A-SIT Technology
E-Government Innovationszentrum

26. November 2020 Studie · Blockchain · 13 Min. Lesezeit

# Komprimierbare Blockchain

**Archiviert.** Dieser Inhalt wird nicht mehr aktualisiert. Er bleibt zur Information verfügbar.

**Stand** 2020-11-26
**Quelle** https://technology2.a-sit.at/veroeffentlichungen/komprimierbare-blockchain/

In diesem Projekt wird ein Ansatz für eine komprimierbare Blockchain vorgestellt. Der Ansatz sieht eine zweite, verlinkte Blockchain vor, in dessen Blöcken das aktuelle UTXO-Set gespeichert wird. Dadurch muss ein neuer Knoten nur den letzten Block in der zweiten Kette, sowie alle danach erzeugten Blöcke in der ersten Kette laden um alle relevanten Informationen zu bekommen. Mit diesem Ansatz ergeben sich im besten Fall Speicherplatz- und Bandbreiteneinsparungen von 93%.

## 1. Einleitung

In diesem Projekt wird ein Ansatz für eine komprimierbare Blockchain vorgestellt. Durch die verringerte Größe kann die Blockchain von neuen Teilnehmern im Netzwerk wesentlich schneller geladen werden. Je nach verwendetem Computer und Internetverbindung kann die Synchronisierung mit der Bitcoin-Blockchain derzeit mehrere Tage bis Wochen dauern. Im Vergleich zu sogenannten Lightweight Nodes, welche in Abschnitt 2.3 beschrieben werden, bietet dieser Ansatz zudem erhöhte Sicherheit und Privatsphäre. Weiteres erlaubt dieser Ansatz, nach einer vollständigen Synchronisation das Netzwerk aktiv zu unterstützen, ähnlich zu sogenannten Full Nodes (siehe Abschnitt 2.2), welche die gesamte Blockchain herunterladen und speichern. In den nächsten Abschnitten wird zuerst das nötige Hintergrundwissen vermittelt und anschließend der neue Ansatz für eine komprimierbare Blockchain vorgestellt.

## 2. Hintergrund Wissen

In diesem Bericht wird davon ausgegangen, dass ein Grundverständnis von Bitcoin vorhanden ist. Daher werden nur die für den vorgestellten Ansatz relevanten Aspekte beschrieben. Für die Grundlagen wird auf das Bitcoin Whitepaper [1] verwiesen.

### 2.1. UTXO Set

In der Bitcoin-Blockchain besteht jede Transaktion aus einem oder mehreren Transaktions-Inputs und Transaktions-Outputs. Diese werden in Folge als Inputs und Outputs bezeichnet. Outputs können als Inputs in darauffolgenden Transaktionen verwendet werden, wodurch Währungseinheiten verschoben oder

aufgeteilt werden können. Eine Ausnahme sind sogenannte Coinbase-Transaktionen. Diese Transaktionen haben keinen Input, dürfen aber eine definierte Menge an Währungseinheiten erschaffen. Diese Outputs stellen die Belohnung für Miner da, die einen gültigen Block gefunden und die Blockchain erweitert haben. Die Menge aller noch nicht ausgegebenen Outputs wird als UTXO-Set bezeichnet. UTXO steht für „Unspent Transaction Output“. Mittels dieses Sets können neu empfangene Blöcke und Transaktionen effizient auf ihre Gültigkeit überprüft werden, ohne die gesamte Blockchain durchsuchen zu müssen. Ein Vorteil dieses Sets gegenüber der gesamten Blockchain ist zudem, dass das UTXO Set nicht stetig größer wird, sondern auch schrumpfen kann. Im Vergleich dazu werden an die Blockchain laufend Blöcke angehängt, wodurch diese mit der Zeit immer größer wird. Mitte September 2020 benötigte die Bitcoin-Blockchain bereits ca. 300GB Speicherplatz [2]. Abbildung 1 zeigt das Wachstum der Bitcoin-Blockchain seit ihrem Start. Es ist zu sehen, dass der benötige Speicherplatz anfangs nur sehr langsam stieg. In den letzten Jahren wurde Bitcoin bekannter und beliebter, was sich auch in volleren Blöcken und dadurch größerem Wachstum zeigt. Im Vergleich zu den ca. 300GB, benötigt das derzeitige UTXO-Set ca. 3,8GB [3], wobei die durchschnittliche Größe ca. 2,5GB beträgt.

