Mobina Mehrazar 810100216

FalconDB: Blockchain-based Collaborative Database

Yanqing Peng, Min Du, Feifei Li, Raymond Cheng, Dawn Song

Key Words

FalconDB, Blockchain, Collaborative Database, Database Security, Authenticated Data Structures (ADS)

General Subject

امروزه به علت گسترش اینترنت، سیستمی برای همکاری چندین موجودیت و تامین مثلث امنیت[1]، سازگاری[2] و کارایی[3]، به همراه تحمیل هزینه کم به سمت کاربر، مورد نیاز است که راه حل‌های قبلی، مشکلاتی را حل نکرده‌اند.

Specific Subject

قبل از ارائه ایده‌های نوین، اعتماد بین کاربران از طریق اعتماد به سرور مرکزی صورت می‌گرفت که خطر مخرب بود سرور مرکزی و دسترسی خودسرانه کاربر برای اهداف خودش را به همراه دارد.

این مقاله، یک پایگاه‌داده مشترک، ایمن، و کارآمد و نیازمند سخت‌افزار معقول در سمت کاربران را مطرح می‌کند. در پلتفرم بلاک‌چین و ذخیره‌سازی داده ADS این طراحی، حضور یک یا چند گره سرور برای پاسخ‌گویی به درخواست‌های کاربران، مطرح است.

ایده اولیه : تا به حال، تکنیک بلاک‌چین برای از بین بردن مرکزیت سرور و شامل پروتکل اجماع برای حل مشکل شکست بیزانتین[4] در پلتفرم‌هایی[5] ارائه شده که کاربران به علت هزینه بالای تحمیل‌شده برای ذخیره‌سازی کامل بلوک‌ها، اجرای درخواست‌ها به صورت محلی، سرعت نامطلوب، استقبال نشده‌است.

---

[1] Security
[2] Compatibility
[3] Efficiency
[4] Byzantine failure
[5] نظیر HotStuff، Tendermint، Hyperledger و BigchainDB

گره‌های سرور : مسئول پاسخ‌گویی به درخواست و تغییرات کاربران هستند و همه‌ی پایگاه‌داده و بلاک‌چین در داده‌ساختارهای تایید شده را ذخیره کرده‌است. تنها از طریق درخواست کاربران قادر به تغییر روی داده هستند.[6]

گره‌های کاربر : محتوای پایگاه‌داده را به صورت محلی ندارند و تنها سرصفحه‌ها[7]را ذخیره دارند. دسترسی خواندن و نوشتن در بخش دلخواه پایگاه‌داده را از طریق درخواست به گره سرور، دارند و از طریق تعامل با ADS از درستی نتایج مطمئن می‌شوند.

قرارداد هوشمند[8] : برنامه کامپیوتری و ناظر مستقیم دارایی‌های دیجیتال است. کاربران با این برنامه تعامل می‌کنند.

داده‌ساختارهای تایید شده (ADS) : مسئول احراز هویت پرس و جو[9]ها است.

سرور نتایج درخواست و امضا را فراهم می‌کند. با این حال، کاربران می‌توانند با پرداخت هزینه اضافی، درخواست تولید اثبات[10] کنند. در این طراحی، هویت همگی گره‌ها مشخص است و یک بلاک‌چین مجاز[11] داریم. امنیت با اجماع BFT[12] تضمین شده و گزارش[13]ها توسط همگی گره‌ها مورد توافق واقع می‌شود.

درخواست‌های Standard که بر روی آخرین ورژن تمرکز می‌کنند و رکوردهایی که منقضی نشده‌اند را انتخاب می‌کنند. $(VT = \infty)$

درخواست‌های Full Historical که همه‌ی سوابقی که شرط خاصی را برآورده می‌کنند را خروجی می‌دهد.

درخواست‌های Range Historical که رکوردهای دارای شرایط range (متغیر VT و VF) را خروجی می‌دهد.

درخواست‌های Delta که اعمال انجام‌شده روی تراکنش‌های مشخص روی بلوک مدنظر را خروجی می‌دهد.

آپدیت insertion که رکورد با $VF = h$ (ارتفاع کنونی بلوک) و $VT = \infty$ را به پایگاه‌داده اضافه می‌کند.

آپدیت deletion که مقدار VT رکورد را به ارتفاع بلوک فعلی تغییر می‌دهد. $(VT = h)$

آپدیت value change که رکورد قبلی را delete کرده و رکورد جدید را insert می‌کند.

