

Addendum: Skalierung bis zehn Millionen Teilchen

Separate Erweiterung der CFK-Versuche vom 16. September 2026

Forschungsprotokoll des Projekts CFK

Beginn der Serie: 16. September 2026

1 Fragestellung und unveränderte Konstruktion

Nach Abschluss des Hauptberichts wurde die zuvor ausgesetzte Millioner-Stufe freigegeben und zusätzlich die Stufe mit zehn Millionen Teilchen beauftragt. Untersucht wird, wie die Zeit des minimalen konformen Baumaufbaus mit der Größe einer gleichverteilten Punktmenge wächst. Auf ausdrückliche Rückfrage wurden zunächst drei Wiederholungen je Größenstufe vereinbart. Der ursprüngliche Bericht und seine Daten bleiben unverändert.

Die Wurzel ist weiterhin das Dreieck mit Ecken $(0, 0)$, $(2, 0)$, $(0, 2)$. Die Punkte liegen in $[0, 1]^2$; die Blätter bedecken das gesamte Wurzeldreieck der Fläche 2. Die Codierung beginnt mit 1, Kinder haben Codes $2k, 2k + 1$. Die exakte Integer-Präfixarithmetik, die Randregel und die Konformitätsregeln werden nicht geändert. Für eine Teilungsmenge S bezeichnet $\mathcal{C}(S)$ den kleinsten Abschluss unter Eltern- und Hypotenusenpartner-Abhängigkeiten. Die verteilte Fassung benutzt weiterhin

$$c\left(\bigcup_i S_i\right) = \bigcup_i \mathcal{C}(S_i).$$

Damit kann jeder Arbeiter einen global konformen Teilbaum liefern; dessen Knoten werden zentral vereinigt. Eine Beschränkung der Teilbäume auf geometrische Buckets wäre nicht dieselbe Konstruktion. Herleitung, Minimalitätsbeweis und API sind im Hauptbericht dokumentiert.

2 Vorab festgelegtes Design

2.1 Größen, Varianten und identische Eingaben

Vorgesehen sind $n = 10^5, 10^6, 10^7$, jeweils drei Wiederholungen mit zwei Varianten, also maximal 18 erfolgreiche Einzelmessungen:

Seriell: Instrumentierte Integerfassung mit einem Prozess und zentralem Konformitätsabschluss.

Parallel: Dieselbe instrumentierte Grundlage, vier vorab gestartete Arbeiter und verteilter globaler Konformitätsabschluss.

Die zusätzliche 10^5 -Stufe ist eine zeitnahe Brücke zu U3. Es wird keine neue Optimierung des Algorithmus in diesen Skalierungsvergleich eingemischt. Jede Messung erhält einen frischen Hauptprozess und gegebenenfalls einen frisch vorbereiteten Pool. Anders als im vorbereiteten U3-Batch

wird der Pool nicht zwischen Wiederholungen wiederverwendet. Sein Start ist trotzdem außerhalb des Aufbauzeitfensters und separat protokolliert. Die Variantenreihenfolge wechselt mit der Wiederholung; die Größen werden aufsteigend gemessen. Das ist kein vollständig randomisiertes Design.

Die Seeds folgen unverändert

$$s_{n,r} = \text{int}_{\text{big}}(\text{SHA256}(\text{cfk-uniform-v1:20260916:n:r})).$$

Die ASCII-Zeichenfolge enthält dezimales n, r ohne Leerzeichen. Ein MT19937-Generator zieht pro Punkt zweimal 53 Zufallsbits und skaliert mit 2^{-53} . Die Verteilung ist somit diskret gleichförmig auf dem 53-Bit-Raster des Einheitsquadrats. Varianten erhalten bei gleichem (n, r) identische Punkte. Unterschiedliche Größen verwenden verschiedene Seeds; ihre Punktmenngen sind nicht ineinander geschachtelt.

2.2 Zeitmessung und Validierung

Die primäre Größe ist die von `build_profiled` erfasste Aufbauzeit: Eingabenormalisierung, Integerpackung, Bucketbildung, Präfixarbeit, Konformitätsabschluss, Transfers, Merge und Ergebnismengen. Zufallsziehung, Poolvorbereitung, nachträgliche Validierung, Payload-Schätzung, Sortierung und Hashbildung gehören nicht dazu. Die komplette überwachte Versuchsdauer wird separat gespeichert. Prozessstartkosten sind somit nicht verschwunden, sondern bewusst getrennt.

