

*

بررسی اجمالی کد گذاری برای سیستم های ذخیره سازی شده
توزیع شده

زهرا آقاعلیزاده

استاد راهنما:

دکتر احمد یوسفیان دارانی

دانشگاه محقق اردبیلی

بهمن 1399

مقدمه

نظریه کدگذاری یک مدل ریاضی است با چنین هدفی که با تکیه بر دانش ریاضی از اطلاعات در زمان ارسال آنها محافظت می‌کند. به عبارت دیگر نظریه کدگذاری از ریاضیات برای افزودن افزونگی به یک فایل از طریق اطلاعات به روشی بسیار موثر استفاده می‌کند.

مقدمه

نظریه کدگذاری یک مدل ریاضی است با چنین هدفی که با تکیه بر دانش ریاضی از اطلاعات در زمان ارسال آنها محافظت می‌کند. به عبارت دیگر نظریه کدگذاری از ریاضیات برای افزودن افزونگی به یک فایل از طریق اطلاعات به روشی بسیار موثر استفاده می‌کند.

فرض کنید هدف ما فرستادن واژه‌ی HELLO در یک کانال باشد. فرض کنید که در روند ارسال این پیام یکی از حروف تغییر کند. مثلاً HALLO در گیرنده دریافت شده باشد.

مقدمه

نظریه کدگذاری یک مدل ریاضی است با چنین هدفی که با تکیه بر دانش ریاضی از اطلاعات در زمان ارسال آنها محافظت می‌کند. به عبارت دیگر نظریه کدگذاری از ریاضیات برای افزودن افزونگی به یک فایل از طریق اطلاعات به روشی بسیار موثر استفاده می‌کند.

فرض کنید هدف ما فرستادن واژه‌ی HELLO در یک کانال باشد. فرض کنید که در روند ارسال این پیام یکی از حروف تغییر کند. مثلاً HALLO در گیرنده دریافت شده باشد.

یکی از راه‌های شناسایی خطای ایجاد شده این است که پیام HELLO بیش از

یکی از راه‌های شناسایی خطای ایجاد شده این است که پیام HELLO بیش از یکبار ارسال شود. احتمال این که در هر بار ارسال پیام مجدداً حرف E تغییر کند یا اینکه حرف E هر بار به A تغییر پیدا کند یک احتمال ضعیف خواهد بود.

یکی از راه‌های شناسایی خطای ایجاد شده این است که پیام HELLO بیش از یکبار ارسال شود. احتمال این که در هر بار ارسال پیام مجدداً حرف E تغییر کند یا اینکه حرف E هر بار به A تغییر پیدا کند یک احتمال ضعیف خواهد بود. به عنوان مثال در دومین ارسال پیام ZELLO دریافت شود. سه حرف آخر در هر دو ارسال ثابت هستند. لذا با احتمال زیاد آنها به درستی مخابره شده‌اند.

مقدمه

یکی از راه‌های شناسایی خطای ایجاد شده این است که پیام HELLO بیش از یکبار ارسال شود. احتمال این که در هر بار ارسال پیام مجدداً حرف E تغییر کند یا اینکه حرف E هر بار به A تغییر پیدا کند یک احتمال ضعیف خواهد بود. به عنوان مثال در دومین ارسال پیام ZELLO دریافت شود. سه حرف آخر در هر دو ارسال ثابت هستند. لذا با احتمال زیاد آن‌ها به درستی مخابره شده‌اند. دو حرف نخست در هر دو ارسال متفاوت هستند اما احتمال این که هر دو همزمان اشتباه ارسال شده باشند ضعیف است. بنابراین می‌توانیم حدس بزنیم که پیام اصلی یا HELLO است و یا ZALLO. با ارسال مجدد پیام این موضوع می‌تواند روشن شود که پیام اصلی چه بوده است.

این روش اگر چه در این مثال کارآمد به نظر می‌رسد ولی در حالت کلی روشی بسیار

این روش اگر چه در این مثال کارآمد به نظر می‌رسد ولی در حالت کلی روشی بسیار ابتدایی بر هزینه و ناکارآمد است.

در حقیقت استفاده از روش ارسال مکرر اطلاعات مستلزم بکارگیری حافظه بسیار از کامپیوتر است که بتوانیم شانس شناسایی پیام درست را داشته باشیم.

مقدمه

این روش اگر چه در این مثال کارآمد به نظر می‌رسد ولی در حالت کلی روشی بسیار ابتدایی پر هزینه و ناکارآمد است.

در حقیقت استفاده از روش ارسال مکرر اطلاعات مستلزم بکارگیری حافظه بسیار از کامپیوتر است که بتوانیم شانس شناسایی پیام درست را داشته باشیم.

همین‌گ و سایر همکاران در این زمینه تلاش خود را بر این اساس گذاشتند که

روش‌هایی را مطرح کنند که علاوه بر اینکه شانس شناسایی پیام درست را بالا

می‌برند همزمان حافظه کمتری را اشغال کنند. سیستم توزیع‌شده مجموعه‌ای از

چندین کامپیوتر مستقل است که به وسیله‌ی یک واسطه به یکدیگر متصل شده‌اند و

این میان افزار است که منابع را بین این کامپیوترها به اشتراک می‌گذارد.

کد خطی

کدخطی

هر زیرمجموعه M عضوی از F_q مانند C را یک کدخطی روی F_q می‌نامیم.

کدخطی

هر زیرمجموعه M عضوی از F_q مانند C را یک کدخطی روی F_q می‌نامیم.
عدد n را طول C و M را اندازه C می‌گوییم و می‌نویسیم C یک (n, M) -کد است.

کدخطی

هر زیرمجموعه M عضوی از F_q مانند C را یک کدخطی روی F_q می‌نامیم. عدد n را طول C و M را اندازه C می‌گوییم و می‌نویسیم C یک (n, M) -کد است. عناصر C را کدواژه می‌نامیم. به ازای هر $x = (x_1, \dots, x_n)$ و $y = (y_1, \dots, y_n)$ در F_q^n فاصله (همینگ) x

کدخطی

هر زیرمجموعه M عضوی از F_q مانند C را یک کدخطی روی F_q می‌نامیم. عدد n را طول C و M را اندازه C می‌گوییم و می‌نویسیم C یک (n, M) -کد است. عناصر C را کدواژه می‌نامیم.

به ازای هر $x = (x_1, \dots, x_n)$ و $y = (y_1, \dots, y_n)$ در F_q^n فاصله (همینگ) x و y را با $d(x, y)$ نشان داده و برابر است با تعداد i هایی که $x_i \neq y_i$. قرار می‌دهیم:

$$d = d(C) = \min\{(x, y) : x, y \in C, x \neq y\}$$

کدخطی

هر زیرمجموعه M عضوی از F_q مانند C را یک کدخطی روی F_q می‌نامیم. عدد n را طول C و M را اندازه C می‌گوییم و می‌نویسیم C یک (n, M) -کد است. عناصر C را کدواژه می‌نامیم.

به ازای هر $x = (x_1, \dots, x_n)$ و $y = (y_1, \dots, y_n)$ در F_q^n فاصله (همینگ) x و y را با $d(x, y)$ نشان داده و برابر است با تعداد i هایی که $x_i \neq y_i$. قرار می‌دهیم:

$$d = d(C) = \min\{(x, y) : x, y \in C, x \neq y\}$$

در اینصورت d را مینیمم فاصله C می‌گوییم و می‌نویسیم C یک (n, m, d) -کد است.

کدخطی

هر زیرمجموعه M عضوی از F_q مانند C را یک کدخطی روی F_q می‌نامیم. عدد n را طول C و M را اندازه C می‌گوییم و می‌نویسیم C یک (n, M) -کد است. عناصر C را کدواژه می‌نامیم.

به ازای هر $x = (x_1, \dots, x_n)$ و $y = (y_1, \dots, y_n)$ در F_q^n فاصله (همینگ) x و y را با $d(x, y)$ نشان داده و برابر است با تعداد i هایی که $x_i \neq y_i$. قرار می‌دهیم:

$$d = d(C) = \min\{(x, y) : x, y \in C, x \neq y\}$$

در اینصورت d را مینیمم فاصله C می‌گوییم و می‌نویسیم C یک (n, m, d) -کد است.