Methodology

ستاپ اولیه سیستم : قبل از اجرای پروتکل، پارامترها و توابع زیر باید توسط همگی گره‌ها پذیرش شوند : بلوک hardcode شده‌ی نخستین B_0، یک تابع هش $(hash(s) \to s')$، کلید عمومی و خصوصی (pk_i, sk_i)، تابعی برای امضا کردن $(sign(sk, s) \to s')$، تابع برای تایید صلاحیت امضا $(VerifySig(pk, s, s'))$.

هر بلوک شامل یک تراکنش پایگاه‌داده با سایز دلخواه می‌باشد. در هر بلوک $B = (H, C)$، محتوای بلوک (C) (حاوی تراکنش) و سرصفحه‌ی بلوک $H = (M, V)$، metadata و verification data ذخیره می‌شود.

---

[6] در غیر این صورت، اثبات مجازی برای درخواست‌های آینده نخواهد داشت چرا که خلاصه واقعی پایگاه‌داده با خلاصه موجود در بلاک‌چین مغایرت دارد.

[7] block headers
[8] smart contract
[9] Query Authentication
[10] proof generation
[11] permissioned blockchain
[12] Byzantine fault tolerance
[13] logs

فیلدهای metadata شامل ارتفاع یا همان اندیس بلوک ($height$)، هش آخرین بلوک زنجیره، هش بلوک کنونی، خلاصه[14]از ورژن کنونی پایگاه‌داده، هش اعمال درخواست‌شده از سرور ($hash(RW)$) و مشخصات سرور می‌باشد. فیلدهای block validation یا V، شامل امضای سرور ($s_0 = sign(sk(e_0), M)$)، گره‌ها و امضای گره‌هایی که بلوک را مورد بررسی قرار داده‌اند.

مراحل Update Authentication process :

1. سرور با دریافت درخواست آپدیت کاربر، دسترسی‌های آن کاربر بررسی می‌شود. در صورت صلاحیت، تعامل بین کاربر و سرور برای انجام تغییرات و تولید digest های مربوطه صورت می‌گیرد.
2. همه گره‌های کامل به روی نتیجه‌ی حاصل به اجماع می‌رسند[15] و سرور گیرنده درخواست، یک بلوک حاوی خلاصه‌های نهایی به همراه گزارش تعامل‌ها و همه‌ی خلاصه‌های تولیدشده در فرایند، تولیدکرده و به شبکه پیشنهاد می‌دهد.
3. همه‌ی گره‌ها دسترسی کاربر و درستی خلاصه‌های بلوک را با مراجعه به گزارش‌ها بررسی می‌کنند.
4. پس از اعمال پروتکل اجماع، بلوک جدید تایید شده و همه‌ی گره‌ها خلاصه‌های محلی خود را به روز می‌کنند.

Result(s)

درستی کارکرد طراحی : کاربر مجموعه‌ی تغییرات از سمت سرور موجود در سرصفحه‌ها را اعتبارسنجی می‌کند. با این حال، می‌تواند درخواست احراز به قرارداد هوشمند بدهد و با پرداخت پول بیشتر به سرور، از سرور ADS proof گرفته و از طریق خلاصه تولید شده درستی‌سنجی انجام دهد.

جریمه گره‌های خراب‌کار : در صورت تشخیص خرابکار بودن گره سرور، همه‌ی پاداش و اکانت آن سرور بسته می‌شود. در صورت تشخیص خرابکار بودن گره کاربر، دسترسی‌های او توسط قرارداد هوشمند گرفته می‌شود.

تراکنش‌های متعارض : همانند پایگاه‌داده‌های سنتی، کنترل OCC[16] استفاده می‌شود تا مشخص شود بلوک حاوی تراکنش مربوطه، هیچ conflict ای در طول بلوک مشخص‌شده و بلوک کنونی نباشد. در صورت رد صلاحیت، بلوک رد شده و تراکنش لغو می‌شود.

چند شاخگی : این پروتکل، تضمین می‌کند شبکه در نهایت بر روی یک زنجیره بدون fork توافق می‌کند و دوشاخگی (دو بلوک با ارتفاع و دو تراکنش و commit timestamp یکسان) نداریم.

تغییر ناپذیری[17] و شفافیت[18]: همگی تغییرات در هر مرحله به طور دائمی در پایگاه‌داده ذخیره شده و قابل تایید و در طول زمان قابل بازیابی هستند.

اضافه‌شدن گره :یک از تفاوت بزرگ این طراحی با طراحی‌های گذشته این است که کاربر جدید، می‌تواند مخرب بودن بقیه گره‌ها را تشخیص دهد. کاربر از طریق ADS، همگی بلوک‌ها را صحت‌سنجی می‌کند و از

---

[14] digest

[15] در نتیجه یکپارچگی (integrity) با وجود اکثریت نودهای صادق تضمین می‌شود.