Die disjunkten Hauptprozessphasen summieren sich zur Aufbauzeit. Arbeiterintervalle überlappen mit ihnen und dürfen nicht hinzuaddiert werden. Die Diagnosefassung hält zusätzliche Daten vor; die Untersuchung misst deren Skalierung, nicht die einer hypothetischen speicheroptimierten Fassung. Der unveränderte instrumentierte Code ist unter `data/source/` archiviert.

Jeder erfolgreich abgeschlossene Versuch prüft die Konformität, $|N| = 2|L| - 1$ und die exakte Flächensumme

$$\sum_d h_d 2^{1-d} = 2,$$

wobei h_d die Blattzahl auf Tiefe d ist. Koordinaten- und Blatthashes sowie Tiefenhistogramme und Teilungszahlen werden zwischen den Varianten verglichen. Die 10^5 -Stufe wird zusätzlich gegen U1 geprüft. Für größere Stufen liegt keine unabhängige rationale Großmessung vor; dort ist der Vergleich ein Konsistenztest zweier Varianten mit gemeinsamer geometrischer Implementierung.

2.3 Speicherbudget und Abbruchregel

Die Maschine verfügt über ungefähr 16 GB RAM. Verwendet werden wie im Hauptbericht der Intel Core Ultra 7 255U (12 physische Kerne, 14 logische CPUs), Linux und Python 3.13.9. Die Überwachung verwendet `psutil` 7.0.0. Vor der Serie wird ein Budget von höchstens 6 GiB, zusätzlich begrenzt durch den anfänglich verfügbaren Speicher abzüglich 2 GiB, festgelegt. Die genauen Bytewerte und Systemangaben stehen in den Metadaten. Alle 0,1 Sekunden wird die Summe der RSS-Werte des Messprozesses und seiner Nachkommen abgefragt. Übersteigt sie das Budget, wird die Versuchsprozessgruppe beendet. Dasselbe gilt bei weniger als 2 GiB systemweit verfügbarem Speicher oder nach 1800 Sekunden. Ein zusätzliches Adressraumlimit in Budgethöhe gilt pro Prozess und wird an die Arbeiter vererbt.

RSS-Summen können gemeinsam gemappte Seiten mehrfach zählen und die Abtastung kann kurze Spitzen verpassen. Das Adressraumlimit misst zudem virtuellen und nicht residenten Speicher. Die Schutzmechanismen sind konservative praktische Grenzen, keine präzise Messung des minimal benötigten physischen Speichers. Der Monitor verursacht kleinen zusätzlichen Aufwand außerhalb der Messprozesse.

Nach dem ersten Ressourcenabbruch werden die übrigen Wiederholungen *dieser Größenstufe und Variante* ausgelassen; die andere Variante wird separat versucht. Ein abgebrochener Versuch ist keine Messung der Zeit eines fertig aufgebauten Baums. Seine verstrichene Zeit wird ausdrücklich nicht in Laufzeitmittelwerte aufgenommen. Ein Lauf kann auch erst in der Validierung oder Kennzahlberechnung scheitern; ohne vollständiges Ergebnis wird er ebenfalls nicht als Erfolg ausgewertet.

3 Ergebnisse

3.1 Vollständigkeit und Korrektheit

Es liegen 12 vollständig abgeschlossene Versuche vor. 2 Versuche wurden abgebrochen; 0 Wiederholungen wurden gemäß Ressourcenregel ausdrücklich als ausgelassen protokolliert. 4 weitere geplante Messungen wurden wegen des Abbruchs der Gesamtserie nicht mehr gestartet. Alle 6 verfügbaren seriell-parallelen Paare stimmen exakt in den Codesignaturen und Strukturkennzahlen überein. Die historische Referenzprüfung der kleinen Stufe wurde vom Controller durchgeführt.

3.2 Aufbauzeit, Durchsatz und Baumgröße

n	Variante	r	Zeit [s]	$[\mu\text{s}/\text{Punkt}]$	$[10^3 \text{ Punkte/s}]$	RSS [GiB]
100000	parallel (4)	3	$2,840 \pm 0,059$	28,40	35,21	0,44
100000	seriell	3	$3,918 \pm 0,287$	39,18	25,52	0,24
1000000	parallel (4)	3	$29,250 \pm 0,373$	29,25	34,19	2,68
1000000	seriell	3	$44,652 \pm 1,149$	44,65	22,40	1,84
10000000	parallel (4)	0	—	—	—	—
10000000	seriell	0	—	—	—	—

Zeit: Mittelwert und Stichprobenstandardabweichung; RSS: größter abgetasteter Wert über die erfolgreichen Gesamtversuche einer Konfiguration, nicht nur während des Aufbaus. Fehlende Zeiten bedeuten kein vollständiges Ergebnis, nicht null Sekunden.

n	mittlere Blätter	mittlere Knoten	Bereich $d_{\max}$
100000	331712,0	663423,0	32–36
1000000	3297135,0	6594269,0	39–42

3.3 Gemessene Skalierungsfaktoren

Variante	Größenpaar	R	α
seriell	100000 $\rightarrow$ 1000000	11,396	1,057
parallel (4)	100000 $\rightarrow$ 1000000	10,298	1,013

Bei $n = 100000$ beträgt das Verhältnis serieller zu paralleler Mittelzeit 1,380.