کد خطی

کد خطی

C را یک کد خطی با طول n روی F_q می‌گوییم هرگاه C یک زیر فضای برداری F_q^n باشد، در واقع خطی بودن C به این معنی است که به ازای هر $\alpha, \beta \in C$ و هر

کد خطی

C را یک کد خطی با طول n روی F_q می‌گوییم هرگاه C یک زیر فضای برداری F_q^n باشد، در واقع خطی بودن C به این معنی است که به ازای هر $\alpha, \beta \in C$ و هر $a \in F_q$ ، $a\alpha + \beta \in C$.

اگر C یک زیر فضای برداری k بعدی از F_q^n باشد آنگاه از جبر مقدماتی می‌دانیم که $C = F_q^k$ و لذا $M = |C| = q^k$. در این حالت می‌گوییم C یک $[n, k, d]$ -کد است.

کد خطی

C را یک کد خطی با طول n روی F_q می‌گوییم هرگاه C یک زیر فضای برداری F_q^n باشد، در واقع خطی بودن C به این معنی است که به ازای هر $\alpha, \beta \in C$ و هر $a \in F_q$

$$a\alpha + \beta \in C$$

اگر C یک زیر فضای برداری k بعدی از F_q^n باشد آنگاه از جبر مقدماتی می‌دانیم که $C = F_q^k$ و لذا $M = |C| = q^k$. در این حالت می‌گوییم C یک $[n, k, d]$ -کد است. هر گاه نیازی نباشد که به d تاکید کنیم می‌نویسیم C یک $[n, k]$ -کد است. در تعریف بالا اگر $k = 0$ آنگاه $c = 0$ تنها شامل کدواژه صفر است و کد بدیهی نامیده می‌شود. اگر $n = k$ آنگاه $C = F_q^n$ تمام فضای برداری است.

کد خطی

C را یک کد خطی با طول n روی F_q می‌گوییم هرگاه C یک زیر فضای برداری F_q^n باشد، در واقع خطی بودن C به این معنی است که به ازای هر $\alpha, \beta \in C$ و هر $a \in F_q$

$$a\alpha + \beta \in C$$

اگر C یک زیر فضای برداری k بعدی از F_q^n باشد آنگاه از جبر مقدماتی می‌دانیم که $C = F_q^k$ و لذا $M = |C| = q^k$. در این حالت می‌گوییم C یک $[n, k, d]$ -کد است. هر گاه نیازی نباشد که به d تاکید کنیم می‌نویسیم C یک $[n, k]$ -کد است. در تعریف بالا اگر $k = 0$ آنگاه $c = 0$ تنها شامل کدواژه صفر است و کد بدیهی نامیده می‌شود. اگر $n = k$ آنگاه $C = F_q^n$ تمام فضای برداری است.

قانون کدگشایی

قانون کدگشایی

در یک کانال ارتباطی با کدگذاری، فقط کدواژه‌ها ارسال می‌شود. فرض کنید کلمه W دریافت شده است اگر W یک کدواژه صحیح باشد ممکن است نتیجه بگیریم که هیچ خطایی در ارسال وجود ندارد در غیر اینصورت می‌دانیم که خطایی رخ داده است. در این حالت برای پیدا کردن کدواژه ارسال شده با احتمال زیاد به یک قانون نیاز داریم چنین قانونی به عنوان قانون کدگشایی شناخته می‌شود.

قانون کدگشایی

در یک کانال ارتباطی با کدگذاری، فقط کدواژه‌ها ارسال می‌شود. فرض کنید کلمه W دریافت شده است اگر W یک کدواژه صحیح باشد ممکن است نتیجه بگیریم که هیچ خطایی در ارسال وجود ندارد در غیر اینصورت می‌دانیم که خطایی رخ داده است. در این حالت برای پیدا کردن کدواژه ارسال شده با احتمال زیاد به یک قانون نیاز داریم چنین قانونی به عنوان قانون کدگشایی شناخته می‌شود.

افزونگی

افزونگی

افزونگی در سیستم ذخیره‌سازی به معنای ذخیره‌سازی همزمان ویدئو یا هر نوع اطلاعات دیگر در چند مکان مختلف و تهیه‌ی نسخه پشتیبان از آنهاست تا در صورت خرابی سرویس‌دهنده ذخیره‌سازی اصلی، از آنها استفاده شود.

افزونگی

افزونگی در سیستم ذخیره‌سازی به معنای ذخیره‌سازی همزمان ویدئو یا هر نوع اطلاعات دیگر در چند مکان مختلف و تهیه‌ی نسخه پشتیبان از آنهاست تا در صورت خرابی سرویس‌دهنده ذخیره‌سازی اصلی، از آنها استفاده شود.

پهنای باند

افزونگی

افزونگی در سیستم ذخیره‌سازی به معنای ذخیره‌سازی همزمان ویدئو یا هر نوع اطلاعات دیگر در چند مکان مختلف و تهیه‌ی نسخه پشتیبان از آن‌هاست تا در صورت خرابی سرویس‌دهنده ذخیره‌سازی اصلی، از آن‌ها استفاده شود.

پهنای باند

پهنای باند حداکثر میزان اطلاعاتی است که می‌تواند در لحظه بر روی شبکه ارتباطی

افزونگی

افزونگی در سیستم ذخیره‌سازی به معنای ذخیره‌سازی همزمان ویدئو یا هر نوع اطلاعات دیگر در چند مکان مختلف و تهیه‌ی نسخه پشتیبان از آن‌هاست تا در صورت خرابی سرویس‌دهنده ذخیره‌سازی اصلی، از آن‌ها استفاده شود.

پهنای باند

پهنای باند حداکثر میزان اطلاعاتی است که می‌تواند در لحظه بر روی شبکه ارتباطی منتقل شود.

گراف جریان اطلاعاتی

جریان



گراف جریان اطلاعاتی

جریان

در یک شبکه یا گراف $G = (V, E, c)$ با دو رأس مشخص $s \in V$ به عنوان مبدأ و $t \in V$ به عنوان مقصد، یک جریان از رأس s به t ، تابعی است که به هر یال e مقدار نامنفی را نسبت می‌دهد، $f : E \rightarrow \mathbb{R}_0^+$ ، $f(e)$ نشان دهنده‌ی میزان جریانی است که یال e حمل می‌کند. جریان f دو ویژگی زیر را برآورد خواهد کرد:

گراف جریان اطلاعاتی

جریان

در یک شبکه یا گراف $G = (V, E, c)$ با دو رأس مشخص $s \in V$ به عنوان مبدأ و $t \in V$ به عنوان مقصد، یک جریان از رأس s به t ، تابعی است که به هر یال e مقدار نامنفی را نسبت می‌دهد، $f : E \rightarrow \mathbb{R}_0^+$ ، $f(e)$ نشان دهنده‌ی میزان جریانی است که یال e حمل می‌کند. جریان f دو ویژگی زیر را برآورد خواهد کرد:

$$(1) \text{ محدودیت ظرفیت: } \forall e \in E, 0 \leq f(e) \leq c(e)$$

گراف جریان اطلاعاتی

جریان

در یک شبکه یا گراف $G = (V, E, c)$ با دو رأس مشخص $s \in V$ به عنوان مبدأ و $t \in V$ به عنوان مقصد، یک جریان از رأس s به t ، تابعی است که به هر یال e مقدار نامنفی را نسبت می‌دهد، $f: E \rightarrow \mathbb{R}_0^+$ ، $f(e)$ نشان دهنده‌ی میزان جریانی است که یال e حمل می‌کند. جریان f دو ویژگی زیر را برآورد خواهد کرد:

$$(1) \text{ محدودیت ظرفیت: } \forall e \in E, 0 \leq f(e) \leq c(e)$$

$$(2) \text{ حفظ جریان:}$$

$$\forall u \in V - \{s, t\}, \sum_{v:(v,u) \in E} f(v, u) = \sum_{v:(u,v) \in E} f(u, v)$$

گراف جریان اطلاعاتی

جریان

در یک شبکه یا گراف $G = (V, E, c)$ با دو رأس مشخص $s \in V$ به عنوان مبدأ و $t \in V$ به عنوان مقصد، یک جریان از رأس s به t ، تابعی است که به هر یال e مقدار نامنفی را نسبت می‌دهد، $f: E \rightarrow \mathbb{R}_0^+$ ، $f(e)$ نشان دهنده‌ی میزان جریانی است که یال e حمل می‌کند. جریان f دو ویژگی زیر را برآورد خواهد کرد:

$$(1) \text{ محدودیت ظرفیت: } \forall e \in E, 0 \leq f(e) \leq c(e)$$

$$(2) \text{ حفظ جریان:}$$

$$\forall u \in V - \{s, t\}, \sum_{v:(v,u) \in E} f(v, u) = \sum_{v:(u,v) \in E} f(u, v)$$

گراف جریان اطلاعاتی

یال جریان

گراف جریان اطلاعاتی

یال جریان

به یال e در شبکه $G = (V, E, c)$ که $f(e) > 0$ ، یال جریان می‌گوییم.