[16] optimistic concurrency control (OCC)

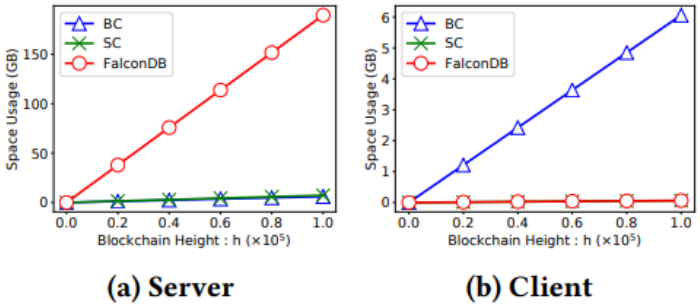
[17] immutability

[18] transparency

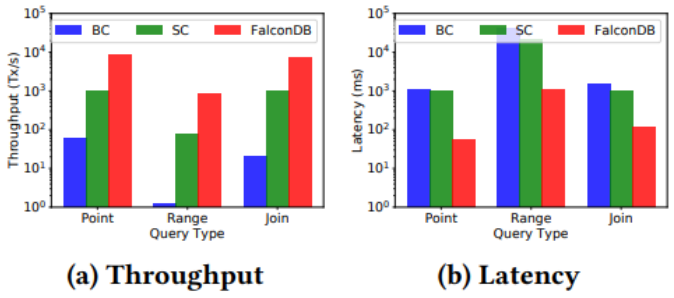
آنجا که خلاصه اولیه در اختیار همه قرار گرفته، به سرعت به اولین بلوک غیرقابل قبول در زنجیره رسیده و آن را رد می‌کند. شبکه با وجود تنها یک گره سالم هم به کار خود ادامه می‌دهد و سرور جدید معرفی می‌شود[19].

نگرانی حریم خصوصی : مانند بقیه پایگاه‌داده‌های عمومی و شفاف، به علت در دسترس بودن داده این نگرانی دارد. برای حل مشکل، داده‌ها و گزارش‌ها باید رمزگذاری شوند که پشتیبانی این روش، به علت قابل بررسی بودن درستی توسط کاربران دشوار است.

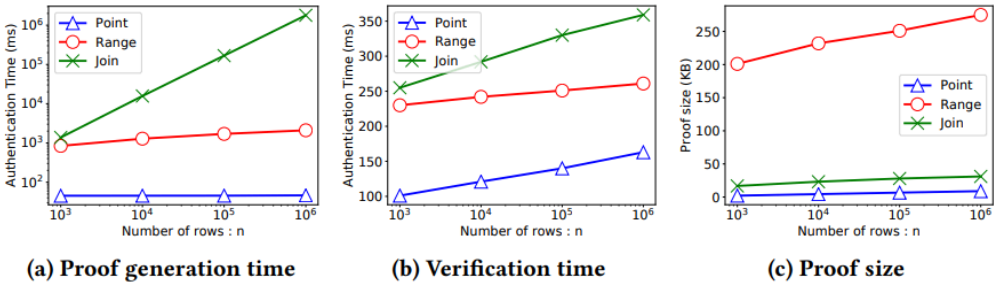
مقایسه عملکرد FalconDB با BC[20] و SC[21] با یک پلتفرم بلاک‌چین زیربنایی :



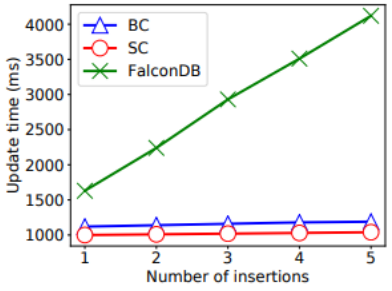
مقایسه‌ی میزان فضای مصرف شده برای جدولی با 10h سطر و 10 ستون. FalconDB، هزینه‌ی غالب را از دوش کاربر برداشته و به سرور تحمیل می‌کند.



مقایسه‌ی عملکرد پرس‌وجو روی جدولی با 106 سطر و 10 ستون.



مقایسه‌ی تاخیر authentication به روی جدولی با n سطر و 10 ستون.



---

[19] server reliability and system liveness
[20] a naive blockchain based shared database where each user stores a full data copy
[21] a smart contract based solution where a user could submit smart contracts to query full nodes and get consented results

مقایسه‌ی عملکرد به‌روزرسانی به روی جدولی خالی با 10 ستون.

