CFK-Simulationen auf drei Linux-Rechnern

Messreihe MACHINES_G1, 17. September 2026

1 Fragestellung und Versuchsaufbau

Wir untersuchen die Auslagerung bestehender Simulationen und die relative Rechenleistung. Staubphysik und variable Teilchenzahlen sind nicht Gegenstand dieser Messung. Der Bericht verwendet die Grundgestaltung der bereitgestellten Vorlage, ohne deren Aufgaben- und Lösungsteil.

Auf `dreihemden-2` wurden zunächst der eingefrorene C++-Quellstand kompiliert und die drei nativen Tests erfolgreich ausgeführt. Alle elf kleinen Portabilitätsläufe mit 2048 Teilchen auf den drei Rechnern lieferten bytegleiche Endzustände und gespeicherte Trajektorien, trotz unterschiedlicher Compiler.

Für den Leistungstest verwenden alle Rechner *dieselbe* portable GCC-11.4-Programmdatei: C++20, `-O3 -DNDEBUG -ffp-contract=off -fopenmp`, Boost 1.83, ohne Fast-Math oder CPU-spezifisches `march=native`. Es wurden keine Systempakete installiert.

Die gemeinsame Eingabe enthält $n = 10^5$ gleich schwere Teilchen mit Gesamtmasse 1, gleichverteilten Positionen in $[0.2, 0.8)^2$ und Geschwindigkeitskomponenten in $[-0.002, 0.002)$. Seed: 2026091709. Die gespeicherte Datei ist die Referenz. Wir rechnen fünf KDK-Schritte mit $\Delta t = 10^{-4}$, $\epsilon = 0.01$ und $\theta = 0.4$, klassischem Öffnungskriterium und Topologiewiederverwendung. Das ergibt sechs Kraftauswertungen. Ausgaben erfolgen nur am Anfang und Ende.

Je Maschine wird ein voller Aufwärmlauf ausgeschlossen. Danach folgen drei Wiederholungen je Threadzahl in rotierender Reihenfolge. Gitter und Kraft nutzen jeweils T Threads, die Vorbereitung $\min(T, 4)$. Es gibt keine explizite Kernbindung. Jeder Rechner führt nur einen Testlauf gleichzeitig aus.

2 Ergebnis und erste Einsatzempfehlung

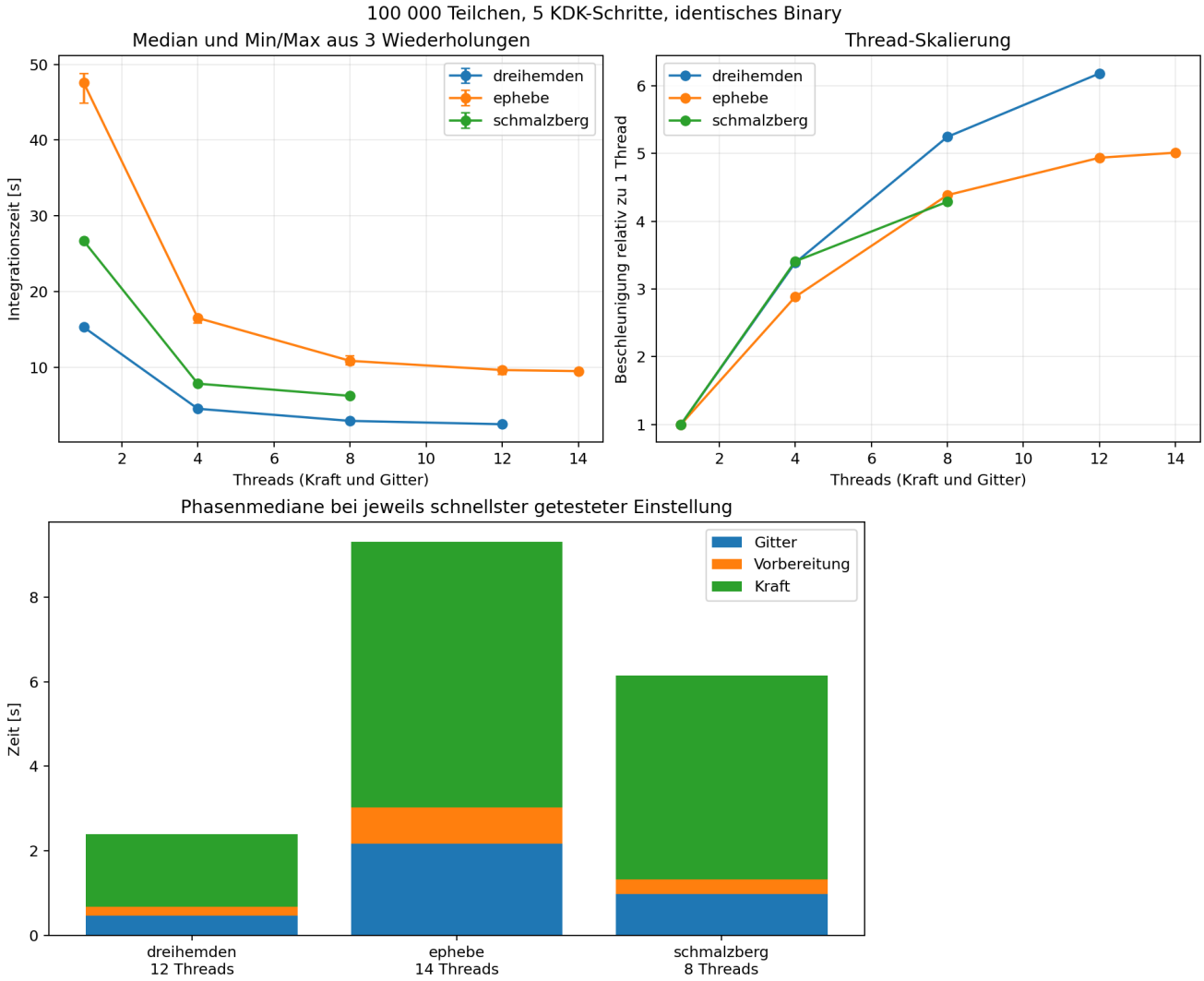
Rechner	RAM ca.	Threads	Median [s]	Min–Max [s]
dreihemden-2	32 GB	12	2.473	2.448–2.478
schmalzberg	16 GB	8	6.221	6.209–6.424
ephebe (lokal)	16 GB	14	9.487	9.248–9.543