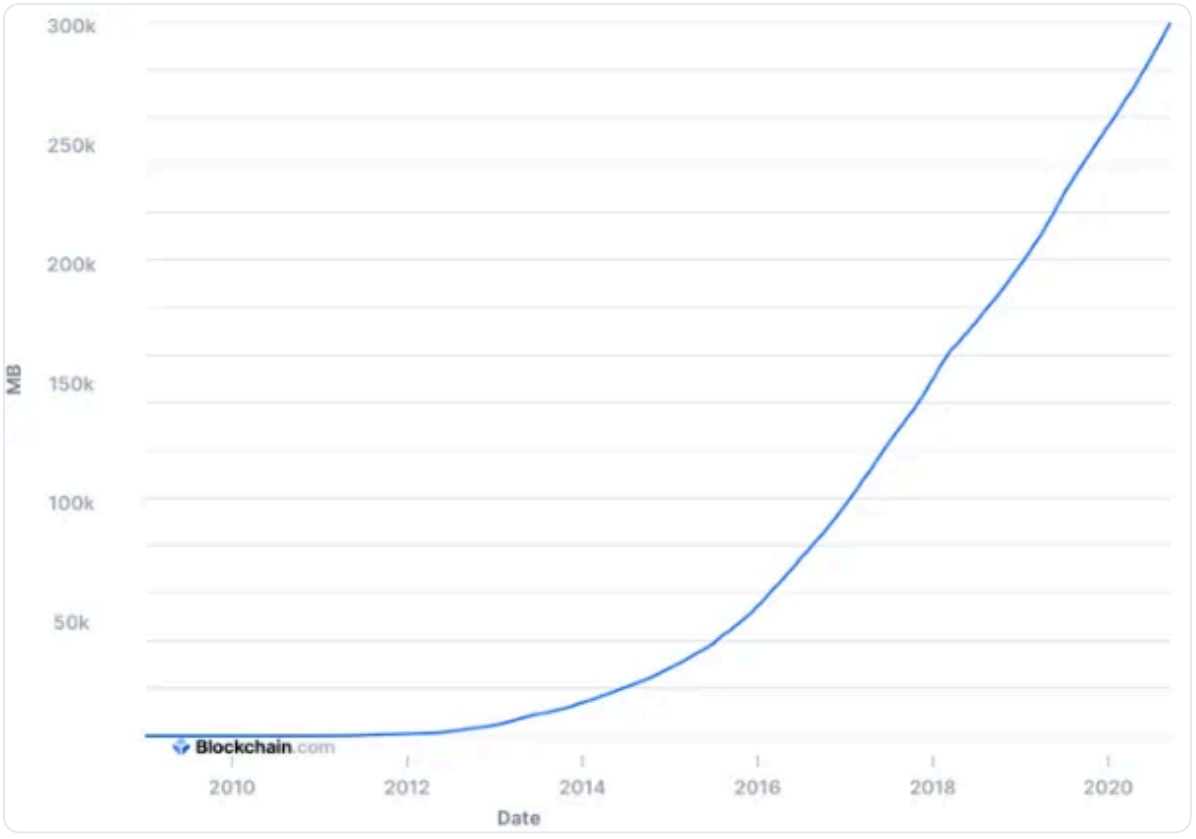



Abbildung 1: Größer der Bitcoin-Blockchain (Quelle: [2])

Um das UTXO-Set aktuell zu halten, muss es nach jedem akzeptierten Block aktualisiert werden. Dabei werden alle Outputs, die in dem neuen Block als Inputs verwenden werden, entfernt und alle neu erstellten Outputs hinzugefügt. Dabei werden die Transaktionen genau in der Reihenfolge abgearbeitet, wie sie im Block hinterlegt sind. Das UTXO-Set ist speziell für Full Nodes relevant. Diese werden im nächsten Abschnitt vorgestellt.