Bei $n = 1000000$ beträgt das Verhältnis serieller zu paralleler Mittelzeit 1,527.

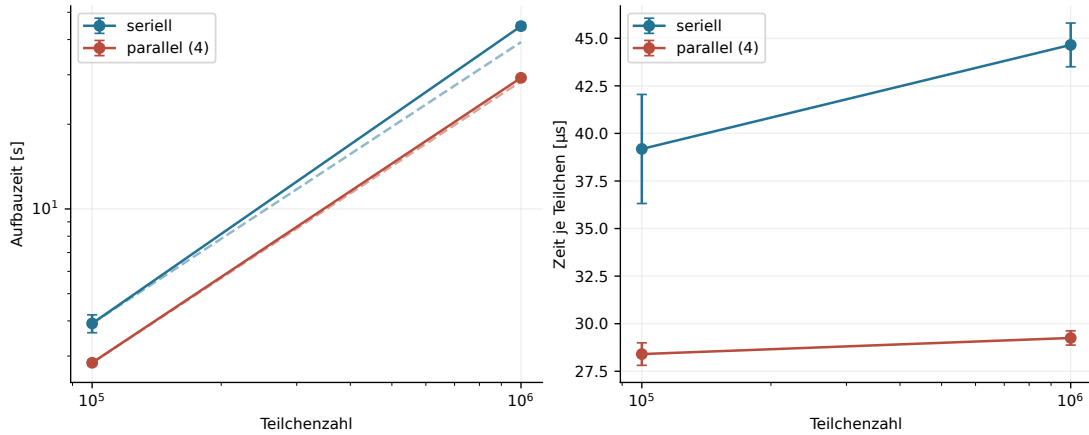


Abbildung 1: Nur vollständige Versuche: mittlere Aufbauzeit und Zeit je Teilchen. Fehlerbalken: Stichprobenstandardabweichung. Gestrichelt: lineare Fortsetzung ab der kleinsten gemessenen Größe als Orientierung, nicht als Messung.

3.4 Aufteilung der Aufbauzeit

n	Variante	Norm.	Pack	Bucket	Abschluss	Merge	Rest
100000	parallel (4)	1,573	0,201	0,277	0,000	0,141	0,647
100000	seriell	1,515	0,189	0,000	1,244	0,011	0,959
1000000	parallel (4)	16,808	1,160	3,160	0,000	1,367	6,755
1000000	seriell	16,586	1,138	0,000	12,648	0,078	14,202

Alle Phasenwerte sind Mittelzeiten in Sekunden. Rest enthält insbesondere das Warten auf Arbeiter beziehungsweise die serielle Jobausführung. Ein nahezu verschwindender zentraler Abschluss in der parallelen Variante bedeutet, dass diese Arbeit bereits in den Jobs erledigt wird.

3.5 Ressourcenabbrüche und ungemessene Stufen

n	Variante	r	Abbruchgrund	Dauer [s]	RSS [GiB]
10000000	parallel (4)	1	Programmfehler	195,6	5,89
10000000	seriell	1	Adressraumlimit	195,0	5,89

Diese Dauern sind ausschließlich Zeiten bis zum Abbruch. Sie dürfen weder als Aufbauzeiten noch als gesicherte Untergrenzen einer unbeschränkten Ausführung interpretiert werden. Aus einem Adressraumlimit folgt nicht, dass die Maschine physisch exakt bei diesem RSS-Wert erschöpft wäre. Weitere Wiederholungen derselben gescheiterten Konfiguration wurden nicht erneut gestartet.

In der Variante seriell trat der Speicherfehler beim Packen der exakten Koordinaten in Integertripel auf, also vor der Präfixarbeit und dem Konformitätsabschluss. Das Fehlerprotokoll enthält den vollständigen Aufrufpfad.

Die parallele Variante endete ebenfalls beim Aufruf `_pack(exact)`, jedoch mit `SystemError: error return without exception set`. Die Nähe zum Adressraumlimit legt einen Zusammenhang mit Speicherdruck nahe, beweist ihn aber nicht. Dieser Fehler bleibt als ungeklärter Laufzeitfehler klassifiziert. Der Controller unterbrach deshalb die Gesamtserie; der originale Status und die Protokolle wurden nicht nachträglich als erfolgreicher Abschluss umgeschrieben.

