس توزیع PRSD از رابطه زیر به‌دست می‌آید

$$\Omega(d) = \frac{\theta(d) + \tau(d)}{\sum_{d=1}^k \theta(d) + \tau(d)} \quad d = 1, 2, \dots, k \quad (۱۹-۳)$$

برای پاسخ دادن به الزامات طراحی یک توزیع درجه خوب، میانگین درجه نمادهای کد شده باید تا حد ممکن کم باشد. به‌این دلیل که RSD توسط $\rho(d)$ و $\tau(d)$ ساخته شده و PRSD استفاده از $\theta(d)$ و $\tau(d)$ تولید شده است، مسئله میانگین درجه می‌تواند به‌صورت تقریبی به شکل امیدهای ریاضی $\theta(d)$ و $\rho(d)$ دوباره فرموله گردد. به‌صورت ایده‌آل، مقدار مناسب λ می‌تواند از برابر

قراردادن امید ریاضی $\rho(d)$ با امید ریاضی $\theta(d)$ استنباط شود. به این دلیل که امید ریاضی $\theta(d)$ با افزایش k افزایش می‌یابد، مطلوب است که k تا حد امکان کوچک باشد. بر اساس خصوصیات ریاضی PD، PD وقتی $k < 20$ است مشابه BD می‌شود [37]. بنابراین مقدار λ به دست آمده از نقطه 20 $k =$ را در نظر می‌گیریم. بر طبق رابطه $E[\rho(k=20)] = E[\theta(k=20)] = 3.577$ ، $\lambda \approx 3.04$ به دست می‌آید.



شکل ۳-۵. توزیع PRSD برای $k=100$ ، $c=0.08$ ، $\delta=0.1$ و $\lambda=3.04$

توزیع Combined Poisson Robust Soliton

با توجه به نرخ موفق کدگشایی توزیع پواسون بهبودیافته^۱ (IPD) زمانی که کدگشا تعداد نماد کد گذاری کمتری نسبت به RSD دریافت می‌کند، در [38] با ترکیب این توزیع و RSD، توزیع جدیدی پیشنهاد شده است که به کاهش سربار و کاهش زمان مصرفی برای کدگذاری و کدگشایی کمک می‌کند. با نرمالیزه کردن توزیع اصلاح شده پواسون، IPD به صورت زیر تعریف می‌گردد

$$\eta(d) = \frac{\theta(d)}{\sum_{d=1}^k \theta(d)} \quad d = 1, 2, \dots, k \quad (20-3)$$

در این توزیع نیز با توجه به توضیحات قبلی در PRS و با توجه به رابطه

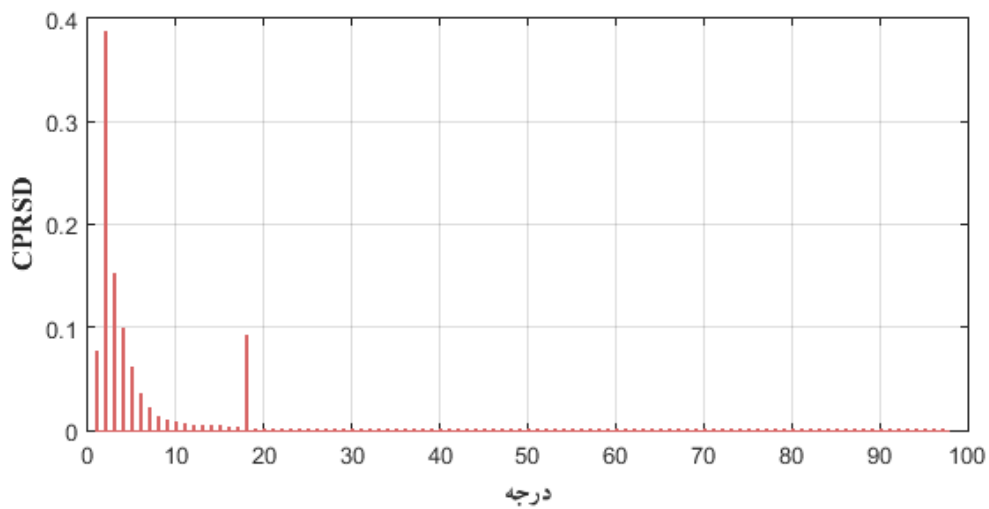
۱. Improved Poisson Distribution

بیان می‌شود $E[\eta(k=20)] = E[\rho(k=20)] = 3.577$ ، $\lambda \approx 3$ به دست می‌آید. توزیع CPRS به صورت زیر

بیان می‌شود

$$\Lambda(d) = \eta(d).a + \mu(d).(1-a) \quad (21-3)$$

که a بین ۰ و ۱ قرار دارد. برای a بزرگ‌تر از ۰/۵ ویژگی IPD به طور عمده در توزیع درجه نشان داده می‌شود. هیستوگرام درجه این توزیع در شکل زیر به نمایش گذاشته شده است.



شکل ۳-۶. توزیع CPRSD برای $k=100$ ، $c=0.08$ ، $\delta=0.1$ ، $\lambda=3.04$ و $a=0.42$

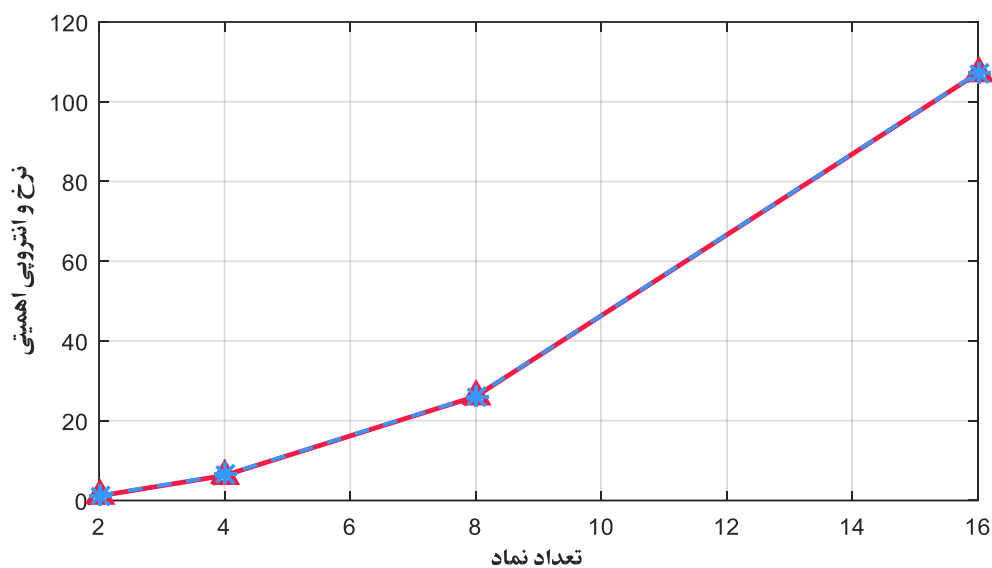
فصل ۴ : نوآوری و نتایج شیه سازی

۴-۱ مقدمه

در این فصل ابتدا به بررسی نرخ و آنتروپی برحسب MIM پرداخته، سپس نرخ را برای روش‌های فشرده‌سازی MIM-هافمن، هافمن، شانون و شانون فانو الیاس محاسبه خواهیم کرد و در ادامه نتایج عددی مقایسه این روش‌ها را برحسب افزونگی نشان خواهیم داد. سیستم ذخیره‌سازی ابری پیشنهادی و مدل‌هایی که برای ذخیره‌سازی در نظر گرفته‌ایم را توضیح داده و دو روش ذخیره‌سازی مبتنی بر تکرار و مبتنی بر کدگذاری را روی سیستم پیشنهادی پیاده‌سازی خواهیم کرد. در ذخیره‌سازی ابری بر پایه کدهای LT، کدگذاری با استفاده از چهار توزیع درجه صورت خواهد گرفت و به مقایسه و بررسی احتمال موفقیت بازیابی آن‌ها خواهیم پرداخت. در انتها الگوریتمی به‌منظور بهبود زمان بازیابی ارائه خواهیم داد. تمامی شبیه‌سازی‌ها توسط نرم‌افزار MATLAB انجام شده‌اند.

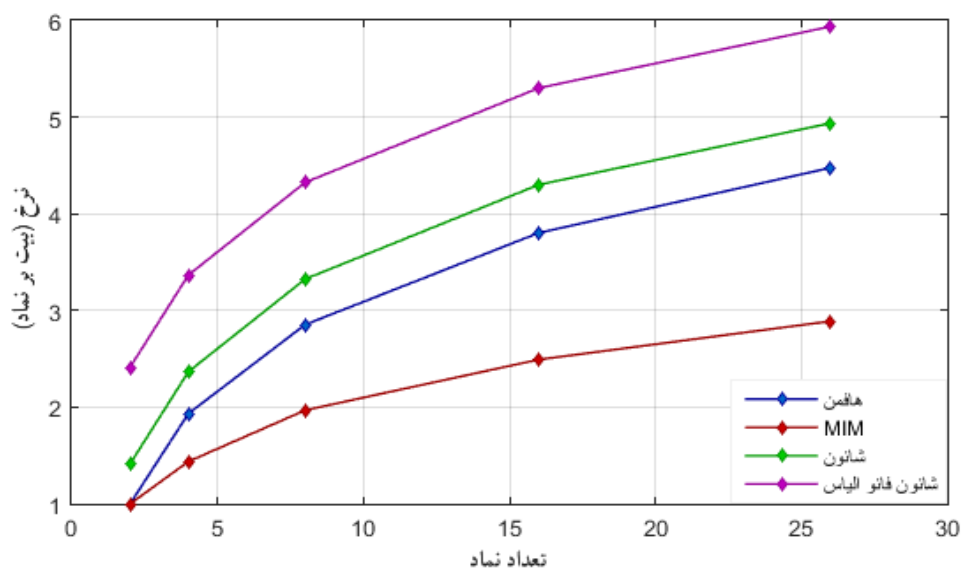
۴-۲ کدگذاری منبع بر پایه MIM

به‌منظور فشرده‌سازی با استفاده از MIM، ابتدا یک منبع را با تعداد نمادهای مختلف $\{۲، ۴، ۸، ۱۶\}$ در نظر گرفته و دنباله‌ای از این نمادها به‌صورت تصادفی تولید می‌کنیم. سپس ضریب اهمیت هر یک از نمادها $M(p_i)$ را بر اساس احتمالات آن‌ها محاسبه کرده و بر اساس یک آستانه معین، نمادهایی که دارای اهمیت کمتری هستند را حذف می‌نماییم. پس از اعمال کدگذاری هافمن روی دنباله فشرده‌شده، مقادیر دو مفهوم جدید آنتروپی اهمیتی و نرخ اهمیتی که میانگین را به‌جای احتمالات ساده برحسب اهمیت محاسبه می‌کنند را به‌دست می‌آوریم. در این شبیه‌سازی آستانه، چارک اول در نظر گرفته شده است به این معنا که یک‌چهارم نمادهایی که دارای کمترین میزان اهمیت هستند حذف می‌گردند. شکل ۴-۱ مقادیر دو مفهوم جدید آنتروپی اهمیتی و نرخ اهمیتی که میانگین را به‌جای احتمالات ساده برحسب اهمیت محاسبه می‌کنند را به ازای نمادهای مختلف نشان می‌دهد.



شکل ۴-۱. نرخ و آنتروپی اهمیت.

در ادامه دنباله‌ای به طول ۱۰۰۰۰ نماد را بر اساس MIM و سپس کدگذاری هافمن در ۱۰۰۰۰ تکرار فشرده می‌کنیم. همچنین دنباله را بدون در نظر گرفتن MIM، با استفاده از کدگذاری‌های شانون، شانون فانو الیاس و هافمن نیز فشرده کرده، به مقایسه این روش‌ها بر اساس نرخ آن‌ها می‌پردازیم.