## 2.2. Full Node

Bei einem Full Node handelt es sich um eine Software, welche sich zum Bitcoin-Netzwerk verbindet, die gesamte Blockchain herunterlädt, validiert und speichert. Die Software verifiziert jeden Block und jede Transaktion auf Einhaltung der Regeln. Blöcke müssen beispielsweise den Vorgängerblock referenzieren, dürfen nur einen definierten Maximalbetrag im Rahmen der Coinbase-Transaktion auszahlen und alle enthaltenen Transaktionen müssen gültig sein. Zudem dürfen Transaktionsoutputs nicht mehrmals ausgegeben werden oder mehr ausbezahlt werden, als einbezahlt wurde. Es dürfen keine negativen Beträge vorkommen. Weiters muss die im Transaktionsoutput definierte Bedingung erfüllt sein, um diesen Output ausgeben zu können. In der Regel ist eine gültige Signatur erforderlich. Nachdem sich ein Full Node mit dem Netzwerk synchronisiert hat, d.h. die gesamte Blockchain geladen und verifiziert hat, beginnt die Software das Netzwerk zu unterstützen. Beispielsweise durch die Verifikation von empfangenen Transaktionen, die, sofern sie gültig sind, weiter im Netzwerk verteilt werden. Als Basis für diese Überprüfung zieht jeder Full Node jeweils die aktuelle lokale Kopie der Blockchain heran. Im Idealzustand haben alle Knoten im Netzwerk dieselbe lokale Kopie. Durch Netzwerkverzögerungen und andere Ereignisse kann es jedoch vorkommen, dass Knoten temporär unterschiedliche lokale Kopien haben. Da definiert ist, dass die regelkonforme Kette mit dem größten summierten Proof-of-Work als gültig zu werten ist, einigen sich alle Knoten üblicherweise relativ schnell auf eine gemeinsame Kette. Der Hauptvorteil eines Full Nodes liegt in der Sicherheit, da keinem fremden Anbieter vertraut werden muss. Zudem wird dem Netzwerk durch die Verifizierung und Weiterleitung von Transaktionen und Blöcken geholfen. Der größte Nachteil liegt im benötigen Speicherbedarf, welcher derzeit bei einigen hundert Gigabytes liegt [1] . Zusätzlich muss ein Full Node noch das derzeit aktuelle UTXO-Set errechnen und aktuell halten. Das UTXO-Set enthält alle noch nicht ausgegebenen Transaktionsoutputs und benötigt einige Gigabyte an Speicher. Dieses Set wird benötigt, um neue Transaktionen schnell und effizient auf Gültigkeit zu überprüfen. Da der Speicherbedarf eines Full Nodes mit der Zeit immer wächst, aber nicht alle Geräte über so viel Speicherplatz verfügen bzw. der Speicher anderweitig benötigt wird, wurden Ansätze entwickelt, wie ein Bitcoin Knoten mit geringeren Speicher- oder Bandbreitenanforderungen betrieben werden kann. Zwei dieser Ansätze, die von Bitcoin Core, der Referenzimplementierung von Bitcoin unterstützt werden heißen „Pruning“ and „blocksonly“. Diese werden in den nächsten beiden Abschnitten kurz vorgestellt.

### 2.2.1. Pruning

Seit Bitcoin Core Version 0.11.0 (2015) wird Pruning unterstützt [4]. Pruning erlaubt das Betreiben eines Full Nodes ohne die komplette Blockchain lokal zu speichern. Stattdessen wird ein Limit für den maximal zu verwendeten Speicherplatz angegeben. Der Knoten speichert nur die aktuellsten Blöcke und löscht ältere sobald das Limit überschritten wird. Bei diesem Ansatz muss trotzdem die gesamte Blockchain heruntergeladen und verifiziert werden. Sollte es zu Problemen kommen oder ein „rescan“ notwendig sein, muss die gesamte Blockchain neu geladen werden.

