Data flow programming—a high performance and highly complicated programming concept?


Jens Breitbart

Computer Architecture
Technische Universität München

j.breitbart@tum.de
Warning

• This talk is not about Chapel … at least not directly.

• It is about a library called GASPI and I was part of the group developing it.

• It is my opinion.

• It is more like a user experience report.
So, why should you care?

- GASPI is a PGAS implementation as a library.

- GASPI's notification mechanism is similar to sync variables.

- But we promote mostly a single different programming style:
  - a state machine that reacts based on notifications

- Provides great performance.
What we have learned (so far) talking to the designer

• Performance is a great conversation starter.

• The dataflow / notification concepts is easy to understand

• and it exposes a lot of parallelism.

• Promote a migration path (GASPI is MPI compatible).
What we have learned (so far) talking to the developer

• Performance is a great conversation starter.

• The dataflow / notification concepts is easy to understand,
  • but how do I implement a high performance multi threaded state machine again?

• and it exposes a lot of parallelism
  • which one to use?

• Promote a migration path (GASPI is MPI compatible),
  • but it requires rewriting large parts of my existing application.
Final words

- We have met people who like to try something that is not MPI.
- And people who have been looking for something that is simpler than what we offered.