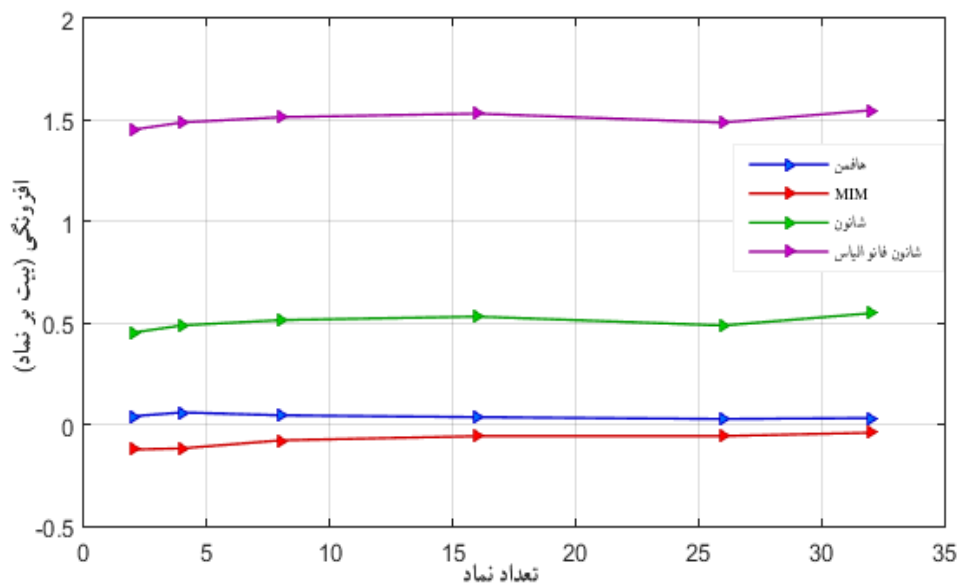
شکل ۴-۲. مقایسه نرخ در روش‌های فشرده‌سازی MIM، هافمن، شانون و شانون فانو الیاس.

همان‌طور که در شکل ۴-۲ مشاهده می‌شود نرخ به دست آمده از روش MIM بسیار کمتر از سایر

روش‌های فشرده‌سازی است بنابراین می‌توان با حجم کمتری به ذخیره‌سازی داده پرداخت. به‌منظور مقایسه روش‌های مختلف از مفهوم افزونگی که به‌صورت زیر تعریف می‌گردد استفاده می‌کنیم

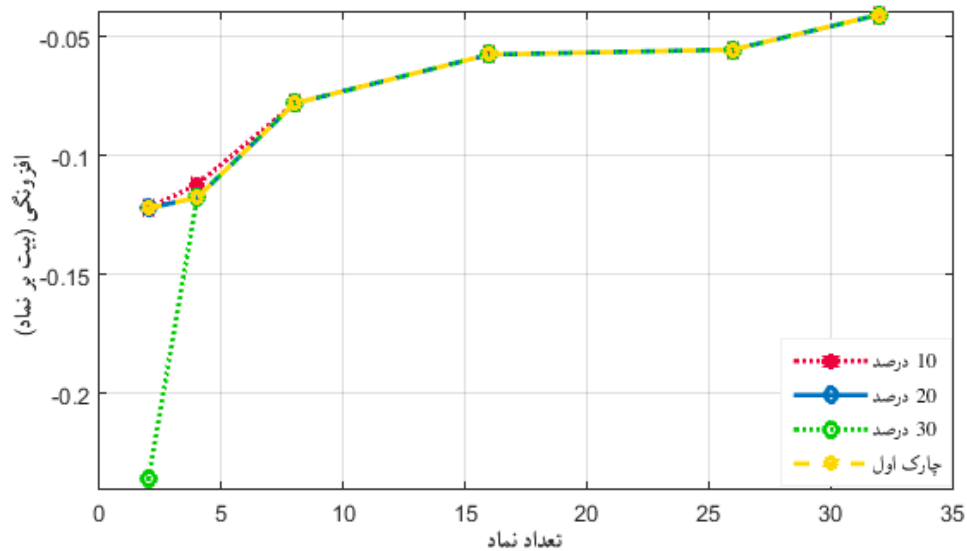
$$\text{Redundancy} = R - H \quad (1-4)$$

نتیجه حاصل از شبیه‌سازی به‌منظور مقایسه افزونگی برای آستانه چارک اول و ۱۰۰۰۰ بار تکرار در شکل زیر آورده شده است. اگرچه میزان افزونگی روش جدید منفی است اما به‌دلیل این‌که اساس قضیه کدگذاری منبع از میانگین در بی‌نهایت و مجموعه نوعی به‌دست آمده است، نمی‌توان از این قضیه برای این روش جدید استدلال کرد؛ با این‌وجود با توجه به ماهیت روش و حذف نمادها، MIM نیز یک روش فشرده‌سازی با اتلاف است.



شکل ۴-۳. مقایسه افزونگی در روش‌های هافمن، MIM، شانون و شانون فانو الیاس.

آستانه حذف نمادها می‌تواند بر اساس محدودیت‌های ذخیره‌سازی و یا سایر الزامات تعیین گردد. به این منظور ما آستانه‌های ۱۰، ۲۰ و ۳۰ درصد را برای حذف نمادها اعمال کرده و تغییرات افزونگی را برای آستانه‌های مختلف و همچنین آستانه چارک اول مورد بررسی قرار داده‌ایم. شکل ۴-۴، میزان افزونگی برای چهار آستانه حذف نماد در تعداد نمادهای مختلف را نشان می‌دهد.



شکل ۴-۴. افزونی برای آستانه‌های مختلف.

بنابر نتایج فوق، هنگامی که تعداد نمادها زیاد می‌شود، اثر آستانه حذف نمادها کاهش می‌یابد و

افزونی برای تمامی آستانه‌ها برابر می‌گردد.

دنباله‌ای به طول ۱۰۰۰۰ نماد، تولیدشده توسط منبعی با ۸ نماد را در نظر می‌گیریم، چنانچه این دنباله با روش ASCII فرستاده شود معادل ۳۰۰۰۰ بیت خواهد بود. به منظور نشان دادن میزان صرفه جویی در فضای ذخیره‌سازی با استفاده از روش جدید، طول دنباله کدگذاری شده با استفاده از روش های MIM-هافمن، هافمن، شانون و شانون فانو الیاس را محاسبه کرده و این بهبود را در جدول ۴-۱ نشان داده‌ایم.

همان‌طور که در جدول مشاهده می‌شود، فضای موردنیاز برای ذخیره‌سازی با استفاده از روش فشرده‌سازی بر پایه MIM کاهش داشته و می‌توان فضای ذخیره‌سازی را از تا حدود ۳۰ درصد با توجه به انتخاب آستانه کاهش داد.

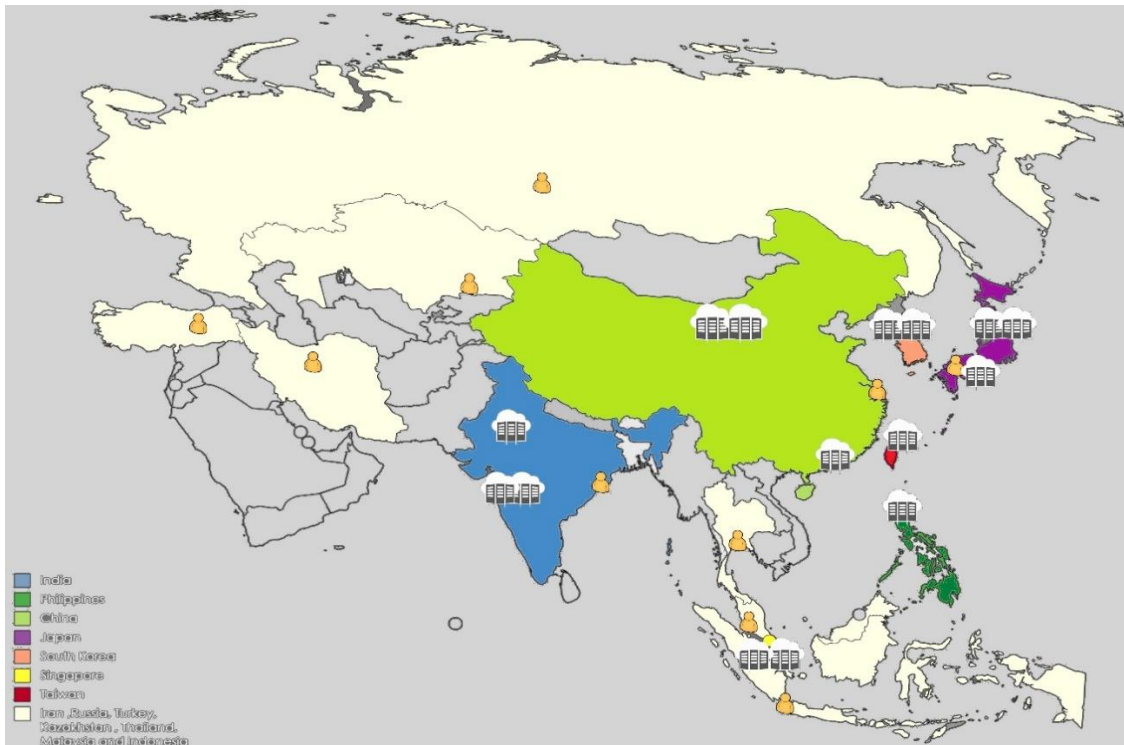
جدول ۴-۱. مقایسه حجم ذخیره‌سازی (بیت) حاصل از روش‌ها به‌ازای آستانه‌های مختلف

شانون فانو الیاس	شانون	هافمن	MIM- هافمن	روش آستانه
۴۳۲۲۲	۳۳۲۲۲	۲۸۳۵۵	۲۴۲۰۰	۱۰
۴۳۲۴۵	۳۳۲۴۵	۲۸۵۵۳	۱۹۶۹۳	۲۰
۴۳۲۲۶	۳۳۲۲۶	۲۸۵۲۳	۱۹۶۶۹	چارک اول
۴۳۲۴۲	۳۳۲۴۲	۲۸۵۳۲	۱۹۶۵۸	۳۰

۴-۳ سیستم ذخیره‌سازی ابری پیشنهادی

در این پایان‌نامه ما دو روش ذخیره‌سازی مبتنی بر تکرار و ذخیره‌سازی مبتنی بر کدهای بدون نرخ را روی یک سیستم ذخیره‌سازی ابری با ۵ منطقه و ۳ ناحیه دسترسی در هر منطقه، در آسیا اعمال می‌کنیم. ذخیره‌سازی خود به دو صورت منطقه‌ای و چندمنطقه‌ای انجام می‌شود. در ذخیره‌سازی مبتنی بر تکرار و در حالت ذخیره‌سازی منطقه‌ای، سه رونوشت از داده روی سه ناحیه متفاوت درون یک منطقه توزیع می‌گردد. در حالی که در حالت چندمنطقه‌ای، شش رونوشت از داده کاربر روی دو منطقه متفاوت ذخیره می‌شود. در روش مبتنی بر کدهای بدون نرخ نیز داده پس از کدگذاری بنابر انتخاب کاربر به دو صورت منطقه‌ای و یا چندمنطقه‌ای ذخیره می‌گردد. چهارچوب کلی برای سیستم ذخیره‌سازی ابری موردنظر در این پایان‌نامه در شکل زیر به نمایش گذاشته شده است.

فرض می‌کنیم کاربران داده خود را از کشورهای ایران (تهران)، روسیه (نوووسیبیرسک)، ترکیه (استانبول)، مالزی (کوالالامپور)، اندونزی (جاکارتا)، چین (شانگهای)، ژاپن (اوساکا)، تایلند (بانکوک)، هند (کلکته)، قزاقستان (آلماتی) به ذخیره‌سازی ابری ارسال کرده و درخواست دستیابی به آن‌ها را خواهند داشت.