### 2.2.2. Blocks Only

In Bitcoin Core Version 0.12 wurde die Option „-blocksonly“ hinzugefügt. Diese ermöglicht, den Netzwerkverkehr zu reduzieren. Wird diese Option aktiviert, fordert der Knoten keine unbestätigten Transaktionen mehr an und leitet diese auch nicht weiter. Somit reduziert der Knoten den Netzwerkverkehr

auf ein notwendiges Minimum [5]. 2016 wurde bei einem Experiment ein um 88% reduzierter Netzwerkverkehr beobachtet, wenn „-blocksonly“ aktiviert wurde [6]. Eine andere 1 Stand 15.9 ca. 300GB für die Blockchain.

Möglichkeit, Speicher- und Bandbreite zu sparen, bieten Lightweight Nodes. Diese werden im nächsten Abschnitt vorgestellt.

### 2.3. Lightweight Node

Lightweight Nodes laden nicht die komplette Blockchain herunter und benötigen daher weniger Speicherplatz und Bandbreite. Stattdessen laden Lightweight Nodes nur die Blockheader herunter. Dadurch können sie überprüfen, ob die Kette prinzipiell gültig ist, d.h. ob die Verlinkung und der Schwierigkeitsgrad passen. Zur Transaktionsvalidierung wird ein Ansatz namens „Simplified Payment Verification“ oder kurz SPV verwendet. Mittels SPV kann ein Lightweight Node sicherstellen, dass eine Transaktion in einem Block aufgenommen wurde. Dafür muss der Lightweight Node einen Full Node kontaktieren, welcher eine Bestätigung schickt. Weiteres benachrichtigt der Full Node den Lightweight Node über Transaktionen, die ihn betreffen. Lightweight Nodes müssen dem Full Node vertrauen, dass dieser Transaktionen und Blöcke entsprechend den Regeln verifiziert und auch, dass sie über alle relevanten Transaktionen informiert werden. Zusätzlich hat die Verwendung eines Lightweight Nodes Auswirkungen auf die Privatsphäre, da zu beobachtende Adressen bekannt gegeben werden müssen. Daher sollten Lightweight Nodes nur mit eigenen oder vertrauenswürdigen Full Nodes verbunden werden. Im nächsten Abschnitt wird ein neuer Ansatz vorgestellt, welcher höhere Sicherheit und Privatsphäre bietet und gleichzeitig den Speicherplatz und Bandbreitenbedarf senkt.

## 3. Komprimierbare Blockchain

Ziel dieser Idee ist es, die Synchronisation schneller und mit weniger Speicher- und Bandbreitenbedarf zu ermöglichen. Weiteres soll nach der erfolgreichen Synchronisation keine Abhängigkeit von Full Nodes bestehen. Stattdessen soll der Node selbst Blöcke und Transaktionen verifizieren können und das Netzwerk unterstützen. Die Grundidee ist, neben der Hauptkette eine zweite Kette zu bauen, deren Blöcke das aktuelle UTXO-Set enthalten. Diese Blöcke werden Snapshot-Blöcke genannt und enthalten alle relevanten Informationen aus allen Blöcken der ersten Kette bis zum Erstellungszeitraum. Mithilfe dieser Informationen können neue Transaktionen effizient auf Gültigkeit überprüft werden. Die Validität des jeweiligen Snapshot-Blockes wird über die Verlinkung durch die Blöcke in der ersten Kette sichergestellt. Der Ansatz ist in Abbildung 2 visualisiert. Blöcke in der Hauptkette (in schwarz dargestellt) zeigen neben ihrem Vorgängerblock auch noch auf den derzeit aktuellsten Snapshot-Block. Gibt es noch keinen Snapshot-Block, wird stattdessen der gemeinsame Genesis-Block verlinkt. Der erste Block wird Genesis-Block genannt und dient als vertrauenswürdiger Startpunkt. Dies ist in Abbildung 2a dargestellt. Die Snapshot-Blöcke werden regelmäßig erstellt, beispielsweise alle 10.000 Blöcke. Abbildung 2b zeigt eine komprimierbare Blockchain mit einem Snapshot-Block und Abbildung 2c zeigt die komprimierbare Blockchain nachdem ein zweiter Snapshot-Block hinzugefügt wurde. Es ist nun erkennbar, dass die Snapshot-Blöcke eine zweite Blockchain bilden (in grün dargestellt).