Summary of key Points

اولین پلتفرمی که به کاربران اجازه‌ی همکاری بر روی پایگاه‌داده‌ی امن، با هزینه معقول و کارایی بالا، به روی سیستمی غیر متمرکز ارائه کرده‌است.

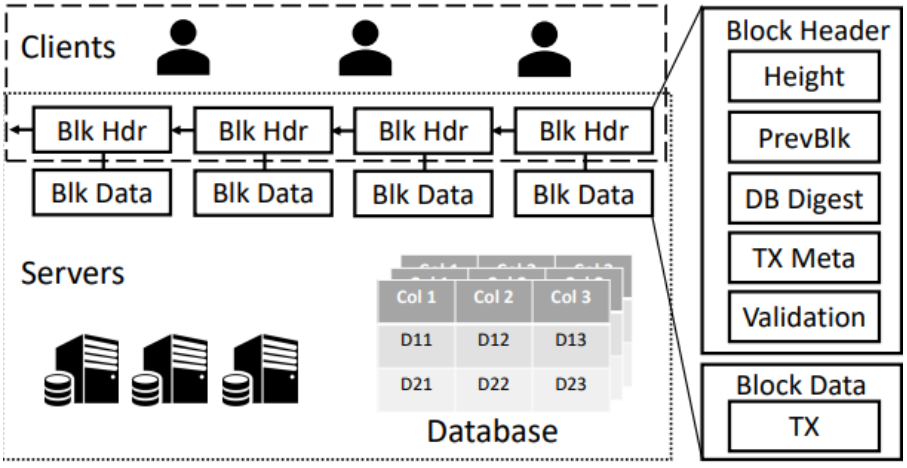
Context

این مقاله با به‌کار گیری پلتفرم بلاک‌چین و استفاده از پروتکل‌های آن، مدل پایگاه‌داده‌ای کارا معرفی کرده که طبق ویژگی‌های آن، حریم خصوصی وجود ندارد.

Significance

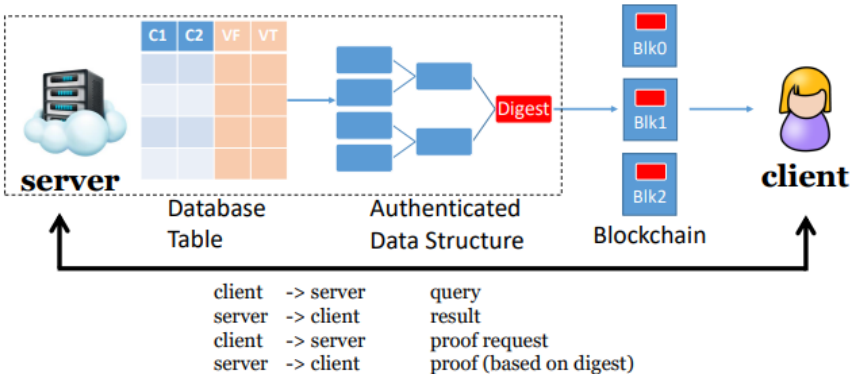
به علت نیاز به پایگاه‌داده با هزینه معقول، کارا و امن این طراحی معرفی شده‌است. پلتفرم بلاک‌چین و پروتکل اجماع[22] آن و سیستم توزیع‌شده، نیاز به تامین اعتماد بین موجودیت‌ها را از دوش کاربران برداشته و هزینه غالب را به سرور تحمیل کرده و مدلی انگیزشی ارائه داده‌است. طراحی تا ⅓ کل گره‌های کل به عنوان گره‌های مخرب را تحمل می‌کند و می‌تواند با وجود تنها یک نود صادق در شبکه، به کار خود ادامه دهد.

Important Figure and/or Tables



همانطور که از تصویر مشخص است، کاربران تنها سرصفحه‌ها را ذخیره می‌کنند و سرورها نسخه‌ی کامل بلاک‌چین و محتوای پایگاه‌داده را ذخیره دارند. کاربران به سرور دلخواه خود از طریق permissioned blockchain protocol بوده تعامل می‌کنند.

[22] Consensus



بعد از تولید خلاصه‌ی حاصل از ADS، نتیجه بر صفحه بلاک‌چین ذخیره شده و کاربران با این صفحات دسترسی داشته و از آن برای احراز نتایج گرفته‌شده از سمت سرور استفاده می‌کنند.

Other Comments

اپلیکیشن‌های زیادی می‌توانند از طراحی FalconDB بهره ببرند، از جمله :

Credit score ها که از یک طراحی برای یک دفتر اعتباری غیر متمرکز استفاده کرده و به نهادها امکان به‌روزرسانی و تایید مستقیم اطلاعات اعتباری را با تضمین کردن اعتماد و قابلیت ردیابی، می‌دهد.