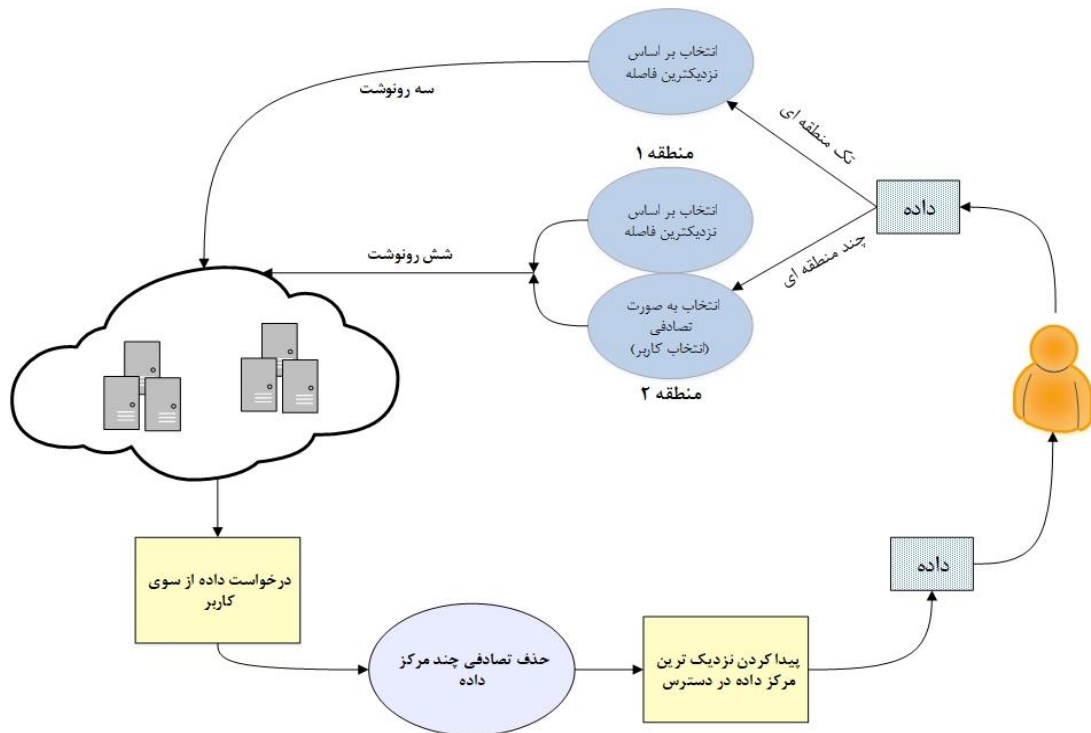
شکل ۴-۵. ساختار کلی سیستم ذخیره‌سازی ابری مدنظر.

در شبیه‌سازی‌ها، ۲۰ داده با اندازه‌های مختلف از ۱۰ کاربر در سرتاسر آسیا و در دو حالت ذخیره‌سازی منطقه‌ای و چندمنطقه‌ای فرض شده است. اگرچه کاربر قادر است که منطقه موردنظر برای نگهداری اطلاعات خود را تعیین کند، در حالت پیش‌فرض ذخیره‌سازی منطقه‌ای، منطقه بر اساس نزدیک‌ترین فاصله از محل درخواست کاربر تا مراکز داده (میانگین فاصله مراکز داده واقع در نواحی دسترسی تا کاربر) مشخص می‌شود. در ذخیره‌سازی چندمنطقه‌ای، فرض بر این است که اولین منطقه بر اساس نزدیک‌ترین فاصله تا کاربر و دومین منطقه به‌صورت تصادفی انتخاب می‌گردد. این فرض تصادفی بودن را می‌توان معادل با انتخاب کاربر در نظر گرفت.

با وجود اینکه سرویس‌دهندگان ابری بزرگ قابلیت دسترسی کاربران به داده خود را ۹۹/۹۹٪ اعلام کرده‌اند [۲۹]، به‌منظور نزدیک‌کردن شبیه‌سازی به واقعیت در صورت بروز اختلالات بزرگ و بررسی اثر حذف مراکز داده، از بین ۱۵ مرکز داده، ۱۰ و سپس ۲ مرکز را به‌صورت تصادفی به‌عنوان مراکز غیرقابل‌دسترسی انتخاب خواهیم کرد.

۴-۴ ذخیره‌سازی بر پایه تکرار

در فرایند شبیه‌سازی بازیابی ذخیره‌سازی مبتنی بر تکرار، فرض می‌شود که درخواست بازیابی اطلاعات به‌صورت تصادفی از یکی از ۱۰ نقطه مشخص‌شده در آسیا اعلام می‌شود. این فرض بر پایه یکی از اهداف اصلی فضای ابری است که کاربر در سرتاسر جهان از طریق اینترنت بتواند به‌راحتی به اطلاعات خود دست یابد. سپس از بین مراکز داده در دسترس و با استفاده از جدول نواحی دسترسی که در ادامه آمده است، نزدیک‌ترین مرکز داده را برای بازیابی پیدا می‌کنیم. شکل زیر مراحل ذخیره‌سازی بر پایه تکرار را نشان می‌دهد.



شکل ۴-۶. ذخیره‌سازی و بازیابی داده در حالت ذخیره‌سازی بر پایه تکرار.

بازیابی داده در ذخیره‌سازی بر پایه تکرار با حذف ۱ و ۲ مرکز داده همچنان به صورت کامل صورت می‌گیرد و با حذف ۳ مرکز داده نیز احتمال موفقیت بازیابی داده، ۹۹/۸۹ درصد است. همان‌طور که از پیش نیز انتظار می‌رفت این روش کارآمدی بالایی در میزان در دسترس بودن و بازیابی داده دارد اما به‌منظور تضمین قابلیت اطمینان و قابلیت دسترسی نیاز به حجم زیادی فضای

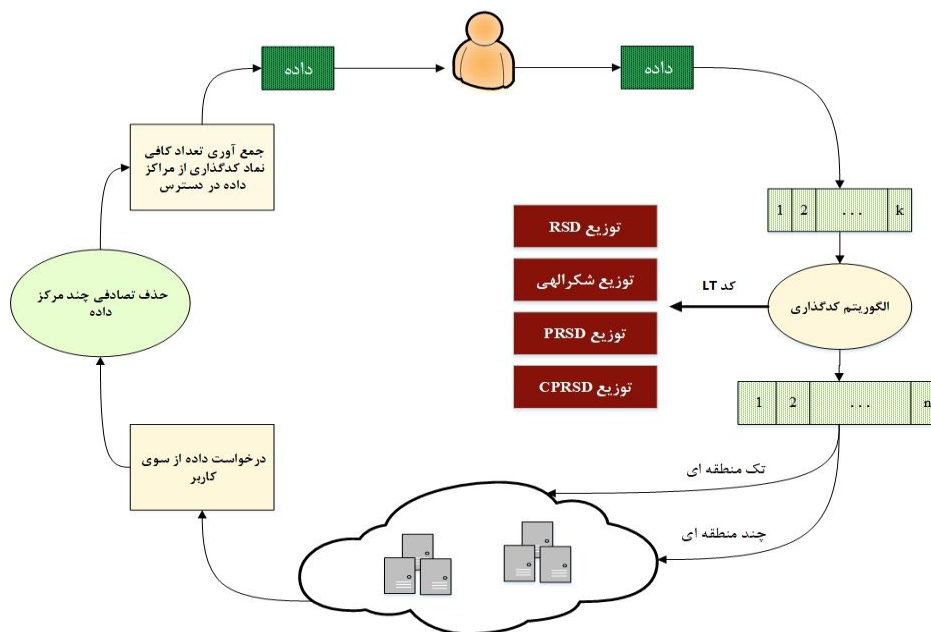
ذخیره‌سازی است که این مسئله از نگاه کاربر و فراهم‌آوردندگان سرویس ابری مطلوب نیست و همچنین برای آینده ذخیره‌سازی ابری باید رویکردهای دیگری را به کار برد.

۴-۵ ذخیره‌سازی بر پایه کدهای LT

در روش ذخیره‌سازی ابری بر پایه کدگذاری پس از برون‌سپاری داده توسط کاربر، ابتدا داده به k بخش با طول برابر تقسیم‌شده که نمادهای اصلی را تشکیل می‌دهند، سپس با استفاده از روش کدگذاری موردنظر، این نمادها به n نماد کدگذاری تبدیل می‌شوند. همانند ذخیره‌سازی مبتنی بر تکرار، کاربر یکی از دو حالت منطقه‌ای یا چندمنطقه‌ای را برای ذخیره‌سازی داده خود انتخاب می‌کند.

زمانی که کاربر داده خود را درخواست می‌کند، ابتدا بر اساس اطلاعات داده که در ماتریسی به نام «متادیتا» ذخیره می‌گردد، مراکز داده و نمادهای کدگذاری معینی که در آن‌ها نگهداری می‌شوند مشخص شده و فرایند بازیابی با کمترین تعداد نماد کدگذاری موردنیاز یعنی k آغاز می‌گردد. چنانچه فرایند کدگشایی نتواند با k نماد کدگذاری، داده اصلی را بازیابی کند، در هر گام درصدی از تعداد نماد به آن‌ها اضافه می‌گردد و این فرایند تا بازیابی کامل داده و یا تا بیشینه تعداد نماد کدگذاری مجاز ادامه پیدا می‌کند. ما در هر گام شبیه‌سازی یک درصد به تعداد نمادها اضافه کرده و مشابه آنچه گفته شد، ابتدا ۱ مرکز داده و سپس ۲ مرکز داده را به‌صورت تصادفی از دسترس خارج می‌کنیم. همچنین ۲۰ داده با اندازه‌های ۱۰۰۰۰ تا ۱۰۰۰۰۰ بیت در دو حالت تک‌منطقه‌ای و چندمنطقه‌ای در نظر گرفته‌ایم.

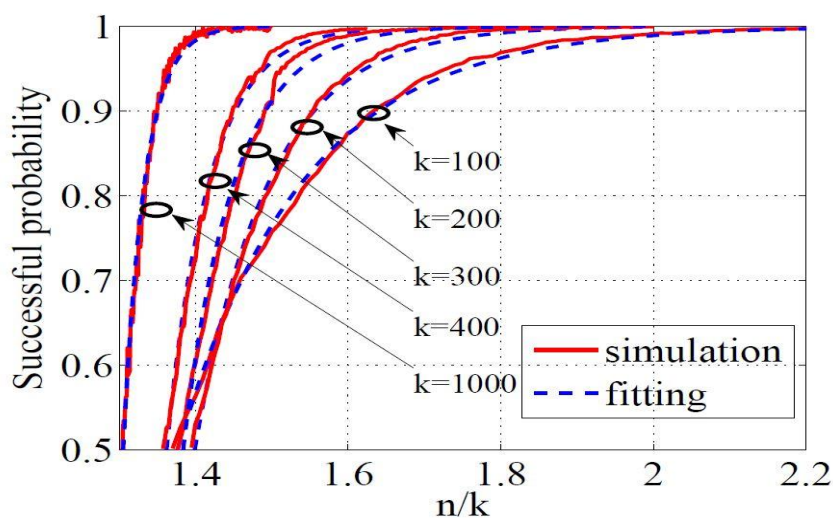
در این پایان‌نامه به‌منظور بهبود موفقیت بازیابی داده و همچنین کاهش تعداد نماد کدگذاری موردنیاز برای کدگشایی، علاوه بر توزیع RSD که در کارهای گذشته استفاده شده است از سه توزیع دیگر استفاده می‌کنیم تا عملکرد سیستم‌های ذخیره‌سازی ابری را بهبود بخشیم. در شکل زیر فرایند توزیع و بازیابی داده بر اساس ذخیره‌سازی LT نشان داده شده است.



شکل ۴-۷. ذخیره‌سازی و بازیابی داده در حالت ذخیره‌سازی بر پایه کدگذاری LT.

۴-۵-۱ انتخاب n

ذخیره‌سازی با کدهای LT با $k = \{100, 250, 500\}$ انجام خواهد شد. به منظور یافتن تعداد نماد کدگذاری n متناظر با هر k ، مقادیر حاصل از رابطه (۳-۱۶) و همچنین مقادیر موجود در شکل ۴-۸ مورد استفاده قرار گرفته است و در نهایت مقادیر حاصل از شبیه‌سازی‌ها در جدول ۴-۲ آورده شده است.