(a) Eine Standard Blockchain mit einer zusätzlichen Referenz auf den letzten Block in der Snapshot-Kette bzw. solange kein Block vorhanden ist auf den gemeinsamen Genesis-Block.

| Genesis | BP BP BP 2 δ-1 δ Snapshot 1 | Col3 |
|---|---|---|
| Genesis | BPδ BPδ-1 Snapshot 1 BP2 | Snapshot 1 |

(b) Komprimierbare Blockchain mit einem Snapshot-Block.

(c) Komprimierbare Blockchain mit zwei Snapshot-Blöcken.

*Abbildung 2: Komprimierbare Blockchain, bei der regelmäßig ein Snapshot-Block erstellt wird. Die Snapshot- Blöcke formen eine zweite Kette.* Jeder Snapshot-Block enthält das zum Erstellungszeitpunkt aktuelle UTXO Set sowie eine Referenz zu dessen Vorgängerblock. Da die Snapshot-Blöcke von der ersten Kette verlinkt werden, kann jeder Knoten beim Synchronisieren überprüfen, ob der empfangene Snapshot-Block, dem verlinkten entspricht. Wenn sich ein neuer Knoten mit dem Netzwerk verbindet erhält er im Idealfall von allen Nachbarn dieselbe Information über die derzeit gültige Kette. Falls nicht, muss der Knoten alle Ketten betrachten und die beste auswählen. Zuerst lädt der Knoten alle Blockheader und überprüft, ob die Verlinkung und die Schwierigkeitsgrade korrekt sind. Schlägt diese Überprüfung fehl werden alle ungültigen Blöcke verworfen. Nachdem alle Ketten überprüft wurden, wird die (gültige) Kette mit dem größten kombinierten Proof-of-Work ausgewählt. Im nächsten Schritt lädt der Knoten den letzten, oder vorletzten Snapshot-Block [2] und alle danach folgenden Blöcke in der ersten Kette. Anschließend muss noch das im Snapshot-Block enthaltene UTXO-Set aktualisiert werden. Dazu werden alle Transaktionen in allen zuvor geladenen Blöcken aus der ersten Kette berücksichtigt. Am Ende dieses Vorganges verfügt der Knoten über das aktuelle UTXO-Set und kann aktiv neue Blöcke und Transaktionen verifizieren und im Netzwerk weiterverteilen.

Abbildung 3 zeigt die Größe der Bitcoin-Blockchain, der einzelnen Blöcke sowie des UTXO-Sets in Abhängigkeit von der Blocknummer in linearer Darstellung (Abbildung 3a) sowie mit logarithmischer Y-Achse (Abbildung 3b). Abbildung 3a verdeutlicht die Größenunterschiede der Blockchain im Vergleich zum UTXO-Set. Durch die logarithmische Y-Achse in Abbildung 3b sind die unterschiedlichen Blockgrößen gut erkennbar. Es ist zu sehen, dass zu Beginn die Blöcke relativ klein waren, dann aber immer voller und größer wurden, bis schließlich die maximale Blockgröße von 1MB erreicht wurde. Mit Block 4777120 wurde BIP 91, bekannt als „Segregated Witness“ aktiviert. Bei „Segregated Witness“ wird die Größe der Zeugendaten nur zu 25% berücksichtigt, wodurch die berechnete Blockgröße unter dem 1MB Limit liegt, gleichzeitig Blöcke aber mehr als 1MB an Speicher benötigen können.

| Col1 | Col2 | Col3 | Col4 | Col5 | Col6 |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| 0 100 | 000 200 | 000 300 | 000 400 | 000 500 | 000 600 |

| Col1 | 250000 200000 [Megabyte] 150000 100000 Größe 50000 0 0 100000 200000 300000 400000 500000 600000 -50000 Blocknummer Blockgröße Blockchain Größe UTXO Größe | Col3 |
|---|---|---|
| (a) | (a) | |

2 Falls gerade erst ein Snapshot-Block erstellt wurde, ist es aus Sicherheitsgründen ratsam, den vorherigen zu laden, oder zu warten, bis mehrere Blöcke den aktuellen Snapshot-Block bestätigen.

