



FastFlow, MTCL, CAPIO: Building Blocks for Parallel and Distributed Computing

Flexible and Scalable Ecosystem targeting from HPC to Edge Platforms

The Evolving Landscape of Computing

- Rapid transition from traditional HPC to Cloud + Edge Computing paradigm
- Increasing complexity in managing heterogeneous systems, communication protocols, and programming abstractions
- Hybrid programming is challenging and time consuming
 - No single approach suitable for all contexts
- Need for flexible adaptable and efficient components to harness the power of heterogeneous distributed computing
- Over the years, we designed and developed three main components:



MTCL

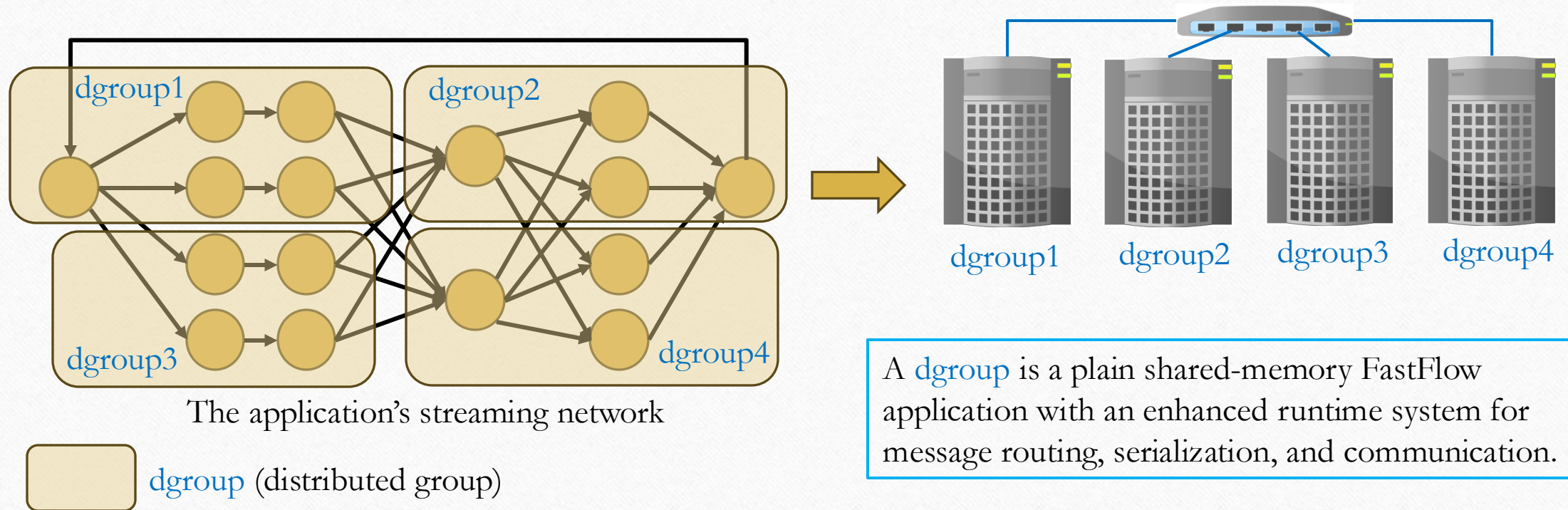


CAPIO



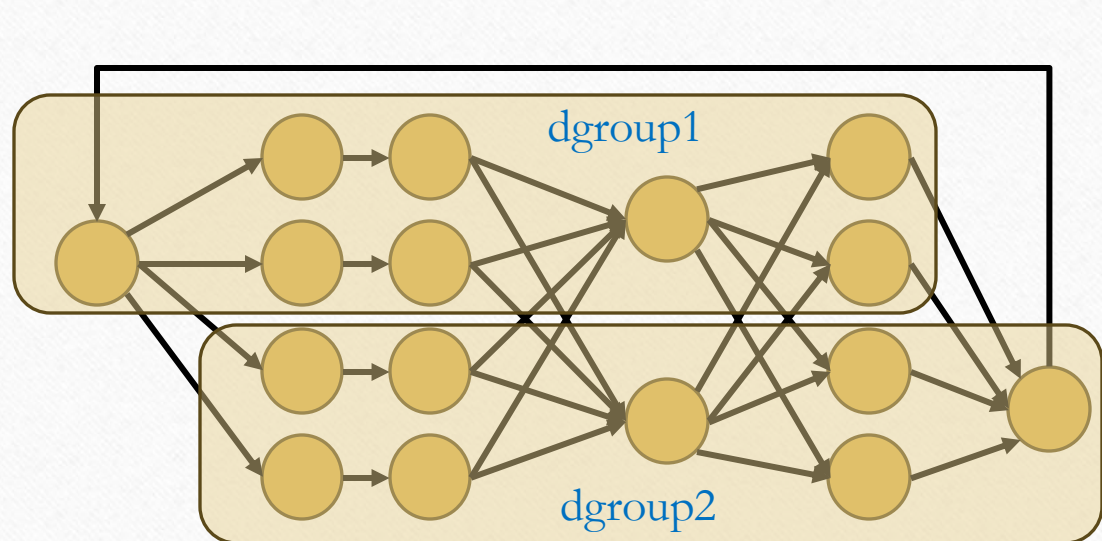
FastFlow – Stream Parallel Programming

- Header-only C++ library, promoting data-flow streaming as a first class concept to design parallel and distributed applications