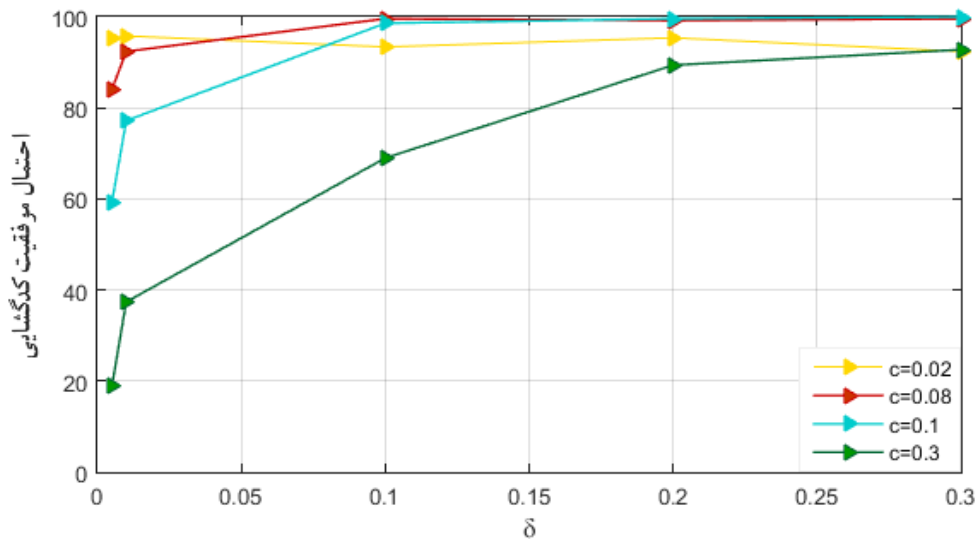
شکل ۴-۸. احتمال موفقیت به ازای تعداد نماد کدگذاری متفاوت [۲۸].

جدول ۴-۲. تعداد نماد کدگذاری با RSD

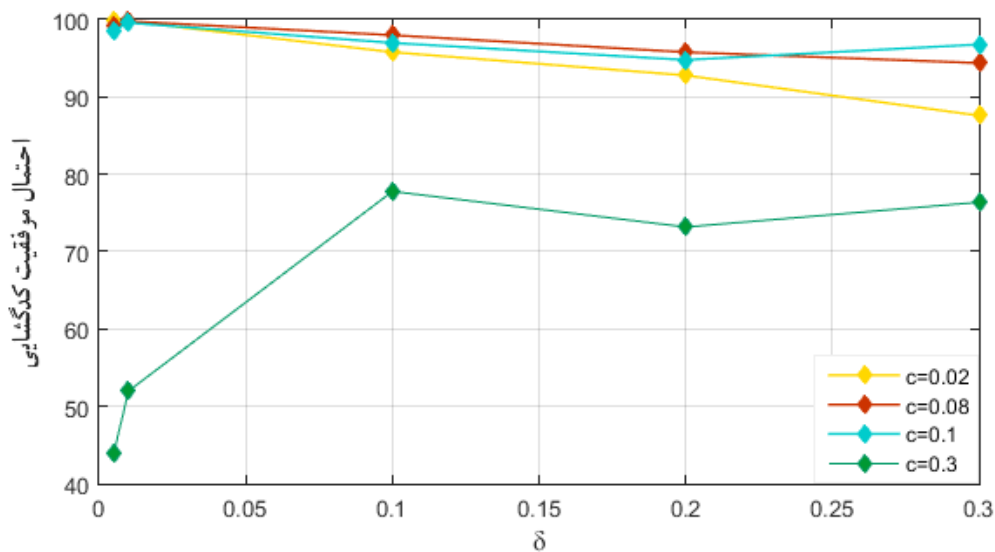
شبه‌سازی	$k + O(\sqrt{k} \cdot \ln^2(k/\delta))$	[۲۸]	n / k
۱۵۸	۱۵۷	۱۶۵	۱۰۰
۲۹۰	۲۷۱	۳۱۰	۲۰۰
۳۸۸	۳۳۳	-	۲۵۰
۴۴۱	۳۹۵	۴۵۰	۳۰۰
۵۶۸	۵۱۸	۵۸۰	۴۰۰
۷۱۰	۶۳۹	-	۵۰۰

۴-۵-۲ انتخاب مؤلفه‌ها

به‌منظور انتخاب مؤلفه‌های c و δ که در عملکرد توزیع درجه‌ها بسیار اثرگذار هستند ابتدا به شبه‌سازی میزان موفقیت کدگذاری کدهای LT با توزیع درجه‌های RSD و PRSD به‌زای c و δ ‌های مختلف پرداخته و بهترین ترکیب را برای شبه‌سازی‌های بعدی برمی‌گزینیم. نتایج زیر برای $k = ۲۵۰$ در ۵۰۰ بار تکرار به‌دست آمده است.



شکل ۴-۹. میزان موفقیت کدگذاری با RSD به‌زای c و δ ‌های مختلف برای $k = ۲۵۰$.



شکل ۴-۱۰. میزان موفقیت کدگشایی PRSD به ازای c و δ های مختلف برای $k=250$.

با توجه به شبیه‌سازی‌های فوق برای $k=100, 250, 500$ ، بهترین انتخاب $c=0.08$ و $\delta=0.1$ است. روند کدگشایی LT احتمالی ست و با توجه به مفهوم δ که احتمال شکست کدگشایی را نشان می‌دهد، $\delta=0.1$ معیاری قابل قبول در عمل و واقعیت است.

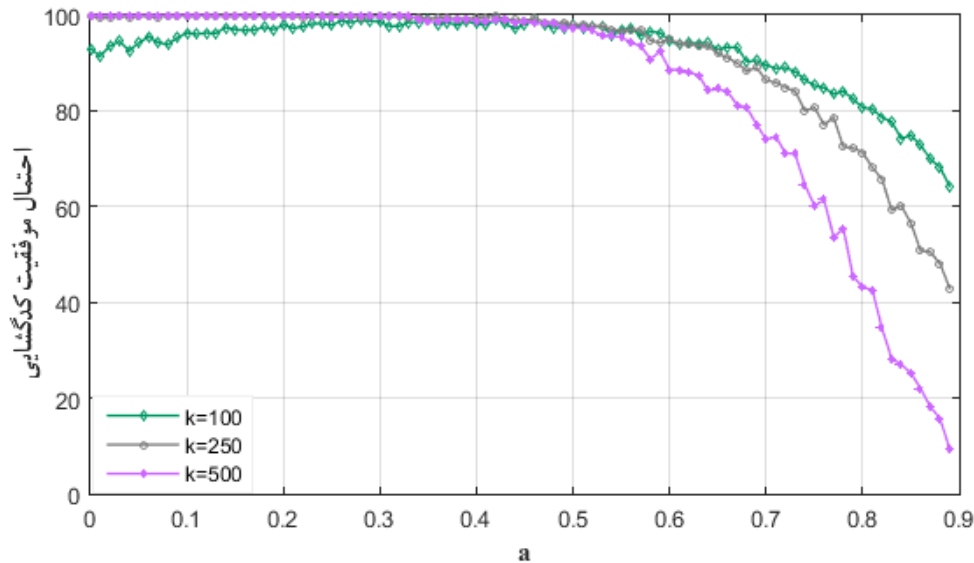
در توزیع CPRSD، مؤلفه a تعیین‌کننده میزان مشارکت و اثرگذاری هر کدام از توزیع‌های IPD و RSD است؛ بنابراین به منظور یافتن بهترین مقدار برای این مؤلفه، عملکرد CPRSD را به ازای مقادیر مختلف a و k مورد بررسی قرار داده‌ایم. همان‌طور که در شکل ۴-۱۱ مشاهده می‌کنید، احتمال موفقیت کدگشایی برای $k=100, 250$ به ترتیب در $a=0.42$ ، $a=0.2$ بالاترین مقدار خود رسیده است و برای $k=500$ دارای چندین بیشینه در بازه $0 < a < 0.33$ است. علاوه بر این، تغییرات a بر روی سربار^۱ نیز اثر می‌گذارد و در انتخاب مؤلفه a می‌توان مصالحه‌ای بین احتمال موفقیت کدگشایی و سربار در نظر گرفت. سربار به صورت زیر تعریف می‌گردد

$$\varepsilon = \frac{n-k}{k} \quad (2-4)$$

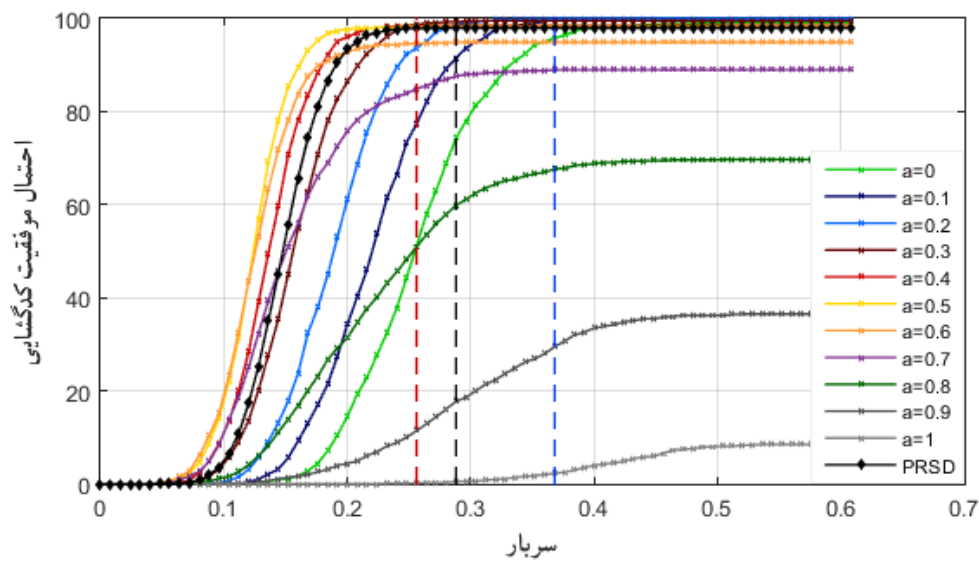
در شکل ۴-۱۲، اثرگذاری a روی سربار برای $k=250$ به نمایش گذاشته شده است؛ برای

۱. Overhead

در $a=0.2$ در سربار 0.368 به احتمال موفقیت کدگشایی 100% رسیده است درحالی که برای $a=0.4$ در سربار 0.256 توانسته است با احتمال 98.5% کدگشایی کند. توزیع PRS نیز در سربار 0.288 به احتمال موفقیت 97.97% دست یافته است. با توجه به توضیحات فوق، در ادامه شبیه‌سازی‌ها از $a=0.42$ برای $k=100$ ، از $a=0.4$ برای $k=250$ و از $a=0.39$ برای $k=500$ استفاده خواهیم کرد.



شکل ۴-۱۱. میزان موفقیت کدگشایی LT با توزیع CPRSD به‌ازای a و k های مختلف.

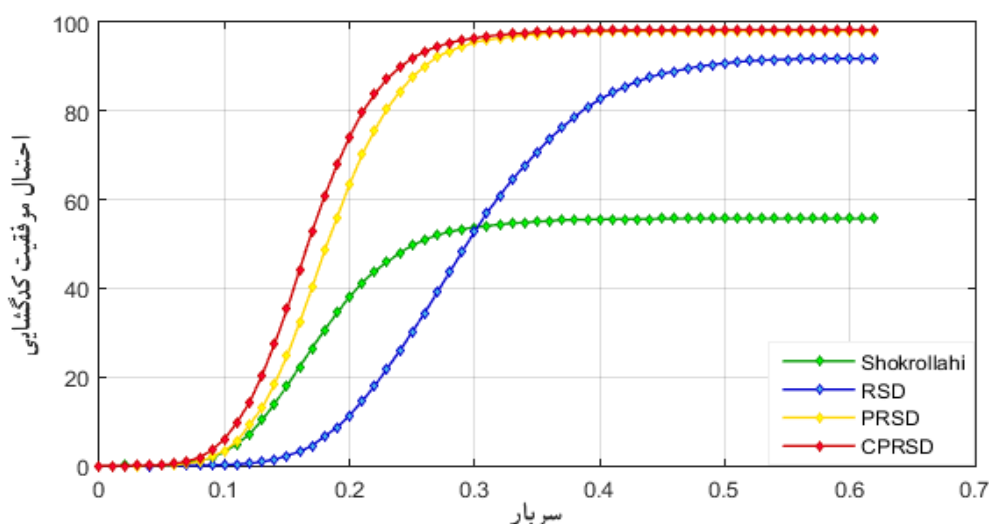


شکل ۴-۱۲. میزان موفقیت کدگشایی LT با توزیع CPRSD برای $k=250$ به‌ازای a های مختلف.