گراف جریان اطلاعاتی

یال جریان

به یال e در شبکه $G = (V, E, c)$ که $f(e) > 0$ ، یال جریان می‌گوییم.

ظرفیت مسیر

گراف جریان اطلاعاتی

یال جریان

به یال e در شبکه $G = (V, E, c)$ که $f(e) > 0$ ، یال جریان می‌گوییم.

ظرفیت مسیر

بیشترین میزان جریانی را که می‌توان از یک مسیر افزایشی p عبور داد، ظرفیت آن مسیر گویند که به صورت زیر محاسبه می‌شود:

گراف جریان اطلاعاتی

یال جریان

به یال e در شبکه $G = (V, E, c)$ که $f(e) > 0$ ، یال جریان می‌گوییم.

ظرفیت مسیر

بیشترین میزان جریانی را که می‌توان از یک مسیر افزایشی p عبور داد، ظرفیت آن مسیر گویند که به صورت زیر محاسبه می‌شود:

$$w_f(p) = \min\{w_f(u, v) : (u, v) \in p\}$$

گراف جریان اطلاعاتی

یال جریان

به یال e در شبکه $G = (V, E, c)$ که $f(e) > 0$ ، یال جریان می‌گوییم.

ظرفیت مسیر

بیشترین میزان جریانی را که می‌توان از یک مسیر افزایشی p عبور داد، ظرفیت آن مسیر گویند که به صورت زیر محاسبه می‌شود:

$$w_f(p) = \min\{w_f(u, v) : (u, v) \in p\}$$

کمینه برش

گراف جریان اطلاعاتی

یال جریان

به یال e در شبکه $G = (V, E, c)$ که $f(e) > 0$ ، یال جریان می‌گوییم.

ظرفیت مسیر

بیشترین میزان جریانی را که می‌توان از یک مسیر افزایشی p عبور داد، ظرفیت آن مسیر گویند که به صورت زیر محاسبه می‌شود:

$$w_f(p) = \min\{w_f(u, v) : (u, v) \in p\}$$

کمینه برش

یک $s - t$ برش کمینه، $s - t$ برشی است که کمترین ظرفیت ممکن را در میان تمام $s - t$ برش‌ها داشته باشد.

گراف جریان اطلاعاتی

یال جریان

به یال e در شبکه $G = (V, E, c)$ که $f(e) > 0$ ، یال جریان می‌گوییم.

ظرفیت مسیر

بیشترین میزان جریانی را که می‌توان از یک مسیر افزایشی p عبور داد، ظرفیت آن مسیر گویند که به صورت زیر محاسبه می‌شود:

$$w_f(p) = \min\{w_f(u, v) : (u, v) \in p\}$$

کمینه برش

یک $s - t$ برش کمینه، $s - t$ برشی است که کمترین ظرفیت ممکن را در میان تمام $s - t$ برش‌ها داشته باشد.

برش - کمینه و جریان - بیشینه

قضیه جریان - بیشینه و برش - کمینه

برش - کمینه و جریان - بیشینه

قضیه جریان - بیشینه و برش - کمینه

- چنانچه f یک جریان در شبکه $G = (V, E, c)$ با مبدأ s و مقصد t باشد، آنگاه سه گزاره زیر هم‌ارزند:

برش - کمینه و جریان - بیشینه

قضیه جریان - بیشینه و برش - کمینه

- چنانچه f یک جریان در شبکه $G = (V, E, c)$ با مبدأ s و مقصد t باشد، آنگاه سه گزاره زیر هم‌ارزند:
 $f - 1$ یک جریان بیشینه در G است.

برش - کمینه و جریان - بیشینه

قضیه جریان - بیشینه و برش - کمینه

- چنانچه f یک جریان در شبکه $G = (V, E, c)$ با مبدأ s و مقصد t باشد، آنگاه سه گزاره زیر هم‌ارزند:
 - 1- f یک جریان بیشینه در G است.
 - 2- هیچ مسیر افزایشی در شبکه باقیمانده G_f وجود ندارد.

برش - کمینه و جریان - بیشینه

قضیه جریان - بیشینه و برش - کمینه

- چنانچه f یک جریان در شبکه $G = (V, E, c)$ با مبدأ s و مقصد t باشد، آنگاه سه گزاره زیر هم‌ارزند:
 - 1- f یک جریان بیشینه در G است.
 - 2- هیچ مسیر افزایشی در شبکه باقیمانده G_f وجود ندارد.
 - 3- یک $s - t$ برش S در G وجود دارد که $|f| = c(S, \underline{S})$.

برش - کمینه و جریان - بیشینه

قضیه جریان - بیشینه و برش - کمینه

- چنانچه f یک جریان در شبکه $G = (V, E, c)$ با مبدأ s و مقصد t باشد، آنگاه سه گزاره زیر هم‌ارزند:
 - 1- f یک جریان بیشینه در G است.
 - 2- هیچ مسیر افزایشی در شبکه باقیمانده G_f وجود ندارد.
 - 3- یک $s - t$ برش S در G وجود دارد که $|f| = c(S, \underline{S})$.

برش - کمینه و جریان - بیشینه

قضیه جریان - بیشینه و برش - کمینه

- چنانچه f یک جریان در شبکه $G = (V, E, c)$ با مبدأ s و مقصد t باشد، آنگاه سه گزاره زیر هم‌ارزند:
 - 1- f یک جریان بیشینه در G است.
 - 2- هیچ مسیر افزایشی در شبکه باقیمانده G_f وجود ندارد.
 - 3- یک $s - t$ برش S در G وجود دارد که $|f| = c(S, \underline{S})$.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده



سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

یک سیستم ذخیره سازی توزیع شده شامل مجموعه ای از سخت افزارها و گره ها است که برای ذخیره ی داده ها در یک روش توزیع شده استفاده می شود: همان فایل را می توان چند بار ذخیره کرد. یک کپی در سخت افزار روی یک مجموعه از دو، سه یا چند سخت افزار، یا قسمتی از همان فایل را می توان در سرتاسر یک مجموعه از سخت افزار ذخیره کرد. دلیل اینکه ممکن است بخواهیم داده ها را به روش توزیع شده ذخیره کنیم عبارت است از:

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

یک سیستم ذخیره سازی توزیع شده شامل مجموعه ای از سخت افزارها و گره هاست که برای ذخیره ی داده ها در یک روش توزیع شده استفاده می شود: همان فایل را می توان چند بار ذخیره کرد. یک کپی در سخت افزار روی یک مجموعه از دو، سه یا چند سخت افزار، یا قسمتی از همان فایل را می توان در سرتاسر یک مجموعه از سخت افزار ذخیره کرد. دلیل اینکه ممکن است بخواهیم داده ها را به روش توزیع شده ذخیره کنیم عبارت است از:

سهولت مقیاس: اگر حافظه سیستم پر شود، یک دیسک بیشتر به سیستم ذخیره سازی اضافه می کند.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

یک سیستم ذخیره سازی توزیع شده شامل مجموعه ای از سخت افزارها و گره هاست که برای ذخیره ی داده ها در یک روش توزیع شده استفاده می شود: همان فایل را می توان چند بار ذخیره کرد. یک کپی در سخت افزار روی یک مجموعه از دو، سه یا چند سخت افزار، یا قسمتی از همان فایل را می توان در سرتاسر یک مجموعه از سخت افزار ذخیره کرد. دلیل اینکه ممکن است بخواهیم داده ها را به روش توزیع شده ذخیره کنیم عبارت است از:

سهولت مقیاس: اگر حافظه سیستم پر شود، یک دیسک بیشتر به سیستم ذخیره سازی اضافه می کند.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده



سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

قابلیت اطمینان: اگر فقط یک دیسک وجود داشته باشد، و آن هم از بین برود، داده های ذخیره شده از بین می روند. در صورتی که اگر چندین دیسک حاوی داده های مشابه موجود باشد، داده ها شانس باقی ماندن دارند ک بستگی به شکست ها و چگونگی ذخیره داده ها دارد. اگر چندین دیسک موجود باشد، اما هر فایل فقط در یک دیسک ذخیره شود، در این صورت قابل اطمینان نیست. زیرا در صورت از بین رفتن دیسک خاص، داده ها از بین می روند. برای همین فراوانی نیاز است.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

قابلیت اطمینان: اگر فقط یک دیسک وجود داشته باشد، و آن هم از بین برود، داده‌های ذخیره شده از بین می‌روند. در صورتی که اگر چندین دیسک حاوی داده‌های مشابه موجود باشد، داده‌ها شانس باقی ماندن دارند ک بستگی به شکست‌ها و چگونگی ذخیره داده‌ها دارد. اگر چندین دیسک موجود باشد، اما هر فایل فقط در یک دیسک ذخیره شود، در اینصورت قابل اطمینان نیست. زیرا در صورت از بین رفتن دیسک خاص، داده‌ها از بین می‌روند. برای همین فراوانی نیاز است. ساده‌ترین نوع فراوانی، تکثیر است. دو یا سه کپی از فایل گرفته می‌شود و همه آن‌ها در دیسک‌های مختلف ذخیره می‌شوند.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

قابلیت اطمینان: اگر فقط یک دیسک وجود داشته باشد، و آن هم از بین برود، داده‌های ذخیره شده از بین می‌روند. در صورتی که اگر چندین دیسک حاوی داده‌های مشابه موجود باشد، داده‌ها شانس باقی ماندن دارند ک بستگی به شکست‌ها و چگونگی ذخیره داده‌ها دارد. اگر چندین دیسک موجود باشد، اما هر فایل فقط در یک دیسک ذخیره شود، در اینصورت قابل اطمینان نیست. زیرا در صورت از بین رفتن دیسک خاص، داده‌ها از بین می‌روند. برای همین فراوانی نیاز است. ساده‌ترین نوع فراوانی، تکثیر است. دو یا سه کپی از فایل گرفته می‌شود و همه آن‌ها در دیسک‌های مختلف ذخیره می‌شوند.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کدگذاری

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کدگذاری

فرآیند اضافه کردن فراوانی به سیگنال ارسال شده، کدگذاری نامیده می شود.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کدگذاری

فرآیند اضافه کردن فراوانی به سیگنال ارسال شده، کدگذاری نامیده می شود.

قضیه کران سینگلتون

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کدگذاری

فرآیند اضافه کردن فراوانی به سیگنال ارسال شده، کدگذاری نامیده می شود.

قضیه کران سینگلتون

برای هر عدد صحیح d و $q > 1$ و عدد صحیح مثبت n بطوری که $1 \leq d \leq n$ داریم:

$$A_q(n, d) \leq q^{n-d+1}$$

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کدگذاری

فرآیند اضافه کردن فراوانی به سیگنال ارسال شده، کدگذاری نامیده می شود.

قضیه کران سینگلتون

برای هر عدد صحیح d و $q > 1$ و عدد صحیح مثبت n بطوری که $1 \leq d \leq n$ داریم:

$$A_q(n, d) \leq q^{n-d+1}$$

به ویژه اگر q توانی اول باشد، پارامترهای $[n, k, d]$ از هر کد خطی روی F_q در رابطه روبرو صدق می نماید: $k + d \leq n + 1$.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کدگذاری

فرآیند اضافه کردن فراوانی به سیگنال ارسال شده، کدگذاری نامیده می شود.

قضیه کران سینگلتون

برای هر عدد صحیح d و $q > 1$ و عدد صحیح مثبت n بطوری که $1 \leq d \leq n$ داریم:

$$A_q(n, d) \leq q^{n-d+1}$$

به ویژه اگر q توانی اول باشد، پارامترهای $[n, k, d]$ از هر کد خطی روی F_q در رابطه روبرو صدق می نماید: $k + d \leq n + 1$.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

فرض کنیم F_q یک میدان متناهی باشد و a_1, \dots, a_n عناصر مجزا از F_q باشند، n و k را طوری انتخاب می کنیم که $k \leq n \leq q$. در اینصورت برای هر $m = (m_0, \dots, m_{k-1}) \in F_q^k$ قرار دهید:

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

فرض کنیم F_q یک میدان متناهی باشد و a_1, \dots, a_n عناصر مجزا از F_q باشند، n و k را طوری انتخاب می کنیم که $k \leq n \leq q$. در اینصورت برای هر $m = (m_0, \dots, m_{k-1}) \in F_q^k$ قرار دهید:

$$f_m(x) = \sum_{i=0}^{k-1} m_i x^i = m_0 + m_1 x + \dots + m_{k-1} x^{k-1}$$

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

فرض کنیم F_q یک میدان متناهی باشد و a_1, \dots, a_n عناصر مجزا از F_q باشند، n و k را طوری انتخاب می کنیم که $k \leq n \leq q$. در اینصورت برای هر $m = (m_0, \dots, m_{k-1}) \in F_q^k$ قرار دهید:

$$f_m(x) = \sum_{i=0}^{k-1} m_i x^i = m_0 + m_1 x + \dots + m_{k-1} x^{k-1}$$

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

واضح است که $f_m(x) \in F_q[x]$ یک چندجمله‌ای از درجه $k - 1$ است. کدگذاری m یک ارزیابی از $f_m(x)$ در همه a_i ‌هاست.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

واضح است که $f_m(x) \in F_q[x]$ یک چندجمله‌ای از درجه $k - 1$ است. کدگذاری m یک ارزیابی از $f_m(x)$ در همه a_i ‌هاست.

$$RS : F_q^k \longrightarrow F_q^n$$

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

واضح است که $f_m(x) \in F_q[x]$ یک چندجمله‌ای از درجه $k - 1$ است. کدگذاری m یک ارزیابی از $f_m(x)$ در همه a_i ‌هاست.

$$RS : F_q^k \longrightarrow F_q^n$$

$$RS(m) = (f_m(a_1), \dots, f_m(a_n))$$

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

واضح است که $f_m(x) \in F_q[x]$ یک چندجمله‌ای از درجه $k - 1$ است. کدگذاری m یک ارزیابی از $f_m(x)$ در همه a_i هاست.

$$RS : F_q^k \longrightarrow F_q^n$$

$$RS(m) = (f_m(a_1), \dots, f_m(a_n))$$

این تصویر را کد رید-سالمون می‌نامیم.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

کد رید-سالمون

واضح است که $f_m(x) \in F_q[x]$ یک چندجمله‌ای از درجه $k - 1$ است. کدگذاری m یک ارزیابی از $f_m(x)$ در همه a_i هاست.

$$RS : F_q^k \longrightarrow F_q^n$$

$$RS(m) = (f_m(a_1), \dots, f_m(a_n))$$

این تصویر را کد رید-سالمون می‌نامیم.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده



سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

در چارچوب سیستم های ذخیره سازی توزیع شده، زمانی که بین قابلیت اطمینان و هزینه ذخیره سازی توازن برقرار باشد، مطلوب می باشند. یک سوال پیش می آید که چرا پس به جای تحقیق در مورد استراتژی های جدید کدگذاری سیستم های ذخیره سازی توزیع شده، فقط کدهای رید-سالمون را اتخاذ نکرده اند؟

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

در چارچوب سیستم های ذخیره سازی توزیع شده، زمانی که بین قابلیت اطمینان و هزینه ذخیره سازی توازن برقرار باشد، مطلوب می باشند. یک سوال پیش می آید که چرا پس به جای تحقیق در مورد استراتژی های جدید کدگذاری سیستم های ذخیره سازی توزیع شده، فقط کدهای رید-سالمون را اتخاذ نکرده اند؟

دلیل آن این است که برنامه نویسی برای سیستم های ذخیره سازی توزیع شده شامل یک بُعد دیگری بالاتر از قابلیت اطمینان و هزینه ذخیره سازی، یعنی تعمیر و نگهداری یا تعمیر مجدد است.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

در چارچوب سیستم های ذخیره سازی توزیع شده، زمانی که بین قابلیت اطمینان و هزینه ذخیره سازی توازن برقرار باشد، مطلوب می باشند. یک سوال پیش می آید که چرا پس به جای تحقیق در مورد استراتژی های جدید کدگذاری سیستم های ذخیره سازی توزیع شده، فقط کدهای رید-سالمون را اتخاذ نکرده اند؟

دلیل آن این است که برنامه نویسی برای سیستم های ذخیره سازی توزیع شده شامل یک بُعد دیگری بالاتر از قابلیت اطمینان و هزینه ذخیره سازی، یعنی تعمیر و نگهداری یا تعمیر مجدد است.