| Col1 | 1000000 100000 10000 [Megabyte] 1000 100 10 1 Größe 0,1 0,01 0,001 0,0001 0 100000 200000 300000 400000 500000 600000 Blocknummer Blockgröße Blockchain Größe UTXO Größe | Col3 |
|---|---|---|
| (b) | (b) | |

*Abbildung 3: Visualisierung der Größe der Blöcke, der UTXO Snapshots und der Größe der gesamten Blockchain.* Der deutliche Größenunterschied zwischen dem UTXO-Set und der Blockchain lässt vermuten, dass eine komprimierbare Blockchain wesentlich weniger Speicherplatz benötigen sollte. Dies wird im nächsten Abschnitt evaluiert.

## 4. Evaluierung

Die komprimierbare Blockchain wurde mit drei unterschiedlichen Snapshot-Intervallen evaluiert. Als kleinstes Intervall wurde 10.000 Blöcke gewählt, das mittlere Intervall liegt bei 20.000 Blöcke und das größte Intervall bei 50.000 Blöcke. Dies entspricht einem Zeitraum von ca. 70 bis 350 Tagen. Abbildung 4 zeigt die Menge der Daten die ein neuer Knoten herunterladen müsste abhängig von der Blockchain-Länge.

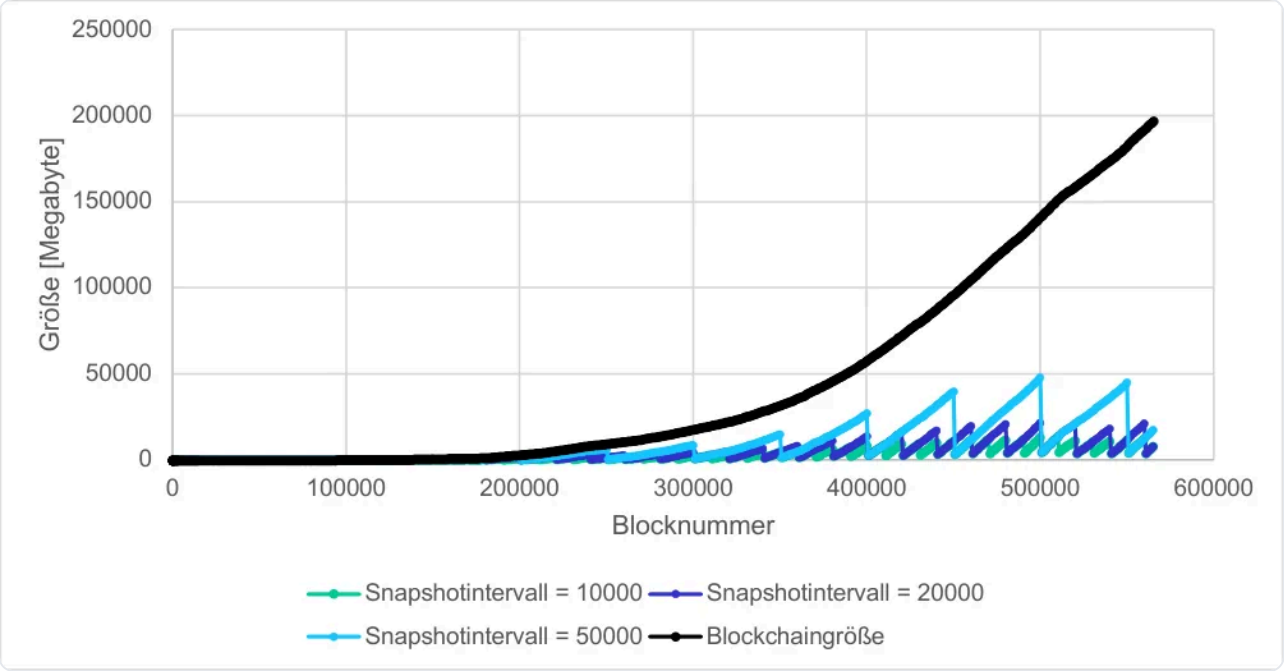



Abbildung 4: Vergleich der Menge an Daten, die bei einer herkömmlichen und einer komprimierbaren Blockchain geladen und gespeichert werden muss. Es ist erkennbar, dass im Vergleich zu einer herkömmlichen Blockchain selbst mit sehr großen Snapshot-Block-Intervallen erheblich weniger Daten geladen und gespeichert werden müssen. Je öfter Snapshot-Blöcke erstellt werden, desto weniger Daten müssen geladen