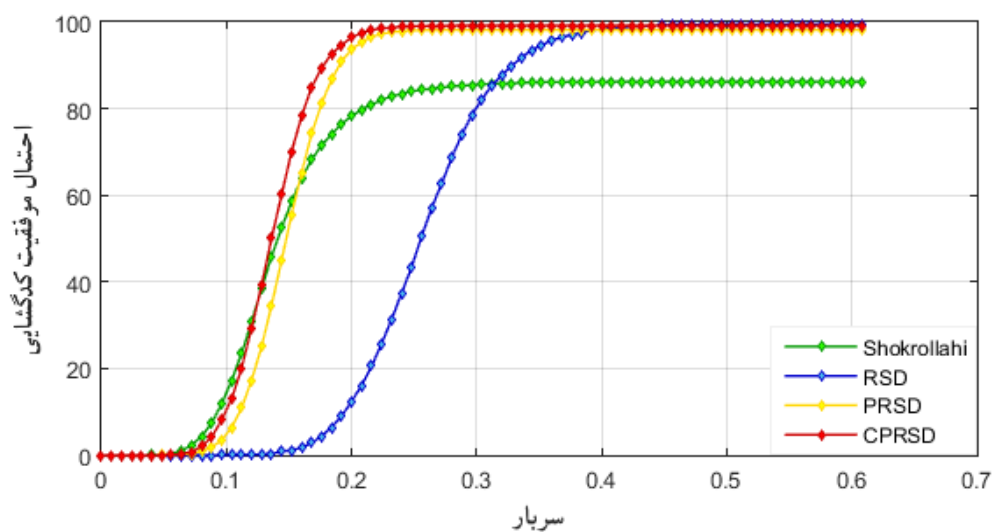
۴-۵-۳ بررسی احتمال موفقیت بازیابی

ابتدا به مقایسه میزان موفقیت بازیابی برای چهار توزیع موردنظر کد LT پیش از حذف مراکز داده می پردازیم، سپس عملکرد این توزیع‌ها در بازیابی تصادفی داده پس از حذف ۱ و ۲ مرکز داده بررسی خواهیم کرد.

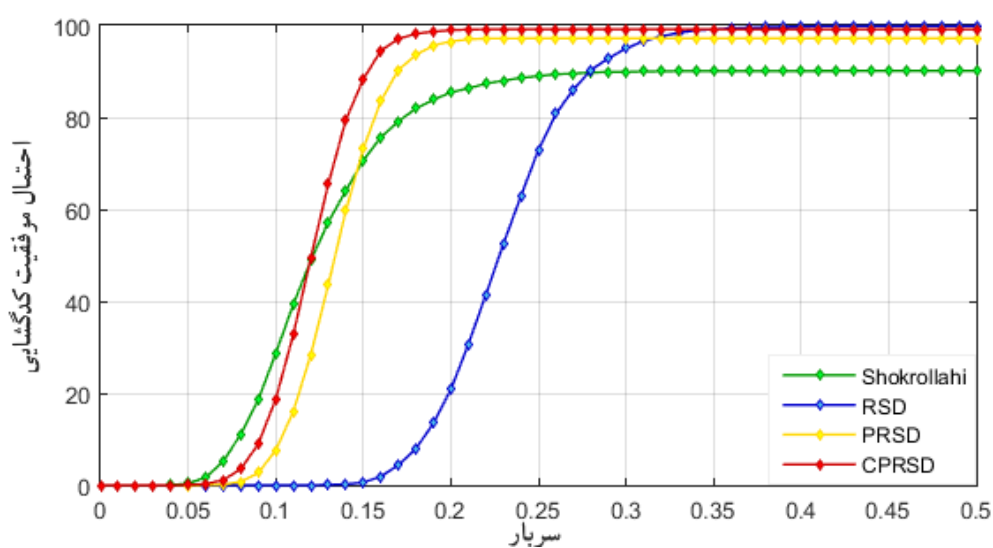
در هنگام بررسی اثر توزیع درجه‌های کد LT در سیستم ذخیره‌سازی ابری پیشنهادی، به دلیل تصادفی بودن ماهیت کدگذاری و کدگشایی LT و همچنین انتخاب تصادفی برخی از مؤلفه‌ها در برنامه، دو تکرار در شبیه‌سازی به منظور اعتباربخشی به نتایج حاصل در نظر گرفته‌ایم. شبیه‌سازی در ۱۰۰ بار تکرار خارجی اجرا شده است که انتخاب درجه، تشکیل ماتریس متادیتا و فرایند کدگذاری و توزیع روی فضای ابری را شامل می‌شود و هر تکرار خارجی خود دارای ۱۰۰ بار تکرار داخلی بازیابی داده‌های مختلف به صورت تصادفی است. نتایج حاصل از مقایسه روش ذخیره‌سازی ابری بر پایه کد های LT با توزیع درجه‌های RSD، PRSD و CPRSD و شکرالهی بدون حذف مرکز داده، در شکل های زیر به ترتیب برای $k = \{100, 250, 500\}$ آورده شده است. در تمام شبیه‌سازی‌هایی که نتیجه آن‌ها در ادامه آمده است؛ $c = 0.08$ ، $\delta = 0.1$ و $\lambda = 3.04$ در نظر گرفته شده است.



شکل ۴-۱۳. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های RSD، PRSD، CPRSD و شکرالهی برای $k = 100$ بدون حذف مراکز داده.



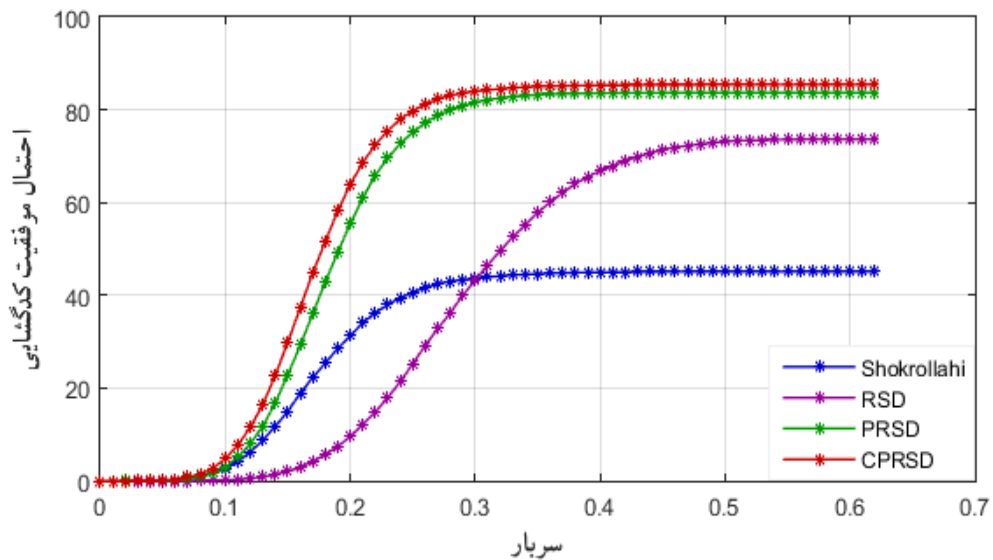
شکل ۴-۱۴. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های RSD، PRSD، CPRSD و شکرالهی برای $k = 250$ بدون حذف مراکز داده.



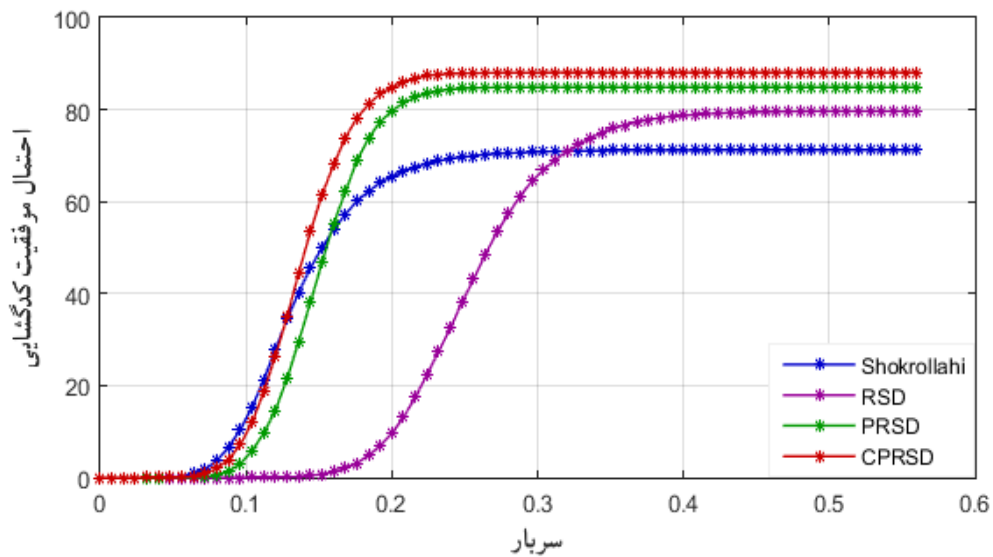
شکل ۴-۱۵. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های RSD، PRSD، CPRSD و شکرالهی برای $k = 500$ بدون حذف مراکز داده.

با توجه به نتایج شبیه‌سازی برای $k = 100$ ، دو توزیع PRSD و CPRSD عملکرد بسیار بهتری از دو توزیع دیگر دارند. با افزایش تعداد نماد اصلی، عملکرد توزیع‌ها به هم نزدیک می‌شود. توزیع شکرالهی به ازای تعداد نماد اصلی بیشتر، در سربار کمتر موفقیت بیشتری در بازیابی نسبت به توزیع RSD دارد. RSD نسبت به سایر توزیع‌ها برای بازیابی نیاز به سربار بیشتری دارد که این خود به

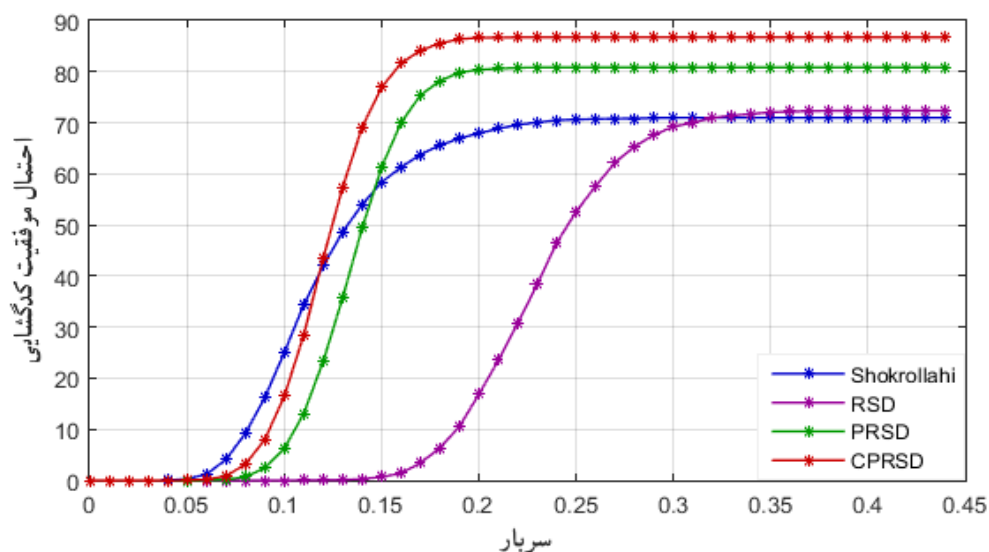
معنای زمان طولانی‌تر برای بازیابی و فضای موردنیاز بیشتر برای ذخیره‌سازی است. بر اساس نتایج به‌دست آمده توزیع CPRSD در صورت انتخاب مؤلفه‌های مناسب می‌تواند در سربار کم به موفقیت بالایی در بازیابی داده در سیستم‌های ذخیره‌سازی ابری دست یابد. در ادامه با حذف یک مرکز داده به بررسی احتمال موفقیت کدگشایی می‌پردازیم.