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده



سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

بنابراین تحقیق در زمینه‌ی برنامه‌نویسی برای سیستم‌های ذخیره‌سازی توزیع شده بر روی دریافت کدها با ویژگی زیر تمرکز دارد:

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

بنابراین تحقیق در زمینه‌ی برنامه‌نویسی برای سیستم‌های ذخیره‌سازی توزیع شده بر روی دریافت کدها با ویژگی زیر تمرکز دارد:

- 1- قابلیت اطمینان بالاتر (تحمل خطا)

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

بنابراین تحقیق در زمینه‌ی برنامه‌نویسی برای سیستم‌های ذخیره‌سازی توزیع شده بر روی دریافت کدها با ویژگی زیر تمرکز دارد:

1- قابلیت اطمینان بالاتر (تحمل خطا)

2- هزینه ذخیره‌سازی کمتر (اگر هزینه ذخیره‌سازی مشکل نباشد، تکثیر یک راه‌حل عملی عالی است)

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

بنابراین تحقیق در زمینه‌ی برنامه‌نویسی برای سیستم‌های ذخیره‌سازی توزیع شده بر روی دریافت کدها با ویژگی زیر تمرکز دارد:

1- قابلیت اطمینان بالاتر (تحمل خطا)

2- هزینه ذخیره‌سازی کمتر (اگر هزینه ذخیره‌سازی مشکل نباشد، تکثیر یک راه‌حل عملی عالی است)

3- مکانیسم‌های تعمیر و نگهداری خوب

سیستم های ذخیره سازی توزیع شده و کدهای پاک کننده

بنابراین تحقیق در زمینه‌ی برنامه‌نویسی برای سیستم‌های ذخیره‌سازی توزیع شده بر روی دریافت کدها با ویژگی زیر تمرکز دارد:

1- قابلیت اطمینان بالاتر (تحمل خطا)

2- هزینه ذخیره‌سازی کمتر (اگر هزینه ذخیره‌سازی مشکل نباشد، تکثیر یک راه‌حل عملی عالی است)

3- مکانیسم‌های تعمیر و نگهداری خوب



کدهای بازسازی، یک کلاس از کدهایی است که هدف آن کاهش میزان دانلود در حین تعمیر و در عین حال حفظ کارایی کدهای سنتی MDS است.

کدهای بازسازی، یک کلاس از کدهایی است که هدف آن کاهش میزان دانلود در حین تعمیر و در عین حال حفظ کارایی کدهای سنتی MDS است.

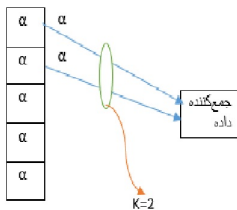
تحت استفاده از کد بازسازی، یک گره جایگزین به هر $d \geq k$ گره موجود متصل می‌شود و β نماد را از آن‌ها بارگیری می‌کند. این تنظیم در شکل زیر نشان داده شده است. در کد بازسازی کل $d\beta$ داده که برای تعمیر بارگیری می‌شود، بسیار کوچکتر از اندازه کل پیام B است. در شکل نشان داده شده است که پارامترهای مرتبط با یک کد بازسازی باید لزوماً در رابطه زیر صدق کند :

کدهای بازسازی، یک کلاس از کدهایی است که هدف آن کاهش میزان دانلود در حین تعمیر و در عین حال حفظ کارایی کدهای سنتی MDS است. تحت استفاده از کد بازسازی، یک گره جایگزین به هر $d \geq k$ گره موجود متصل می شود و β نماد را از آن ها بارگیری می کند. این تنظیم در شکل زیر نشان داده شده است. در کد بازسازی کل $d\beta$ داده که برای تعمیر بارگیری می شود، بسیار کوچکتر از اندازه کل پیام B است. در شکل نشان داده شده است که پارامترهای مرتبط با یک کد بازسازی باید لزوماً در رابطه زیر صدق کند :

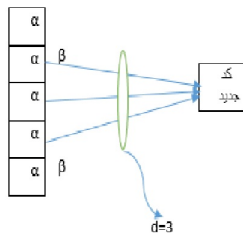
$$B \leq \sum_{i=0}^{k-1} \min\{\alpha, (d - i)\beta\}$$

کدهای بازسازی، یک کلاس از کدهایی است که هدف آن کاهش میزان دانلود در حین تعمیر و در عین حال حفظ کارایی کدهای سنتی MDS است. تحت استفاده از کد بازسازی، یک گره جایگزین به هر $d \geq k$ گره موجود متصل می شود و β نماد را از آن ها بارگیری می کند. این تنظیم در شکل زیر نشان داده شده است. در کد بازسازی کل $d\beta$ داده که برای تعمیر بارگیری می شود، بسیار کوچکتر از اندازه کل پیام B است. در شکل نشان داده شده است که پارامترهای مرتبط با یک کد بازسازی باید لزوماً در رابطه زیر صدق کند :

$$B \leq \sum_{i=0}^{k-1} \min\{\alpha, (d - i)\beta\}$$



(a)



(b)

توضیح شکل

توضیح شکل

شکل فوق نمونه‌ای از پارامترهای سیستم تحت یک کد بازسازی شده (در صورت عدم وجود خطا) و این سیستم شامل $n = 5$ گره ذخیره‌سازی است.

توضیح شکل

شکل فوق نمونه‌ای از پارامترهای سیستم تحت یک کد بازسازی شده (در صورت عدم وجود خطا) و این سیستم شامل $n = 5$ گره دخیره‌سازی است. در شکل (a) بازسازی از هر گره $k = 2$ انجام می‌شود و در شکل (b) تعمیر از هر گره $d = 3$ انجام می‌شود

توضیح شکل

شکل فوق نمونه‌ای از پارامترهای سیستم تحت یک کد بازسازی شده (در صورت عدم وجود خطا) و این سیستم شامل $n = 5$ گره دخیره‌سازی است. در شکل (a) بازسازی از هر گره $k = 2$ انجام می‌شود و در شکل (b) تعمیر از هر گره $d = 3$ انجام می‌شود



اگر معادله قبل برای یک کدبازسازی برابر باشد یعنی اگر

اگر معادله قبل برای یک کد بازسازی برابر باشد یعنی اگر

$$B = \sum_{i=0}^{k-1} \min\{\alpha, (d - i)\beta\}$$

اگر معادله قبل برای یک کد بازسازی برابر باشد یعنی اگر

$$B = \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}$$

آنگاه آن کد بهینه است. چون هم ظرفیت ذخیره‌سازی و پهنای باند با هزینه‌ای همراه هستند، طبیعی است که اگر α و β به حداقل برسد، B نیز بهینه می‌شود.

اگر معادله قبل برای یک کد بازسازی برابر باشد یعنی اگر

$$B = \sum_{i=0}^{k-1} \min\{\alpha, (d - i)\beta\}$$

آنگاه آن کد بهینه است. چون هم ظرفیت ذخیره‌سازی و پهنای باند با هزینه‌ای همراه هستند، طبیعی است که اگر α و β به حداقل برسد، B نیز بهینه می‌شود. هنگامی که یک گره در سیستم ذخیره‌سازی توزیع شده آسیب می‌بیند، سازوکار تعمیر و نگهداری تلاش می‌کند داده‌های از دست رفته را با استفاده از داده‌های موجود در سیستم جایگزین کند. این شامل دانلود داده‌ها از گره‌های باقیمانده است. که هزینه‌های ارتباطی را شامل می‌شود و به عنوان پهنای باند تعمیر شناخته می‌شود.