werden. Abbildung 5 fokussiert sich auf die Menge der Daten, die abhängig vom Snapshot-Intervall geladen werden muss. Hier ist deutlich ein Sägezahnmuster erkennbar, dessen Größe mit dem Snapshot-Intervall wächst. Die lokalen Minima entsprechen den Snapshot-Blöcken, das heißt, zu diesen Zeitpunkten muss ein neuer Knoten nur den aktuellsten Snapshot-Block laden. Startet der Knoten den Synchronisationsprozess später, müssen zusätzlich zum aktuellsten Snapshot-Block auch alle seitdem hinzugefügten Blöcke geladen werden. Bei der Bitcoin-Blockchain müssten bei einem Snapshot-Intervall von 10.000 Blöcken im evaluierten Zeitraum, im schlimmsten Fall 13.406 MB geladen werden. Zu diesem Zeitpunkt müsste ein herkömmlicher Full-Node allerdings 151 GB laden, wodurch sich das benötige Datenvolumen sowie der benötigte Speicherplatz um den Faktor 11 vergrößert. Im besten Fall kann das benötige Datenvolumen sowie der benötigte Speicherplatz um ca. 93% reduziert werden. Für mehr technische Details, sowie den Algorithmen zur Erstellung von Snapshot-Blöcken, zur Synchronisation und Verifikation wird auf die zu diesem Projekt gehörende Publikation verwiesen [7].

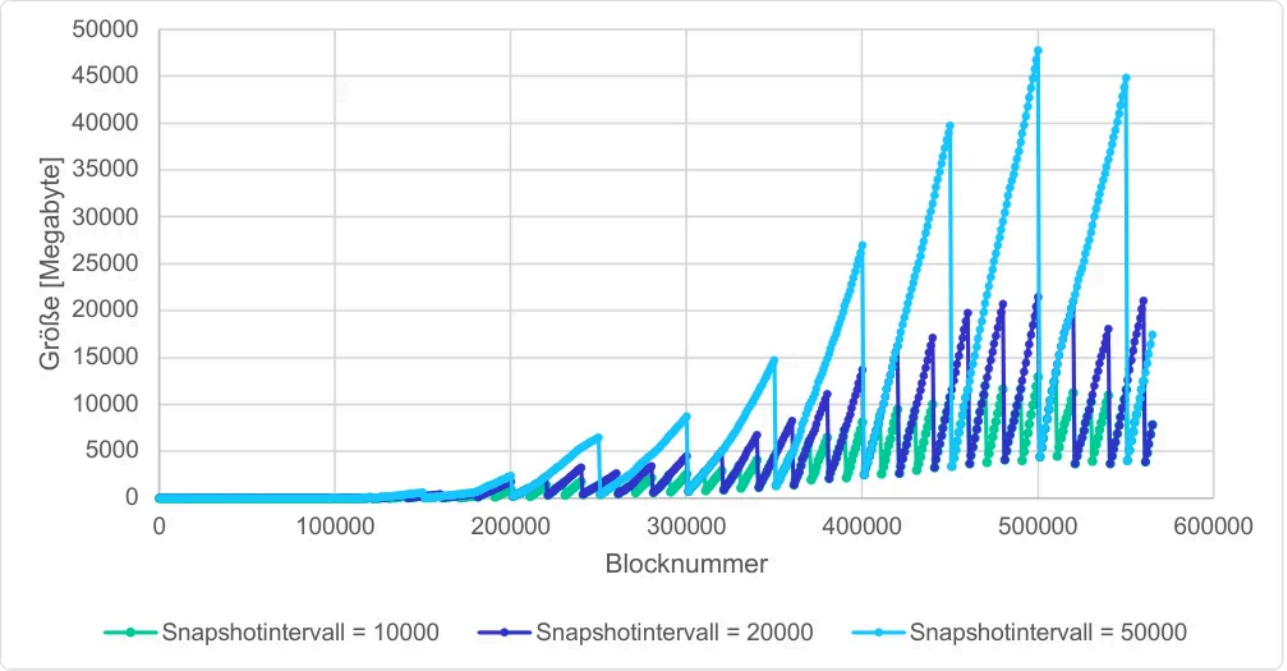



Abbildung 5: Diese Abbildung zeigt dieselben Daten wie die vorherige Abbildung, allerdings ohne der herkömmlichen Blockchain. Dadurch ist die Menge der Daten, die geladen und gespeichert werden muss, besser ablesbar.