Die Tabelle zeigt die jeweils schnellste getestete Einstellung. Die Zeiten umfassen die Integration samt Ausgabecallbacks, nicht das vollständige Starten, Einlesen und abschliessende Schreiben. Die entsprechenden Prozessmediane sind 2.624, 6.427 und 9.873 Sekunden. Der maximale residente Speicherbedarf lag bei diesen Einstellungen bei rund 83.4, 75.9 und 82.9 MiB.

dreihemden-2 (Ryzen 5 7600, 6 Kerne/12 Hardware-Threads) ist die erste Wahl für lange Rechenläufe. **schmalzberg** (i7-8565U, 4 Kerne/8 Threads) eignet sich als zweiter Worker. **ephebe** (Core Ultra 7 255U, 12 Kerne/14 Threads) bleibt gut für Entwicklung und Kontrollläufe; 12 Threads sind ein sinnvoller Ausgangspunkt, da 14 nur etwa 1.5 % Medianvorteil zeigen. Das belegt keine Ursache für die Unterschiede: CPU-Typ, Energieverwaltung, Laufzeitbibliotheken und Hintergrundlast wurden nicht getrennt untersucht.

Rohdaten, Befehle, Hardwareinformationen und Prüfsummen liegen im selben Verzeichnis. Tabellen: `summary.csv`; Prüfung: `validation.json`. Die eingefrorenen Skripte `machine_benchmark.py` und `analyze_machine_benchmark.py` sowie `README.md` dokumentieren Wiederholung und Auswertung.

3 Skalierung und Zeitanteile



4 Reproduzierbarkeit und Grenzen

Alle 36 gemessenen Läufe und drei Aufwärmläufe liefern bytegleiche Endzustände und gespeicherte Trajektorien. Die Prüfung erfolgte nach der Rückübertragung erneut aus den Dateien. Das ist ein positives Testergebnis, keine allgemeine Garantie für lange chaotische Entwicklungen.

Die Beschleunigung ist $S(T) = \text{median}(t_1)/\,\text{median}(t_T)$. Phasenbalken zeigen einzeln berechnete Mediane, deren Summe nicht exakt dem Median der Gesamtzeit entsprechen muss. Die drei Wiederholungen erfassen Zeitstreuung, nicht verschiedene physikalische Anfangszustände.

Empfehlung: unabhängige Simulationen über eine gemeinsame Warteschlange verteilen; jeder freie Rechner holt den nächsten Lauf. Keine starre Drittelung der Arbeit und keine Netzwerkverteilung einzelner Kraftaufrufe. Die Studie misst nur eine uniforme Verteilung und eine Teilchenzahl. Millionenfälle, starke Verklumpung, Dauerlast und mehrere gleichzeitige Jobs brauchen eigene Pilotmessungen; aus dem kleinen Speicherverbrauch darf keine maximale tragbare Teilchenzahl abgeleitet werden.