اگر معادله قبل برای یک کد بازسازی برابر باشد یعنی اگر

$$B = \sum_{i=0}^{k-1} \min\{\alpha, (d - i)\beta\}$$

آنگاه آن کد بهینه است. چون هم ظرفیت ذخیره سازی و پهنای باند با هزینه ای همراه هستند، طبیعی است که اگر α و β به حداقل برسد، B نیز بهینه می شود. هنگامی که یک گره در سیستم ذخیره سازی توزیع شده آسیب می بیند، سازوکار تعمیر و نگهداری تلاش می کند داده های از دست رفته را با استفاده از داده های موجود در سیستم جایگزین کند. این شامل دانلود داده ها از گره های باقیمانده است. که هزینه های ارتباطی را شامل می شود و به عنوان پهنای باند تعمیر شناخته می شود.

بازسازی کدهای

کدهای بازسازی

بازسازی کدهای

دیماکیس و همکاران: تکنیک جریان-بیشینه و برش-کمینه از کدگذاری شبکه تا تنظیم ذخیره‌سازی توزیع‌شده برای بهینه‌سازی پهنای باند تعمیر ارائه کردند. نتیجه تجزیه و تحلیل آن‌ها توازن میان ذخیره‌سازی و پهنای باند تعمیر است. کدهایی که به این کران می‌رسند، کدهای بازسازی نامیده می‌شود.

کدهای بازسازی

بازسازی کدهای

دیماکیس و همکاران: تکنیک جریان-بیشینه و برش-کمینه از کدگذاری شبکه تا تنظیم ذخیره‌سازی توزیع‌شده برای بهینه‌سازی پهنای باند تعمیر ارائه کردند. نتیجه تجزیه و تحلیل آن‌ها توازن میان ذخیره‌سازی و پهنای باند تعمیر است. کدهایی که به این کران می‌رسند، کدهای بازسازی نامیده می‌شود.

ایده اصلی کدهای بازسازی این است که آن‌ها می‌توانند در یک ویژگی مشابه MDS صدق کنند. (فایل را می‌توان توسط اتصال هر انتخابی از k گره‌ها بازیابی کرد) و در فرآیند تعمیر، بسیاری از گره‌ها می‌توانند اتصال داده شوند (معمولاً همه گره‌ها، به جز گره‌های از بین رفته) و نمادهای کمتری دانلود شود، که به نوبه‌ی خود هزینه‌ی تعمیر یک خرابی را کاهش می‌دهد.

کدهای بازسازی

بازسازی کدهای

دیماکیس و همکاران: تکنیک جریان-بیشینه و برش-کمینه از کدگذاری شبکه تا تنظیم ذخیره‌سازی توزیع‌شده برای بهینه‌سازی پهنای باند تعمیر ارائه کردند. نتیجه تجزیه و تحلیل آن‌ها توازن میان ذخیره‌سازی و پهنای باند تعمیر است. کدهایی که به این کران می‌رسند، کدهای بازسازی نامیده می‌شود.

ایده اصلی کدهای بازسازی این است که آن‌ها می‌توانند در یک ویژگی مشابه MDS صدق کنند. (فایل را می‌توان توسط اتصال هر انتخابی از k گره‌ها بازیابی کرد) و در فرآیند تعمیر، بسیاری از گره‌ها می‌توانند اتصال داده شوند (معمولاً همه گره‌ها، به جز گره‌های از بین رفته) و نمادهای کمتری دانلود شود، که به نوبه‌ی خود هزینه‌ی تعمیر یک خرابی را کاهش می‌دهد.

پارمترهای کدهای بازسازی و کران برش-کمینه



پارمترهای کدهای بازسازی و کران برش-کمینه

یک شبکه با N گره و یک فایل f به طول M را در نظر بگیرید که در n تا از این N گره‌ها ذخیره می‌شود. هر گره فرض می‌شود که ظرفیت ذخیره‌سازی مثل α داشته باشد، یعنی α نماد در هر گره ذخیره می‌شود.

پارمترهای کدهای بازسازی و کران برش-کمینه

یک شبکه با N گره و یک فایل f به طول M را در نظر بگیرید که در n تا از این N گره‌ها ذخیره می‌شود. هر گره فرض می‌شود که ظرفیت ذخیره‌سازی مثل α داشته باشد، یعنی α نماد در هر گره ذخیره می‌شود.

هنگامی که یک خرابی اتفاق می‌افتد یک گره باقیمانده (به نام تازه‌وارد) در میان این N گره‌ها که هنوز هنوز اطلاعات را از f ذخیره نکرده، به فرآیند تعمیر می‌پیوندد. تازه‌وارد به d گره باقیمانده متصل می‌شود و از هر یک از آن‌ها β مقدار دریافت می‌کند، تا اطلاعاتی که از طریق گره‌های خراب شده از بین رفته است را تعمیر کند، همانطور که در شکل زیر نشان داده شده است.

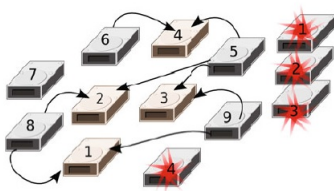
پارمترهای کدهای بازسازی و کران برش-کمینه

یک شبکه با N گره و یک فایل f به طول M را در نظر بگیرید که در n تا از این N گره‌ها ذخیره می‌شود. هر گره فرض می‌شود که ظرفیت ذخیره‌سازی مثل α داشته باشد، یعنی α نماد در هر گره ذخیره می‌شود.

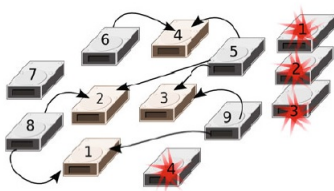
هنگامی که یک خرابی اتفاق می‌افتد یک گره باقیمانده (به نام تازه‌وارد) در میان این N گره‌ها که هنوز هنوز اطلاعات را از f ذخیره نکرده، به فرآیند تعمیر می‌پیوندد. تازه‌وارد به d گره باقیمانده متصل می‌شود و از هر یک از آن‌ها β مقدار دریافت می‌کند، تا اطلاعاتی که از طریق گره‌های خراب شده از بین رفته است را تعمیر کند، همانطور که در شکل زیر نشان داده شده است.

پارمترهای کدهای بازسازی و کران برش-کمینه

پارمترهای کدهای بازسازی و کران برش-کمینه



پارمترهای کدهای بازسازی و کران برش-کمینه



پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

در شکل فوق چهار گره که از 1 تا 4 شماره‌گذاری شده‌اند، از بین رفته‌اند. چهار گره دیگر در مرکز با همان برچسب‌ها اطلاعات مربوط به گره‌های از دست رفته را با اتصال با گره‌های باقیمانده‌ی 5 و 6 و 8 و 9 تعمیر می‌کنند.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

در شکل فوق چهار گره که از 1 تا 4 شماره گذاری شده اند، از بین رفته اند. چهار گره دیگر در مرکز با همان برچسبها اطلاعات مربوط به گره های از دست رفته را با اتصال با گره های باقیمانده ی 5 و 6 و 8 و 9 تعمیر می کنند. ما داریم $d = 2$ یعنی گره 3 داده های خود را از گره های 5 و 9 بارگیری می کند. درمورد کد بازسازی اصلی، چهار شکست به ترتیب تعمیر می شوند و داده های 1 و 2 و 3 و 4 که درگیر تعمیر هستند نیز داده ها را مبادله می کنند

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

در شکل فوق چهار گره که از 1 تا 4 شماره گذاری شده اند، از بین رفته اند. چهار گره دیگر در مرکز با همان برچسبها اطلاعات مربوط به گره های از دست رفته را با اتصال با گره های باقیمانده ی 5 و 6 و 8 و 9 تعمیر می کنند. ما داریم $d = 2$ یعنی گره 3 داده های خود را از گره های 5 و 9 با رگیری می کند. درمورد کد بازسازی اصلی، چهار شکست به ترتیب تعمیر می شوند و داده های 1 و 2 و 3 و 4 که درگیر تعمیر هستند نیز داده ها را مبادله می کنند

پارمترهای کدهای بازسازی و کران برش-کمینه

مثال

پارمترهای کدهای بازسازی و کران برش-کمینه

مثال

- چهار پاکت داده A_1 و A_2 و B_1 و B_2 روی چهار گره گره ذخیره‌سازی توزیع می‌شوند، که هر کدام از این گره‌ها دو پاکت را ذخیره می‌کنند.

