

Abstract

The openPMD API is a DSL used in plasma physical simulations for the description of particle-mesh data according to the openPMD standard. The API offers a number of flexibly interchangeable efficient implementations in terms of IO backends. Having initially been designed with file-based IO in mind, we intend to overcome filesystem bottlenecks by enabling a streaming-aware description of particle-mesh data. The streaming backend is provided by the ADIOS2 framework, currently in development at Oak Ridge National Laboratory.

Extended this way, we use the openPMD API to loosely couple a simulation computed by PIconGPU – a hardware accelerated implementation of the particle-in-cell algorithm developed at Helmholtz-Zentrum Dresden-Rossendorf – with an analysis computed by GAPD, which is a hardware-accelerated atom-based polychromatic diffraction simulation.

In loose coupling, as opposed to tight coupling, two (or more) applications are executed in separate, but cooperate by exchanging data. This way, we want to make use of heterogeneous hardware more flexibly and allow for standalone codes instead of plugins, using a unified streaming-aware API for particle-mesh data and leveraging high-speed communication infrastructure available in modern compute clusters.

We put the coupled simulation to testing on the Taurus compute cluster located at TU Dresden, comparing several execution strategies. When communicating between codes via RDMA networks implemented through Infiniband, we measure that the time performance by data dumps settles between that of filesystem IO and that of raw simulation performance. Scaling behavior depends on the chosen execution strategy.

Kurzfassung

Die openPMD API ist eine DSL, die in plasmaphysikalischen Simulationen für die Beschreibung von Partikel-Mesh-Daten nach dem openPMD-Standard verwendet wird. Die API bietet mehrere effiziente Implementierungen durch IO-Backends, welche nach Bedarf flexibel angewählt und ausgetauscht werden können. Nachdem die ursprüngliche Gestaltung der API ihren Fokus auf dateisystembasiertes IO legte, zielt diese Arbeit darauf ab, im Dateisystem entstehende Flaschenhälse durch eine Beschreibung von Partikel-Mesh-Daten mit Streaming-Semantik zu umgehen. Das Backend hierfür wird vom ADIOS2-Framework bereitgestellt, welches derzeit am Oak Ridge National Laboratory entwickelt wird.

Die hiermit erweiterte openPMD API nutzen wir, um eine Simulation, welche durch PIconGPU berechnet wird – einer hardwarebeschleunigten Implementierung des Particle-in-Cell-Algorithmus – lose mit einer durch GAPD berechneten Analy-

se zu koppeln, welches eine hardwarebeschleunigte atom-basierte polychromatische Beugungssimulation berechnet.

Zwei oder mehr lose gekoppelte Applikationen werden im Gegensatz zu eng gekoppelten Applikationen separat ausgeführt und kooperieren miteinander durch Datenaustausch. Das erlaubt eine flexiblere Ausnutzung heterogener Hardware ebenso wie die Nutzung selbstständiger Codes im Gegensatz zu Plugins. Wir erreichen dies durch eine API mit Streaming-Semantik für Partikel-Mesh-Daten und durch Ausnutzung von Hochgeschwindigkeitskommunikationsinfrastruktur, welche auf modernen Rechenclustern verfügbar ist.

Wir führen Tests mit mehreren Ausführungsstrategien auf dem an der TU Dresden befindlichen Rechencluster Taurus durch. Bei Kommunikation zwischen Codes über RDMA-Netze, die durch Infiniband implementiert sind, messen wir eine Zeitperformanz, welche sich zwischen der Performanz von Dateisystem-IO und derjenigen der eigentlichen Simulation ohne IO ansiedelt. Das Skalierungsverhalten hängt stark von der gewählten Ausführungsstrategie ab.