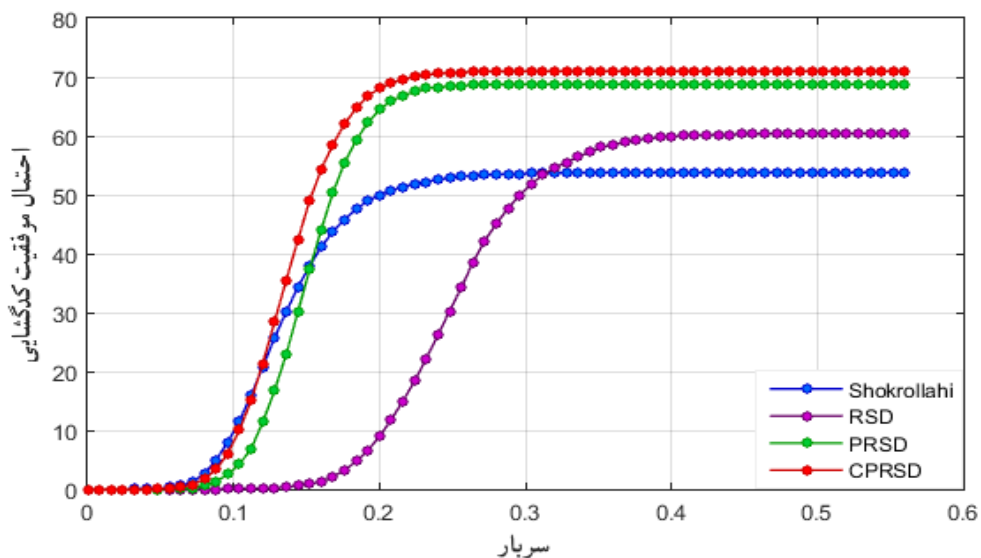
شکل ۴-۱۶. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف برای $k = 100$ و حذف یک مرکز داده.



شکل ۴-۱۷. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف برای $k = 250$ و حذف یک مرکز داده.

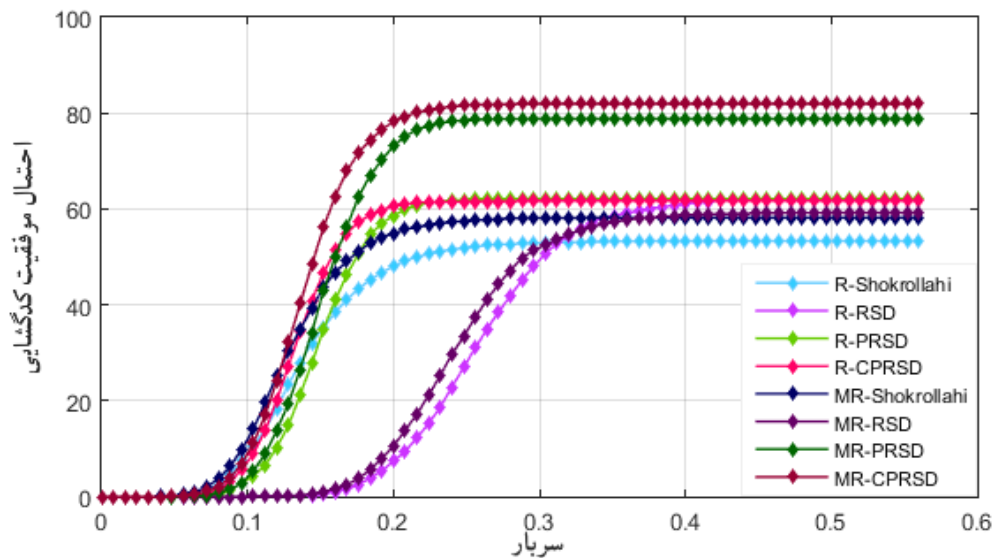


شکل ۴-۱۸. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف برای $k = 500$ و حذف یک مرکز داده. میزان موفقیت کدگشایی در توزیع RSD با حذف یک مرکز داده بیش از سایر توزیع‌ها کاهش یافته است؛ همچنین با افزایش k ، روند کاهش میزان موفقیت کدگشایی برای تمامی توزیع‌ها بیشتر می‌شود. در شکل زیر، احتمال موفقیت با حذف دو مرکز داده برای $k = 250$ نشان داده شده است. با توجه به نتایج به دست آمده و شکل ۴-۱۹، دو توزیع CPRSD و PRSD در صورت حذف دو مرکز داده، با کاهش احتمال موفقیت بازیابی کمتری نسبت به دو توزیع دیگر مواجه هستند.



شکل ۴-۱۹. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف برای $k = 250$ و حذف دو مرکز داده.

در ادامه با هدف بررسی تفاوت دو مدل ذخیره‌سازی تک‌منطقه‌ای و چندمنطقه‌ای، برای حالت حذف دو مرکز داده شبیه‌سازی را انجام داده‌ایم.



شکل ۴-۲۰. بررسی تفاوت دو مدل ذخیره‌سازی تک‌منطقه‌ای و چندمنطقه‌ای برای توزیع‌های مختلف و $k = 250$

همان‌طور که در شکل ۴-۲۰ مشاهده می‌شود و انتظار داشتیم، حالت چندمنطقه‌ای دارای عملکرد بهتری نسبت به حالت تک‌منطقه‌ای است.

به‌منظور مقایسه فضای مورد نیاز برای ذخیره‌سازی داده با دو روش تکرار و کدگذاری، یک داده با حجم ۱۰ مگابایت را در نظر گرفته و فضای مورد نیاز برای ذخیره‌سازی آن را در حالت تکرار تک‌منطقه‌ای، تکرار چندمنطقه‌ای و در حالت کدگذاری با توزیع درجه‌های مختلف در جدول زیر به نمایش گذاشته‌ایم. با توجه به مقادیر مندرج در جدول ۴-۳ می‌توان استدلال کرد که با استفاده از کدگذاری می‌توان فضای ذخیره‌سازی مورد نیاز را از حدود ۴۹ تا ۸۰٪ کاهش داد.

جدول ۴-۳. مقایسه حجم مورد نیاز برای ذخیره‌سازی (مگابایت) برای دو حالت تکرار و کدگذاری

حجم مورد نیاز ذخیره‌سازی (مگابایت)					k
CPRSD	PRSD	RSD	تکرار (چندمنطقه‌ای)	تکرار (تک منطقه‌ای)	
۱۴/۸	۱۵/۱	-	۶۰	۳۰	۱۰۰
۱۲/۷	۱۲/۹۶	۱۳/۹۷	۶۰	۳۰	۲۵۰
۱۱/۷	۱۲/۴	۱۳/۱۵	۶۰	۳۰	۵۰۰

۴-۶ بهبود زمان بازیابی

همان‌طور که پیش‌تر گفته شد برای هر یک از مراکز داده یک صف $M/G/1$ در نظر گرفته‌ایم. به‌منظور

محاسبه زمان بازیابی داده، از زمان درخواست داده تا بارگیری آن از رابطه زیر استفاده می‌کنیم

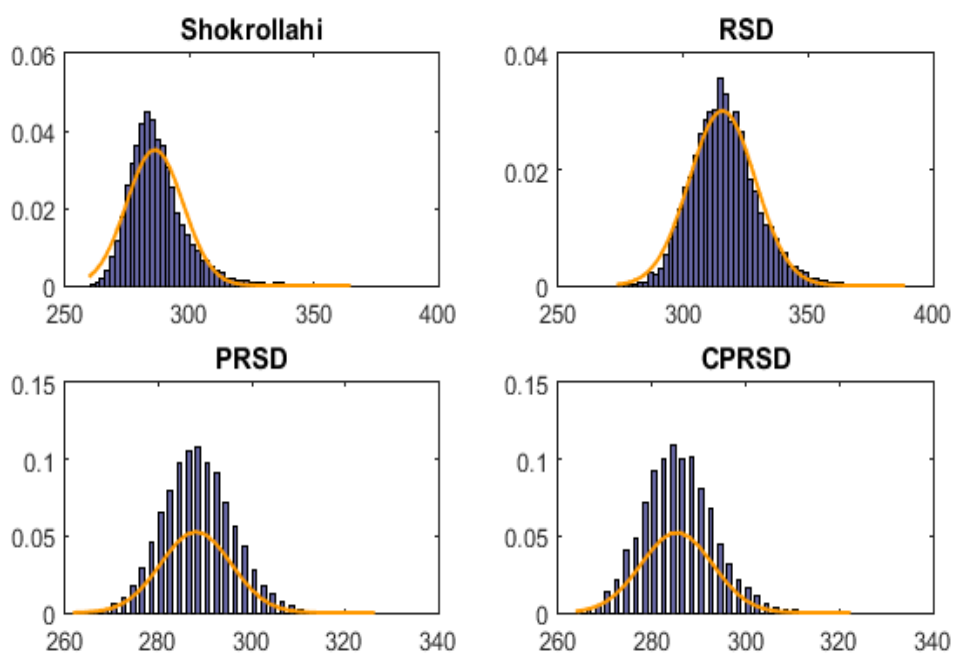
$$T_{total} = T_w + \frac{L}{r} + T_{decoding} \quad (۳-۴)$$

که در آن T_w تأخیر صف، عبارت دوم زمان ارسال داده و $T_{decoding}$ زمانی است که فرایند کدگشایی به‌طول می‌انجامد.

در شبیه‌سازی‌ها، میانگین نرخ ورود به صف مراکز داده به‌صورت تصادفی از توزیع پواسون با میانگین ۵ انتخاب می‌شود. همچنین میانگین و انحراف معیار سرویس‌دهی مراکز داده را در هر دور بازیابی، به‌صورت تصادفی از بین ۲، ۴ و ۶ میلی‌ثانیه برای میانگین و ۱، ۱/۵ و ۲ میلی‌ثانیه برای انحراف معیار انتخاب می‌کنیم. علاوه بر این، یک صف $M/G/1$ دیگر با میانگین نرخ ورود ۱۰ و میانگین و انحراف معیار سرویس‌دهی به ترتیب ۳ و ۱ میلی‌ثانیه برای یک مرکز اصلی در نظر می‌گیریم که این مرکز درخواست‌های بازیابی داده را بر اساس متادیتا آن‌ها به مراکز داده متناظر ارسال می‌کند.

با توجه به میانگین سرعت اینترنت جهانی که در سال ۲۰۱۷ برابر ۷/۲ مگابیت بر ثانیه اعلام شده است [۳۹]، در رابطه فوق r را برابر با این مقدار قرار داده و زمان کدگشایی نیز از شبیه‌سازی‌ها به‌دست می‌آید.

در طول فرایند کدگشایی LT پیش از اعمال بر روی سیستم ذخیره‌سازی ابری دریافتیم که هیستوگرام تعداد نمادهای کدگذاری موردنیاز برای بازیابی موفق داده اصلی، از تابع نرمال تبعیت می‌کند. همان‌طور که در شکل زیر برای $k = ۲۵۰$ و توزیع‌های مختلف آورده شده است.

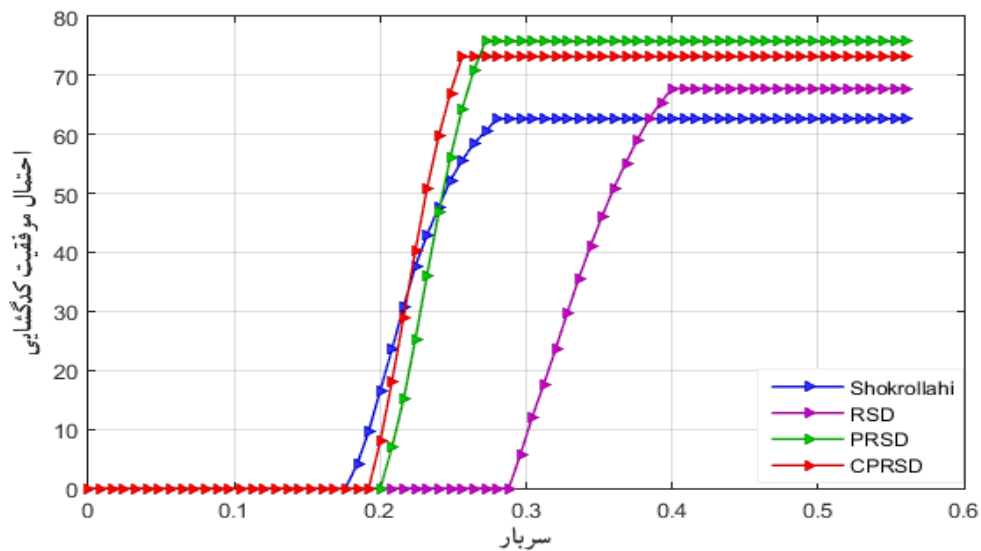


شکل ۴-۲۱. هیستوگرام نمادهای موردنیاز برای کدگشایی به ازای $k = 250$.