پارمترهای کدهای بازسازی و کران برش-کمینه

مثال

- چهار پاکت داده A_1 و A_2 و B_1 و B_2 روی چهار گره گره ذخیره‌سازی توزیع می‌شوند، که هر کدام از این گره‌ها دو پاکت را ذخیره می‌کنند. اولین گره A_1 و A_2 و دومین گره هم B_1 و B_2 ذخیره می‌کند. گره سوم و چهارم گره‌های برابری هستند. گره سوم شامل دو پاکت $A_1 + B_1$ و $2A_2 + B_2$ است و گره چهارم شامل $2A_1 + B_1$ و $A_2 + B_2$ است.

پارمترهای کدهای بازسازی و کران برش-کمینه

مثال

- چهار پاکت داده A_1 و A_2 و B_1 و B_2 روی چهار گره گره ذخیره‌سازی توزیع می‌شوند، که هر کدام از این گره‌ها دو پاکت را ذخیره می‌کنند. اولین گره A_1 و A_2 و دومین گره هم B_1 و B_2 ذخیره می‌کند. گره سوم و چهارم گره‌های برابری هستند. گره سوم شامل دو پاکت $A_1 + B_1$ و $2A_2 + B_2$ است و گره چهارم شامل $2A_1 + B_1$ و $A_2 + B_2$ است. در اینجا یک پاکت به عنوان یک عنصر در میدان متناهی در نظر گرفته می‌شود که عمل‌های این میدان متناهی جمع و ضرب هستند.

پارمترهای کدهای بازسازی و کران برش-کمینه

مثال

- چهار پاکت داده A_1 و A_2 و B_1 و B_2 روی چهار گره گره ذخیره‌سازی توزیع می‌شوند، که هر کدام از این گره‌ها دو پاکت را ذخیره می‌کنند. اولین گره A_1 و A_2 و دومین گره هم B_1 و B_2 ذخیره می‌کند. گره سوم و چهارم گره‌های برابری هستند. گره سوم شامل دو پاکت $A_1 + B_1$ و $2A_2 + B_2$ است و گره چهارم شامل $2A_1 + B_1$ و $A_2 + B_2$ است. در اینجا یک پاکت به عنوان یک عنصر در میدان متناهی در نظر گرفته می‌شود که عمل‌های این میدان متناهی جمع و ضرب هستند.

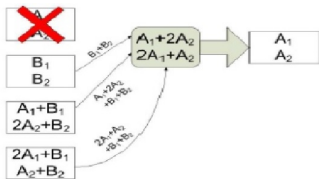
پارمترهای کدهای بازسازی و کران برش-کمینه

مثال

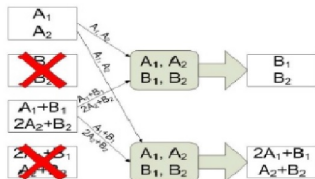
- چهار پاکت داده A_1 و A_2 و B_1 و B_2 روی چهار گره گره ذخیره‌سازی توزیع می‌شوند، که هر کدام از این گره‌ها دو پاکت را ذخیره می‌کنند. اولین گره A_1 و A_2 و دومین گره هم B_1 و B_2 ذخیره می‌کند. گره سوم و چهارم گره‌های برابری هستند. گره سوم شامل دو پاکت $A_1 + B_1$ و $2A_2 + B_2$ است و گره چهارم شامل $2A_1 + B_1$ و $A_2 + B_2$ است. در اینجا یک پاکت به عنوان یک عنصر در میدان متناهی در نظر گرفته می‌شود که عمل‌های این میدان متناهی جمع و ضرب هستند.

پارمترهای کدهای بازسازی و کران برش-کمینه

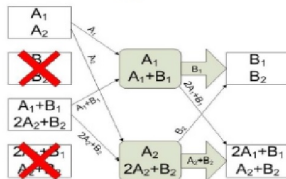
پارمترهای کدهای بازسازی و کران برش-کمینه



(a)

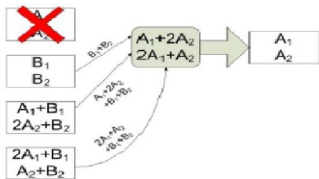


(b)

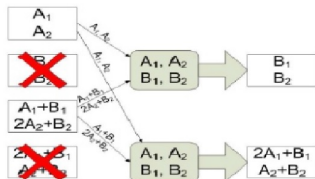


(c)

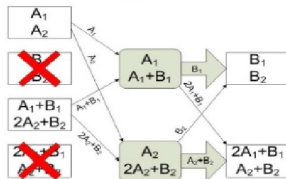
پارمترهای کدهای بازسازی و کران برش-کمینه



(a)



(b)



(c)

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

در شکل فوق (a) - تعمیر فقط یک گره از بین رفته با حداقل پهنای باند، شکل (b) - بازسازی فردی از چندین خرابی و شکل (c) - بازسازی همکار از چندین خرابی را نشان می دهد در ادامه به بررسی بیشتر این شکل ها خواهیم پرداخت.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

در شکل فوق (a) - تعمیر فقط یک گره از بین رفته با حداقل پهنای باند، شکل (b) - بازسازی فردی از چندین خرابی و شکل (c) - بازسازی همکار از چندین خرابی را نشان می دهد در ادامه به بررسی بیشتر این شکل ها خواهیم پرداخت.

توضیح شکل (a)

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

در شکل فوق (a) - تعمیر فقط یک گره از بین رفته با حداقل پهنای باند، شکل (b) - بازسازی فردی از چندین خرابی و شکل (c) - بازسازی همکار از چندین خرابی را نشان می دهد در ادامه به بررسی بیشتر این شکل ها خواهیم پرداخت.

توضیح شکل (a)

در شکل (a) - فرض کنید گره اول از بین برود، برای ترمیم گره اول می توانیم چهار پاکت را از هر دو گره دیگر بارگیری کنیم که از این طریق می توانیم دو پاکت مورد نیاز A_1 و A_2 را بازیابی کنیم.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل

در شکل فوق (a) - تعمیر فقط یک گره از بین رفته با حداقل پهنای باند، شکل (b) - بازسازی فردی از چندین خرابی و شکل (c) - بازسازی همکار از چندین خرابی را نشان می دهد در ادامه به بررسی بیشتر این شکل ها خواهیم پرداخت.

توضیح شکل (a)

در شکل (a) - فرض کنید گره اول از بین برود، برای ترمیم گره اول می توانیم چهار پاکت را از هر دو گره دیگر بارگیری کنیم که از این طریق می توانیم دو پاکت مورد نیاز A_1 و A_2 را بازیابی کنیم.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (a)



پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (a)

به عنوان مثال، اگر پاکت‌ها را از گره دوم و سوم بارگیری کنیم، B_1 و B_1 و A_2 و $A_1 + B_1$ و $2A_2 + B_2$ داریم. سپس می‌توانیم با کم کردن B_1 از $A_1 + B_1$ و A_2 را با محاسبه $\frac{((2A_2 + B_2) - B_2)}{2}$ بازیابی کنیم.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (a)

به عنوان مثال، اگر پاکت‌ها را از گره دوم و سوم بارگیری کنیم، B_1 و B_1 و A_2 و $A_1 + B_1$ داریم. سپس می‌توانیم با کم کردن B_1 از $A_1 + B_1$ و A_2 را با محاسبه $\frac{((2A_2 + B_2) - B_2)}{2}$ بازیابی کنیم.

در ادامه می‌توانیم با سه اتصال به سه گره باقیمانده و بارگیری یک پاکت از هر یک از آنها، پهنای باند را از چهار پاکت به سه پاکت کاهش می‌دهیم هر یک از سه گره‌ی باقیمانده به سادگی دو پاکت را جمع می‌کند و نتیجه را برای تازه‌وارد ارسال می‌کند، که سپس می‌توان $B_1 + B_2$ را کم کرد و $A_1 + 2A_2$ و $2A_1 + A_2$ را به دست آورد که با حل آن می‌توان A_1 و A_2 را بازیابی کنیم.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (b)



پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (b)

هنگامی که دو گره ذخیره‌سازی به طور همزمان خراب شوند، روش محاسباتی که در پاراگراف قبلی ذکر شد دیگر کارساز نیست. فرض کنید گره‌های ذخیره‌سازی دوم و چهارم به طور همزمان خراب شوند. برای ترمیم هر دو آن‌ها به طور جداگانه، هر یک از اعضای تازه‌وارد می‌توانند چهار پکت را از گره‌های باقیمانده بارگیری کنند، پکت‌های A_1 و A_2 و B_1 و B_2 را بازسازی کرده و پکت‌های موردنظر را دوباره کدگذاری می‌کنند.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (b)

هنگامی که دو گره ذخیره‌سازی به طور همزمان خراب شوند، روش محاسباتی که در پاراگراف قبلی ذکر شد دیگر کارساز نیست. فرض کنید گره‌های ذخیره‌سازی دوم و چهارم به طور همزمان خراب شوند. برای ترمیم هر دو آن‌ها به طور جداگانه، هر یک از اعضای تازه‌وارد می‌توانند چهار پاکت را از گره‌های باقیمانده بارگیری کنند، پاکت‌های A_1 و A_2 و B_1 و B_2 را بازسازی کرده و پاکت‌های موردنظر را دوباره کدگذاری می‌کنند.