## 5. Conclusio

Der gezeigte Ansatz einer komprimierbaren Blockchain erlaubt eine erhebliche Reduktion des benötigen Speicherplatzes sowie der benötigten Bandbreite. Im besten Fall ergibt sich eine Einsparung von 93% bei der Synchronisation. Das Einsparpotential hängt vom gewählten Snapshot-Intervall ab. Kleine Intervalle ermöglichen eine effizientere Synchronisation, bedeuten aber auch mehr Aufwand für Full Nodes und Miner. Full Nodes und Miner müssen zusätzlich die Snapshot-Blöcke speichern, wodurch sich ein erhöhter Speicherplatz Bedarf ergibt. Gleichzeitig müssen Miner öfter Snapshot-Blöcke erstellen wodurch sich ein Mehraufwand ergibt. Die Evaluierung zeigte, dass sich selbst mit großen Snapshot-Intervallen enorme Vorteile für Benutzer ergeben.

## Referenzen

[1] S. Nakamoto, „Bitcoin: A Peer-to-Peer Electronic Cash System," [Online]. Available:

https://bitcoin.org/bitcoin.pdf. [Zugriff am 2020].

[2] Blockchain.com, „Blockchain Size," [Online]. Available:

https://www.blockchain.com/charts/blocks-size. [Zugriff am 18 09 2020].

[3] statoshi.info, „Size of Serialized UTXO Set," [Online]. Available:

https://statoshi.info/dashboard/db/unspent-transaction-output-set. [Zugriff am 18 09 2020].

[4] Bitcoin, „Bitcoin Core version 0.11.0," 2015. [Online]. Available:

https://github.com/bitcoin/bitcoin/blob/v0.11.0/doc/release-notes.md. [Zugriff am 16 09 2020].

[5] Bitcoin Project, „Running A Full Node,“ [Online]. Available: https://btcinformation.org/en/full node#blocks-only-mode. [Zugriff am 16 09 2020].

[6] gmaxwell, „Blocksonly mode BW savings, the limits of efficient block xfer, and better relay,“

26 02 2016. [Online]. Available: https://bitcointalk.org/index.php?topic=1377345.0. [Zugriff am 16 09 2020].

[7] A. Marsalek, T. Zefferer, E. Fasllija und D. Ziegler, „Tackling data inefficiency: Compressing the bitcoin blockchain,“ in *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, Rotorua, 2019.

[8] Sjors, „Bitcoin Core -stopatheight is imprecise #13477,“ 15 06 2018. [Online]. Available:

https://github.com/bitcoin/bitcoin/issues/13477. [Zugriff am 18 03 2020].

[9] Bitcoin Core, „Bitcoin Core,“ [Online]. Available: https://github.com/bitcoin/bitcoin. [Zugriff am 18 03 2020].

[10] S. D. Segura und C. Pérez, „Python Bitcoin tools,“ [Online]. Available: https://github.com/sr gi/bitcoin_tools. [Zugriff am 18 03 2020].

[11] C. Pérez, „kill_at_heigh.py,“ [Online]. Available: https://github.com/sr gi/bitcoin_tools/blob/dev/bitcoin_tools/analysis/status/kill_at_heigh.py. [Zugriff am 18 03 2020].

[12] S. Delgado-Segura, C. Pérez-Solà, G. Navarro-Arribas und J. Herrera-Joancomartí, „STATUS,“ [Online]. Available: https://github.com/srgi/bitcoin_tools/tree/master/bitcoin_tools/analysis/status. [Zugriff am 17 03 2020].