به منظور کاهش زمان بازیابی، از این خاصیت بهره برده و در فرایند کدگشایی ابتدا از محدوده یک انحراف معیار از میانگین به دست آمده از تطبیق هیستوگرام با توزیع نرمال (جدول ۴-۴) استفاده کرده ایم. به عبارت دیگر، ۶۸٪ از نمادها که در بازه $(\mu - \sigma, \mu + \sigma)$ قرار دارند به جای بازه (k, n) در نظر گرفته شده است. شکل ۴-۲۲ میزان موفقیت کدگشایی با روش پیشنهادی را نشان می دهد.

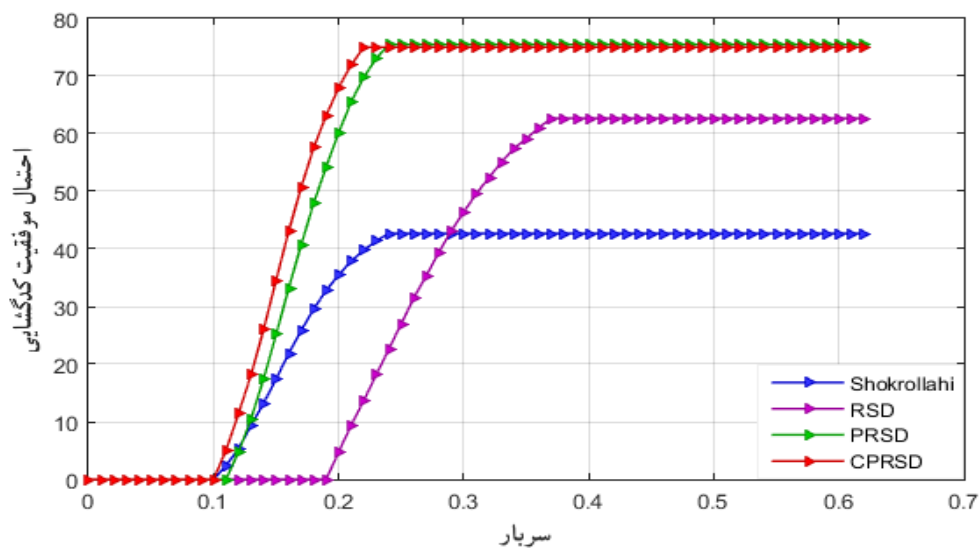
جدول ۴-۴. میانگین و انحراف معیار تعداد نمادهای کدگذاری موردنیاز پس از تطبیق با توزیع نرمال

CPRSD		PRSD		RSD		شکرالهی		k
σ	μ	σ	μ	σ	μ	σ	μ	
۵,۱۴	۱۱۷,۵۵	۵,۲۹	۱۱۹,۰۳	۸,۱	۱۲۹,۶۲	۵,۷۸	۱۱۸,۴۷	۱۰۰
۷,۶	۲۸۵,۲۸	۷,۵۴	۲۸۸	۱۳,۲۲	۳۱۵,۶۱	۱۱,۳۴	۲۸۵,۹۶	۲۵۰
۱۱,۸۲	۵۶۲,۸۷	۱۲,۲۴	۵۶۹,۴۲	۱۹,۳۳	۶۱۷,۶۸	۲۱,۰۲	۵۶۴,۰۴	۵۰۰

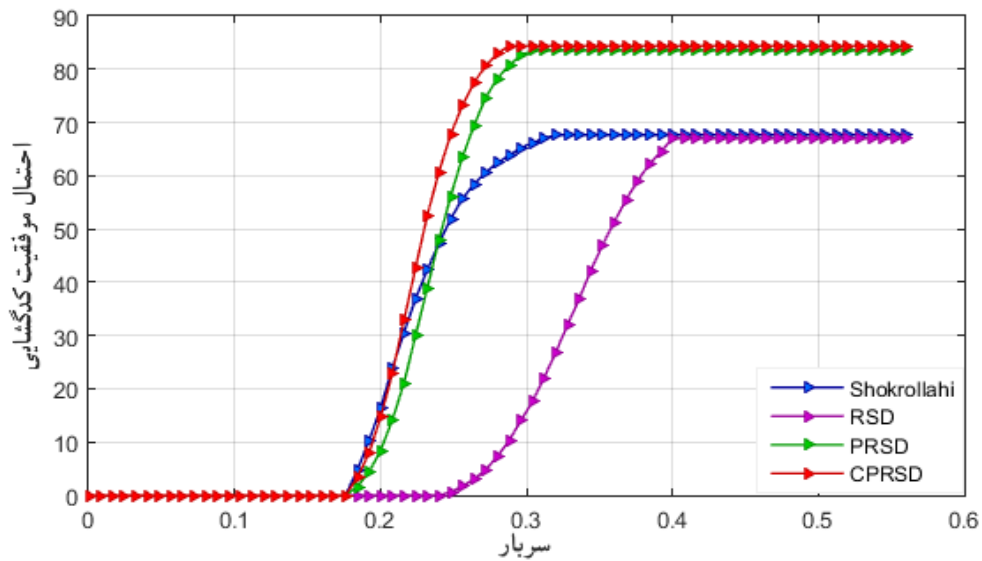


شکل ۴-۲۲. مقایسه احتمال موفقیت کدگذاری LT با توزیع‌های مختلف با استفاده از یک انحراف معیار از میانگین برای $k = 250$.

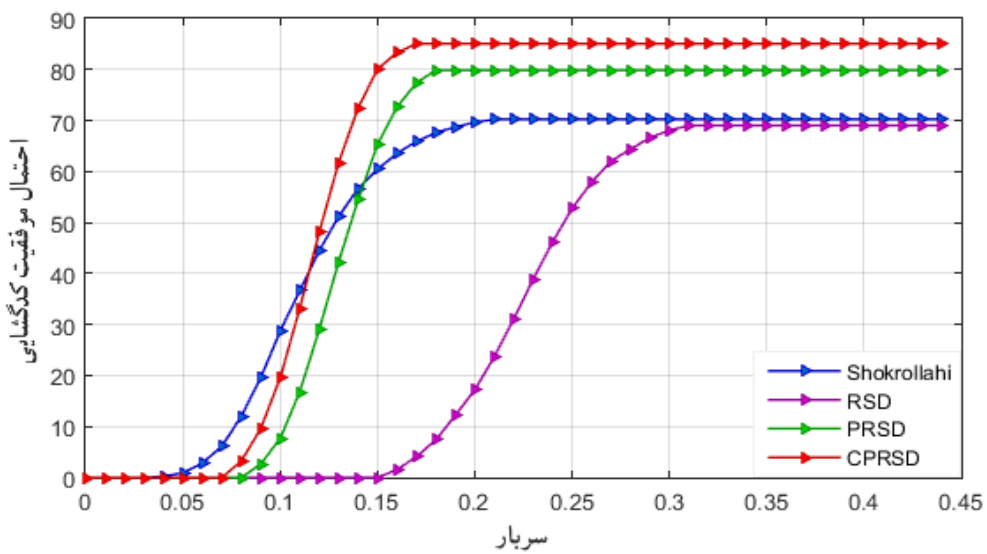
اگرچه این ایده توانسته است زمان موردنیاز برای بازیابی را بسیار کاهش دهد، با این وجود به علت کاهش نسبتاً زیاد موفقیت بازیابی، از بازه $(\mu - 2\sigma, \mu + 2\sigma)$ در ادامه استفاده خواهیم کرد که ۹۵٪ نمادهای کدگذاری را در برمی‌گیرد.



شکل ۴-۲۳. مقایسه احتمال موفقیت کدگذاری LT با توزیع‌های مختلف با استفاده از دو انحراف معیار از میانگین برای $k = 100$.



شکل ۴-۲۴. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف با استفاده از دو انحراف معیار از میانگین برای $k = 250$.



شکل ۴-۲۵. مقایسه احتمال موفقیت کدگشایی LT با توزیع‌های مختلف با استفاده از دو انحراف معیار از میانگین برای $k = 500$.

موفقیت حاصل از این ایده در شکل‌های فوق و در جدول ۴-۵ قابل مشاهده است که مقایسه زمانی عملکرد سیستم پیش از اعمال این الگوریتم و با به کارگیری آن را نشان می‌دهد. در این جدول، اولین زمان در بخش پیشنهادی مربوط به بازه $(\mu - \sigma, \mu + \sigma)$ و دومین عدد بیانگر زمان مورد نیاز برای بازیابی با استفاده از بازه $(\mu - 2\sigma, \mu + 2\sigma)$ است.

جدول ۴-۵. مقایسه زمان بازیابی (ثانیه) در حالت اصلی (استفاده از k و n) و الگوریتم پیشنهادی

زمان بازیابی (ثانیه)								
CPRSD		PRSD		RSD		شکرالهی		k
پیشنهادی	اصلی	پیشنهادی	اصلی	پیشنهادی	اصلی	پیشنهادی	اصلی	
۰,۳۷۷۰	۰,۷۳۸۷	۰,۱۲۴۸	۰,۲۰۶۰	۰,۳۵۳۸	۰,۸۱۸۵	۰,۱۷۰۹	۰,۳۸۲۵	۱۰۰
۰,۴۳۵۴		۰,۱۷۶۷		۰,۴۴۵۱		۰,۲۶۹۷		
۱,۳۲۹۳	۴,۳۸۵۷	۰,۵۷۱۶	۱,۹۹۳۴	۱,۵۰۸۷	۴,۷۷۶۲	۰,۸۷۳۱	۲,۱۱۰۱	۲۵۰
۱,۳۸۰۶		۰,۶۹۲۴		۱,۷۱۷۴		۱,۳۰۹۷		
۳,۲۲۰۵	۹,۲۵۷۹	۱,۰۲۷۰	۳,۳۸۶۲	۳,۱۰۵۳	۹,۱۶۰۵	۱,۴۳۷۴	۲,۷۹۸۷	۵۰۰
۴,۹۳۵۸		۱,۵۰۱۰		۴,۷۵۵۴		۲,۱۶۶۷		

همان‌طور که در شکل‌ها مشاهده می‌شود، پس از اعمال الگوریتم پیشنهادی با دو انحراف معیار از میانگین کاهش احتمال موفقیت بازیابی بسیار کم است در حالی که زمان بازیابی کاهش چشم‌گیری داشته است. بنابراین با استفاده از این روش پیشنهادی و همچنین دو توزیع CPRSD و PRSD می‌توان عملکرد سیستم‌های ذخیره‌سازی ابری را از نظر میزان موفقیت بازیابی داده، زمان بازیابی و همچنین فضای موردنیاز برای ذخیره‌سازی بهبود بخشید.

فصل ۵ : نتیجه گیری و پیشنهاد کارهای آینده

۵-۱ نتیجه گیری

در این پایان نامه ابتدا ضرورت موضوع ذخیره‌سازی ابری توضیح داده شد و روش پیشنهادی به‌منظور بهبود این سیستم‌ها مطرح گردید. در ادامه کارهای پژوهشی و عملی انجام شده در دو زمینه کد گذاری منبع، کدگذاری کانال و روند انتخاب روش‌های متناسب با ذخیره‌سازی ابری تا انتخاب کدهای بدون نرخ در سال‌های اخیر مورد بررسی قرار گرفت. در ادامه یک سیستم ذخیره‌سازی ابری در مقیاس کوچک بر اساس ساختار واقعی این سیستم‌ها در نظر گرفته شد. به‌منظور فشرده‌سازی داده پیش از برون‌سپاری روی ابر از معیار جدید MIM استفاده کردیم و با شبیه‌سازی و نتایج عددی نشان دادیم که می‌توان در مواردی که ضرورت، محدودیت یا انتخاب کاربر حکم می‌کند از این روش به‌منظور کاهش فضای ذخیره‌سازی استفاده کرد.