در **Banking** ها که به احتیاج به انتقال پول بین‌بانکی را با تضمین اعتماد پاسخ می‌دهد.

ایده‌ی جالب دیگر، استفاده در **Government audit** بوده که درج مستقیم رکورد و درخواست‌های عمومی برای نظارت شفاف و قابل اعتماد بین آژانس‌های فدرال و عموم مردم را خواهیم داشت.

Cited References to Follow up on

MPV: [Enabling Fine-Grained Query Authentication in Hybrid-Storage Blockchain]

در این مقاله به انرژی مصرف‌شده در ذخیره‌سازی بلاک‌چین هیبریدی (HSB) برای مدیریت داده‌های مقیاس بزرگ و یکپارچه‌سازی ذخیره‌سازی داده با به کار بردن جفت روش‌های روی زنجیره و خارج زنجیره اشاره می‌شود. همانطور که در FalconDB دیدیم، همگی پایگاه‌داده روی زنجیره شبکه ذخیره نشده، بلکه گره‌های کامل این هزینه را به دوش می‌گیرند. ایده‌ی یکپارچه‌سازی MPV، دو طرح راستی‌آزمایی مبتنی بر برابری چندبعدی بوده تا برای احراز هویت در range query می‌باشد. این طرح‌ها شامل ADS های مبتنی بر انباشته‌سازی بوده و ایده‌ی مشترک، در FalconDB برای احراز هویت استفاده شده‌است.

A Comparative Testing on performance of Blockchain and Relational Database : Foundation for applying Smart Technology into Current Business Systems

این مقاله به بررسی ظرفیت پردازش تراکنش بلاک‌چین و تنگناهای آن پرداخته و پیشنهاد می‌دهد در پایگاه‌داده‌های مبتنی بر بلاک‌پین، داده‌های کوچک در زنجیره ذخیره شوند. همانطور که در FalconDB دیدیم، داده‌ی جدول‌ها که عموما بسیار بزرگ هستند، در حافظه‌ی محلی گره‌های کامل ذخیره شده و در زنجیره، اطلاعات تراکنش و خلاصه و گزارش‌ها ذخیره می‌شوند.

aChain: [A SQL-Empowered Analytical Blockchain as a Database]

این مقاله، رابطه‌هایی بر روی زنجیره تعریف می‌کند که در عین ایمن بودن، خدمات SQL، که معماری execute-order-validate دارند، را پشتیبانی کنند و این پشتیبانی اتمی و سازگار بوده و عملکرد مشابهی داشته‌باشند. همانطور که در FalconDB دیدیم، متغیرها و توابعی برای دستیابی به نتایج ذخیره‌شده در زنجیره تعریف شده‌است که محدوده‌ای از درخواست‌های مبتنی بر SQL را پاسخ می‌دهند. این محدوده، منحصر به عملکرد FalconDB بوده و درخواست‌هایی که پشتیبانی نمی‌شوند، اساسا برای عملکرد FalconDB لازم نبوده‌اند.

BigchainDB : https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf

در این مقاله راجب اکوسیستم غیر متمرکز سازی کلی BigchianDB بحث شده و روش‌هایی برای امتحان‌کردن این طراحی پیشنهاد شده است. در FalconDB، دیدیم آنالیزهای متعددی روی نحوه‌ی عملکرد و موارد مورد مقایسه بحث شده است که ایده‌هایی از آن، از این مقاله بسیار مشابه گرفته شده‌است.

در پایگاه‌داده‌های خیلی بزرگ، که محتوای ذخیره‌شده امکان دسته‌بندی شدن دارند، ایده‌ی فعالیت یک گره همزمان به عنوان گره کاربر و سرور مطرح است. در این ایده، گره مورد بحث، به نسبت سخت‌افزاری که دارد، بخش قابل قبولی از پایگاه‌داده را ذخیره کرده و در ازای پاسخ‌گویی به درخواست‌های قابل پشتیبانی، پاداش می‌گیرد. در صورتی که درخواستی بر روی داده ذخیره‌نشده داشته باشد، از سرورهای دیگر کمک می‌گیرد. باید اجازه‌ی اعمال تغییر مطلوب خود این دست گره‌ها توسط خودشان، گرفته شود. برای تشویق گره‌های کامل که همه‌ی پایگاه‌داده را دارند، می‌توانیم سیستم انگیزشی را به نفع آن‌ها تغییر دهیم.

ID of Telegram : @mobinamehrazar

Representation URL