Für $n = 10000000$ konnte unter dem festgelegten Budget kein vollständig validierter Baum fertiggestellt werden. Die Laufzeit an dieser Größenstufe und der zugehörige Skalierungsfaktor bleiben offen.

4 Interpretation der Skalierung

Für zwei erfolgreich gemessene Größen n_1, n_2 werden das Verhältnis der Mittelzeiten und der empirische Exponent

$$R = \frac{\bar{t}(n_2)}{\bar{t}(n_1)}, \quad \alpha = \frac{\log R}{\log(n_2/n_1)}$$

angegeben. Bei einer Verzehnfachung wäre $R = 10$ beziehungsweise $\alpha = 1$ exakt linear. Der Durchschnitt ist $n/\bar{t}(n)$; die Zeit je Teilchen ist $\bar{t}(n)/n$. Dies sind deskriptive Größen, kein Nachweis einer asymptotischen Komplexität. Insbesondere können nahe Teilchenpaare, Integer-Bitlängen, Hashmengen, Speicherallokation und Cacheeffekte die Skalierung verändern.

Die Stichprobenstandardabweichung beschreibt nur die drei gemessenen Wiederholungen. Prozessaffinität, Takt, Temperatur und Hintergrundlast sind nicht kontrolliert. Mit größerer Punktmenge wächst auch die serielle Normalisierung; Parallelität garantiert daher keinen konstanten Beschleunigungsfaktor. Falls die größte Stufe am Ressourcenbudget scheitert, bleibt ihre Laufzeitskalierung unbestimmt. Ein solcher Befund begründet eine Speicheroptimierung oder Messung auf einer größeren Maschine, aber keine erfundene Zeitkurve.

Bei der aktuellen Diagnosefassung existieren während der Vorbereitung mehrere Darstellungen derselben Eingabe gleichzeitig: ursprüngliche Floatpaare, exakte rationale Koordinaten und gepackte Integerdaten. Später kommen Teilungsmengen, globale Knotenmengen und zwischengespeicherte Jobresultate hinzu. Eine zukünftige Speicheroptimierung sollte die Lebensdauer dieser Darstellungen verkürzen und kompaktere Datenstrukturen prüfen. Sie wäre als eigene Algorithmusversion mit erneuten Gleichheitsprüfungen und neuen Messungen zu behandeln, nicht nachträglich mit U4 zu vermischen.

5 Archiv und Reproduktion

Die archivierte Quellkopie besteht nach Abschluss der Messungen alle 45 Tests einschließlich der neuen Tests für Speicherwächter und Histogramm-Invarianten. Ein vorgeschalteter kleiner End-to-End-Test mit 100 Punkten prüfte beide Ausführungsvarianten. Die Unit-Tests ersetzen keine Aufklärung des unter dem großen Speicherlimit aufgetretenen `SystemError`; dieser bleibt offen. Der abschließende Testlauf liegt in `test_results.txt`.

`data/trials/` enthält vollständige JSON-Ergebnisse, `data/attempts/` zusätzlich Abbrüche und ausgelassene Wiederholungen, `data/logs/` die Fehlerprotokolle. Seeds, Quellen, Grenzen und Umgebung stehen in `data/metadata.json`; Phasen und Jobintervalle sind in jedem erfolgreichen Einzelprotokoll enthalten. `manifest.json` sichert das archivierte Datenverzeichnis mit SHA-256. `build_addendum.py` erzeugt aus diesem Archiv Tabellen, Grafiken und den Ergebnistext, startet aber keine Experimente.

Zum erneuten Setzen werden im Addendum-Verzeichnis ausgeführt:

```
MPLCONFIGDIR=/tmp/cfk-matplotlib python3 build_addendum.py
latexmk -pdf -interaction=nonstopmode -halt-on-error Addendum.tex
```

Eine neue Serie mit denselben Quellen kann in ein *neues* Verzeichnis geschrieben werden:

```
python3 data/source/benchmark_large.py \  
--output /tmp/cfk-u4-replay \  
--sizes 100000 1000000 10000000 \  
--repetitions 3 --budget-gib 6
```

Für den zusätzlichen historischen Vergleich beim Neuanlegen eines solchen Laufs muss die U1-Referenz wieder bereitgestellt werden; die deterministische Seedbildung selbst hängt nicht von ihr ab. Der Originaldatensatz dieses Addendums enthält die verwendeten Referenzprotokolle. Die exakten Zahlen und der Status sind auch ohne PDF aus den JSON- und CSV-Dateien rekonstruierbar.