کدهای بدون نرخ LT برای پیاده‌سازی روی سیستم ذخیره‌سازی ابری پیشنهادی در نظر گرفته شد و از آنجایی عملکرد این کدها بسیار به توزیع درجه آن‌ها وابسته است، سه توزیع جدید شکرالهی، CPRSD، PRSD را علاوه بر توزیع RSD به‌منظور بهبود این سیستم‌ها بیان کردیم. شبیه‌سازی‌های اولیه برای به‌دست آوردن مؤلفه‌های مناسب انجام شده و در ادامه به بررسی عملکرد توزیع‌های جدید پرداختیم. با استفاده از شبیه‌سازی و نتایج عددی حاصل از آن‌ها نشان دادیم که توزیع‌های جدید تأثیر چشم‌گیری در میزان موفقیت کدگشایی و بازیابی داده دارند؛ سربار موردنیاز را کاهش داده و در نتیجه به بهبود زمان بازیابی داده کمک می‌کنند. مقایسه بین میزان موفقیت کدگشایی برای توزیع های ذکر شده به‌ازای تعداد نمادهای اصلی مختلف و در شرایط عدم حذف مراکز داده و حذف یک و دو مرکز داده صورت پذیرفت. همچنین به مقایسه فضای موردنیاز برای ذخیره‌سازی در دو مدل مبتنی بر تکرار و مبتنی بر کدگذاری و میزان موفقیت بازیابی در دو حالت تک‌منطقه‌ای و چندمنطقه‌ای پرداختیم. علاوه بر این، با مشاهده تجربی از نتایج شبیه‌سازی‌ها دریافتیم که تعداد نماد کدگذاری موردنیاز برای کدگشایی موفق داده از توزیع نرمال پیروی می‌کند. با تطبیق توزیع نرمال و

به دست آوردن میانگین و انحراف معیار، الگوریتم جدیدی را جایگزین کرده و نشان دادیم که می توان زمان بازیابی را با به کارگیری آن کاهش داد.

۲-۵ پیشنهاد کارهای آینده

عملکرد کدهای جدید دیگری از جمله کدهای قطبی و کدهای غیرخطی بر روی سیستم های ذخیره سازی ابری بررسی گردد.

انتخاب نمادهای اصلی برای تولید یک نماد کدگذاری به جای تصادفی یکنواخت، بر اساس توزیعی صورت گیرد و عملکرد این انتخاب در این سیستم ها مورد بررسی قرار گیرد.

سوت

جدول ۱-۵. فاصله نواحی دسترسی تا مکان درخواست کاربر برای ذخیره و بازیابی داده

منطقه	ناحیه دسترسی	تهران	نوویسبرسک	استانبول	آلماتی	شانگهای	اوساکا	بانکوک	کوالالامپور	جاکارتا	کلکته
ژاپن	توکیو ۱	۷۶۷۰	۴۷۶۹	۱۲۲۰۸	۶۷۷۲	۴۰۱۴	۴۹۴	۷۰۲۸	۸۴۹۹	۵۷۸۸	۷۷۶۷
	توکیو ۲	۷۶۷۵	۴۷۷۴	۱۲۲۱۳	۶۷۷۷	۴۰۱۹	۴۹۹	۷۰۳۳	۸۴۹۴	۵۷۹۳	۷۷۷۲
	اوساکا	۷۳۹۰	۴۵۸۲	۱۱۷۳۵	۶۲۹۹	۳۵۴۱	۰	۶۵۵۵	۸۰۲۱	۵۴۵۴	۷۲۹۴
سنگاپور و فیلیپین	سنگاپور ۱	۶۶۱۴	۶۲۶۶	۱۱۶۶۶	۸۳۲۵	۵۴۷۸	۸۳۹۵	۱۸۲۷	۳۵۴	۹۰۷	۴۵۱۴
	سنگاپور ۲	۶۶۱۹	۶۲۷۱	۱۱۶۷۱	۸۳۳۰	۵۴۷۳	۸۳۹۰	۱۸۳۲	۳۵۹	۹۰۲	۴۵۱۹
	مانیل	۷۲۲۴	۵۵۶۰	۹۱۲۴	۵۲۶۰	۱۸۵۲	۲۶۶۹	۲۲۱۲	۲۴۷۳	۲۷۸۹	۳۵۴۶
کره جنوبی و	سنول ۱	۶۵۶۱	۳۸۱۷	۱۰۵۱۱	۵۰۷۴	۲۳۱۶	۱۲۲۱	۵۳۳۵	۶۷۹۶	۵۲۹۸	۶۰۶۹
	سنول ۲	۶۵۶۶	۳۸۲۲	۱۰۵۱۶	۵۰۷۹	۲۳۲۱	۱۲۲۶	۵۳۳۰	۶۸۰۱	۵۳۰۳	۶۰۷۴
تایوان	تایپه	۶۷۰۵	۴۵۸۴	۸۴۲۸	۴۵۲۳	۶۹۰	۱۷۲۰	۲۵۳۸	۳۲۳۷	۳۸۲۲	۳۳۸۴
چین	بیجینگ ۱	۵۶۱۱	۲۹۸۵	۹۲۱۳	۳۷۷۷	۱۲۲۰	۲۵۲۸	۴۱۰۵	۵۵۷۳	۵۲۲۷	۴۷۸۹
	بیجینگ ۲	۵۶۱۶	۲۹۹۰	۹۲۱۸	۳۷۸۲	۱۲۱۵	۲۵۲۳	۴۱۱۰	۵۵۷۸	۵۲۲۲	۴۷۹۴
	هنگ کنگ	۶۱۷۸	۴۴۵۵	۱۰۸۲۰	۵۴۰۳	۱۴۶۰	۴۶۷۰	۲۴۶۱	۳۹۲۷	۳۲۷۰	۴۱۴۴
هند	بمبئی ۱	۲۸۰۴	۴۰۹۵	۶۳۷۰	۲۶۳۵	۶۷۶۱	۵۵۲۳	۴۷۵۲	۶۰۶۶	۴۶۷۳	۱۹۱۹
	بمبئی ۲	۲۸۰۹	۴۰۹۰	۶۳۷۵	۲۶۳۰	۶۷۶۶	۵۵۱۸	۴۷۵۷	۶۰۶۱	۴۶۶۸	۱۹۱۴
	دهلی نو	۲۵۴۹	۲۹۷۴	۵۷۰۵	۳۰۵۳	۶۰۱۳	۸۱۲۷	۴۲۷۳	۵۵۸۱	۵۰۱۱	۱۴۸۳

مراج

۲. احمدی، م.ر.، آریانیان، ا.، ملکی، د. (۱۳۹۴) "اصول مجازی‌سازی و رایانش ابری" چاپ اول، ویراست دوم، انتشارات نیاز دانش، ص ۱۱۲ و ۱۱۳

- [1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," 2012.
- [2] <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf> .
- [4] C.E.Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 397-423,623-656, 1948.
- [5] T .M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
- [6] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098-1101, 1952.
- [7] D. Pelleg and A. W. Moore, "Active learning for anomaly and rare-category detection," in *Advances in neural information processing systems*, pp. 1073-1080, 2005.
- [8] S. Bay, K. Kumaraswamy, M. G. Anderle, R. Kumar, and D. M. Steier, "Large scale detection of irregularities in accounting data," in *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pp. 75-86: IEEE,2006.
- [9] A. Zieba, "Counterterrorism systems of spain and poland: Comparative studies," *Przegląd Politologiczny*, vol. 3, pp. 65-78, 2015.
- [10] P. Fan, Y. Dong, J. Lu, and S. Liu, "Message importance measure and its application to minority subset detection in big data," in *Globecom Workshops (GC Wkshps), 2016 IEEE*, pp. 1-5: IEEE, 2016.
- [11] R. She, S. Liu, Y. Dong, and P. Fan, "Focusing on a probability element: Parameter selection of message importance measure in big data," in *Communications (ICC), 2017 IEEE International Conference on*, pp. 1-6: IEEE, 2017.
- [12] S. Liu, R. She, P. Fan, and K. B. Letaief, "Non-parametric Message Important Measure: Storage Code Design and Transmission Planning for Big Data," *arXiv preprint arXiv:1709.10280*, 2017.
- [13] R. She, S. Liu, and P. Fan, "State Variation Mining: On Information Divergence with Message Importance in Big Data," *arXiv preprint arXiv:1801.04064*, 2018.
- [14] S. Liu, R. She, and P. Fan, "How Many Samples Required in Big Data Collection: A Differential Message Importance Measure," *arXiv preprint arXiv:1801.04063*, 2018.

- [15] H. Weatherspoon and J. D. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," in *International Workshop on Peer-to-Peer Systems*, pp. 328-337: Springer,. 2002
- [16] G. Joshi, Y. Liu, and E. Soljanin, "Coding for fast content download," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pp. 326-333: IEEE., 2012
- [17] G. Joshi, Y. Liu, and E. Soljanin, "On the delay-storage trade-off in content download from coded distributed storage systems," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 989-997, 2014.
- [18] A. Kumar, R. Tandon, and T. C. Clancy, "On the latency and energy efficiency of erasure-coded cloud storage systems," *arXiv preprint arXiv:1405.2833*, 2014.
- [19] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300-304, 1960.
- [20] https://cloud.google.com/files/storage_architecture_and_challenges.pdf .
- [21] C. Huang *et al.*, "Erasure Coding in Windows Azure Storage," in *Usenix annual technical conference* , pp. 15-26: Boston, MA, 2012.
- [22] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539-4551, 2010.
- [23] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 476-489, 2011.
- [24] H. Xia and A. A. Chien, "RobuSTore: a distributed storage architecture with robust and high performance," in *Supercomputing, 2007. SC'07. Proceedings of the 2007 ACM/IEEE Conference on*, pp. 1-11: IEEE,2007.
- [25] M. Luby, "LT codes," in *null*, pp. 271-280: IEEE,2002.
- [26] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. T. Hou, "Lt codes-based secure and reliable cloud storage service," in *INFOCOM, 2012 Proceedings IEEE*, pp. 693-701: IEEE, 2012.
- [27] C. Anglano, R. Gaeta, and M. Grangetto, "Exploiting rateless codes in cloud storage systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1313-1322, 2015.
- [28] H. Lu, C. H. Foh, Y. Wen, and J. Cai, "Delay-Optimized File Retrieval under LT-Based Cloud Storage," 2015.
- [29] <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-best-practices.html> .
- [30] <https://cloud.google.com/storage/docs/storage-classes> .
- [31] <https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction> .

- [32] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*. Prentice-Hall International New Jersey, 1992.
- [33] R. Hoffman, *Data compression in digital systems*. Springer Science & Business Media, 2012.
- [34] A. Shokrollahi, "Raptor codes," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2551-2567, 2006.
- [35] M. Luby, "A note on the design of degree distributions," *Private Communication*, unpublished 2001.
- [36] W. Yao, B. Yi, T. Huang, and W. Li, "Poisson Robust Soliton Distribution (PRSD) for LT Codes," *IEEE Communications Letters*, vol. 20, no. 8, 2016.
- [37] S. Q. X. Z. Sheng, and C.Y. Pan, *Probability theory and mathematical statistics*. Zhejiang, China: Zhejiang university press, 2010.
- [38] W. Yao, B. Yi, W. Li, T. Huang, and Q. Xie, "CPRSD for LT codes," *IET Communications*, vol. 10, no. 12, pp. 1411-1415, 2016.
- [39] <https://www.statista.com/statistics/204952/average-internet-connection-speed-by-country> ./

Abstract

Due to significant growth of data and predictions of this trend continuity, offering appropriate approaches for data storage is more necessary than before. By getting started of cloud storage systems in 2006, developing and improvement of this systems is highly considered. In this thesis, we improve the performance of cloud storage by representing novel methods.

First, by exploiting new criterion “Message Importance Measure” that is used in minority subset detection, we compress data before outsourcing on cloud. With comparison of this method with Huffman, Shannon and Shannon Fano Elias, in cases that important part of message is considered, we can reduce storage space up to 30 percent. Also we investigate effect of symbols removing thresholds in this compression technique.

In second part of thesis, we implement coding-based cloud storage with LT codes and in order to improve performance of this systems in data retrieval, we use Shokrollahi, PRSD, CPRSD degree distributions in addition to RSD. By applying new distributions, we demonstrate that by applying new distributions, successful retrieval probability could be increased with fewer overhead. Moreover, we reduce retrieval time by using an algorithm based on average and standard deviation of needed encoding symbols instead of original symbols in decoding process.

Keywords: Cloud Storage, Message Importance Measure, Coding-Based Storage, LT Codes, Degree Distribution



Faculty of Electrical Engineering and Robotic

M.Sc. Thesis in Communication Systems Engineering

A Novel Coding Approach to Improve Performance of Cloud Storage Systems

By: Nastaran Chakani

Supervisor:
Dr. Masoud Mirrezaei

Advisor:
Dr. Ghosheh Abed Hodtani

September 2018