اولین تازه‌وارد به ترتیب A_1 و $A_1 + B_1$ را از گره‌ی اول و سوم دریافت می‌کند در حالی که تازه‌وارد دوم به ترتیب A_2 و $2A_2 + B_2$ را از گره اول و سوم دریافت می‌کند.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (b)

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (b)

تازه‌وارد اول B_1 و $2A_1 + B_1$ را با در نظر گرفتن جمع و تفریق دو ورودی محاسبه می‌کند. پکت B_1 در اولین تازه‌وارد ذخیره می‌شود و $2A_1 + B_1$ به دومین تازه‌وارد ارسال می‌شود. به طور مشابه، تازه‌وارد دوم B_2 و $A_2 + B_2$ را محاسبه می‌کند، $A_2 + B_2$ را در حافظه ذخیره می‌کند و B_2 را به اولین تازه‌وارد ارسال می‌کند. در این فرآیند بازسازی همکار، فقط شش انتقال پکت لازم است.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (c)

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (c)

اگر این دو تازه‌وارد بتوانند داده‌ها را طی فرآیند بازیابی مبادله کنند، در کل می‌توان پهنای باند تعمیر را از هشت پاکت به شش پاکت کاهش داد. دو تا تازه‌وارد برای اولین بار توافق می‌کنند که یکی از آن‌ها پاکت‌هایی با اندیس 1 را بارگیری کند و دیگری پاکت‌های با اندیس 2 را بارگیری کند. اولین تازه‌وارد به ترتیب A_1 و $A_1 + B_1$ را از گره اول و سوم دریافت می‌کند در حالی که تازه‌وارد دوم به ترتیب A_2 و $2A_2 + B_2$ را از گره اول و سوم دریافت می‌کند.

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (c)

پارمترهای کدهای بازسازی و کران برش-کمینه

توضیح شکل (c)

تازه‌وارد اول B_1 و $2A_1 + B_1$ را با در نظر گرفتن جمع و تفریق دو ورودی محاسبه می‌کند. پاکت B_1 در اولین تازه‌وارد ذخیره می‌شود و $2A_1 + B_1$ به دومین تازه‌وارد ارسال می‌شود. به طور مشابه، تازه‌وارد دوم B_2 و $A_2 + B_2$ را محاسبه می‌کند، $A_2 + B_2$ را در حافظه ذخیره می‌کند و B_2 را به اولین تازه‌وارد ارسال می‌کند. در این فرآیند بازسازی همکار، فقط شش انتقال پاکت لازم است.

امنیت سیستم های ذخیره سازی توزیع شده



امنیت سیستم های ذخیره سازی توزیع شده

موقعی که یک سیستم ذخیره سازی توزیع شده با استفاده از گره هایی شکل می گیرد ممکن است ایمن نباشد، از این رو ممکن است یک مزاحم می تواند در گره ها استراق سمع کند و داده های آنها را تغییر دهد مثل ویروس ها و ...

امنیت سیستم های ذخیره سازی توزیع شده

موقعی که یک سیستم ذخیره سازی توزیع شده با استفاده از گره‌هایی شکل می‌گیرد ممکن است ایمن نباشد، از این رو ممکن است یک مزاحم می‌تواند در گره‌ها استراق سمع کند و داده‌های آن‌ها را تغییر دهد مثل ویروس‌ها و ...

در این بخش ما به مشکل امنیت سیستم‌های ذخیره سازی توزیع شده با گره‌هایی که به طور مداوم در حال ترک و پیوستن به سیستم هستند، در برابر چنین مزاحمانی پرداخته ایم. رفتار پویا سیستم می‌تواند با قدرتمندتر شدن مزاحم (متجاوز)، داده‌ها را به خطر اندازد.

امنیت سیستم های ذخیره سازی توزیع شده

موقعی که یک سیستم ذخیره سازی توزیع شده با استفاده از گره‌هایی شکل می‌گیرد ممکن است ایمن نباشد، از این رو ممکن است یک مزاحم می‌تواند در گره‌ها استراق سمع کند و داده‌های آن‌ها را تغییر دهد مثل ویروس‌ها و ...

در این بخش ما به مشکل امنیت سیستم‌های ذخیره سازی توزیع شده با گره‌هایی که به طور مداوم در حال ترک و پیوستن به سیستم هستند، در برابر چنین مزاحمانی پرداخته ایم. رفتار پویا سیستم می‌تواند با قدرتمندتر شدن مزاحم (متجاوز)، داده‌ها را به خطر اندازد.

امنیت سیستم های ذخیره سازی توزیع شده



امنیت سیستم های ذخیره سازی توزیع شده

به عنوان مثال، در حین استراق سمع بر روی یک گره جدید در طی فرآیند تعمیر، متجاوز می‌تواند نه تنها محتوای ذخیره شده‌ی خود بلکه تمام داده‌های بارگیری شده آن را نیز مشاهده کند. علاوه بر این، به یک رقیب این امکان را می‌دهد تا با ارسال پیام نادرست هنگام تماس برای تعمیر، خطاهایی را در گره‌های خارج از کنترل خود وارد کند. در تجزیه و تحلیل، به سه نوع از مزاحمان متمرکز می‌شویم:

امنیت سیستم های ذخیره سازی توزیع شده

به عنوان مثال، در حین استراق سمع بر روی یک گره جدید در طی فرآیند تعمیر، متجاوز می‌تواند نه تنها محتوای ذخیره شده‌ی خود بلکه تمام داده‌های بارگیری شده آن را نیز مشاهده کند. علاوه بر این، به یک رقیب این امکان را می‌دهد تا با ارسال پیام نادرست هنگام تماس برای تعمیر، خطاهایی را در گره‌های خارج از کنترل خود وارد کند. در تجزیه و تحلیل، به سه نوع از مزاحمان متمرکز می‌شویم:

(1): یک استراق سمع

امنیت سیستم های ذخیره سازی توزیع شده

به عنوان مثال، در حین استراق سمع بر روی یک گره جدید در طی فرآیند تعمیر، متجاوز می‌تواند نه تنها محتوای ذخیره شده‌ی خود بلکه تمام داده‌های بارگیری شده آن را نیز مشاهده کند. علاوه بر این، به یک رقیب این امکان را می‌دهد تا با ارسال پیام نادرست هنگام تماس برای تعمیر، خطاهایی را در گره‌های خارج از کنترل خود وارد کند. در تجزیه و تحلیل، به سه نوع از مزاحمان متمرکز می‌شویم:

(1): یک استراق سمع

(2): یک مخالف فعال همه کاره که دانش کاملی از داده‌های ذخیره شده در

سیستم دارد.

امنیت سیستم های ذخیره سازی توزیع شده

به عنوان مثال، در حین استراق سمع بر روی یک گره جدید در طی فرآیند تعمیر، متجاوز می‌تواند نه تنها محتوای ذخیره شده‌ی خود بلکه تمام داده‌های بارگیری شده آن را نیز مشاهده کند. علاوه بر این، به یک رقیب این امکان را می‌دهد تا با ارسال پیام نادرست هنگام تماس برای تعمیر، خطاهایی را در گره‌های خارج از کنترل خود وارد کند. در تجزیه و تحلیل، به سه نوع از مزاحمان متمرکز می‌شویم:

(1): یک استراق سمع

(2): یک مخالف فعال همه کاره که دانش کاملی از داده‌های ذخیره شده در

سیستم دارد.

(3): یک مخالف فعال با دانش محدود از داده‌های ذخیره شده در سیستم دارد.

با سپاس فراوان از حسن توجه اساتید
بزرگوار

با سپاس فراوان از حسن توجه اساتید بزرگوار