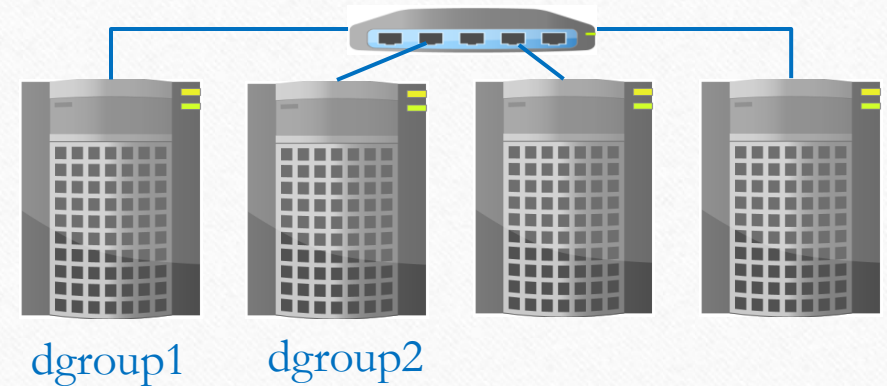


FastFlow – Stream Parallel Programming

- Header-only C++ library. It promotes data-flow streaming as a first class concept to design parallel and distributed applications



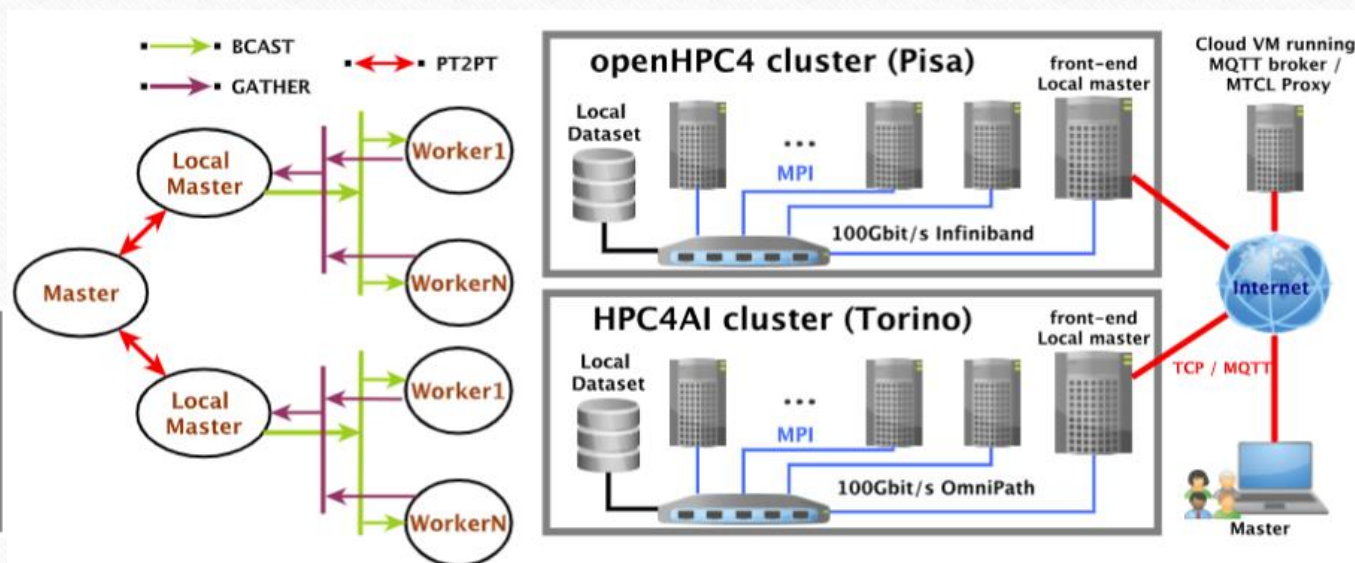
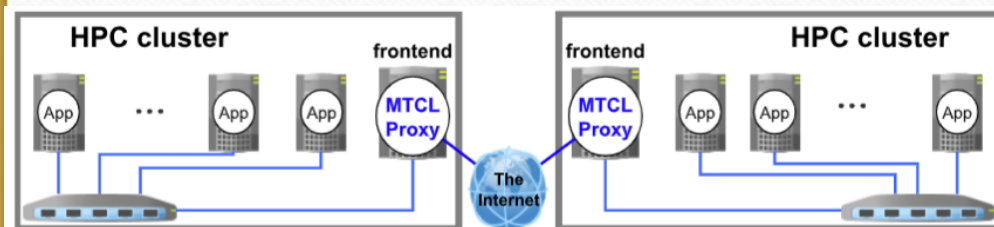
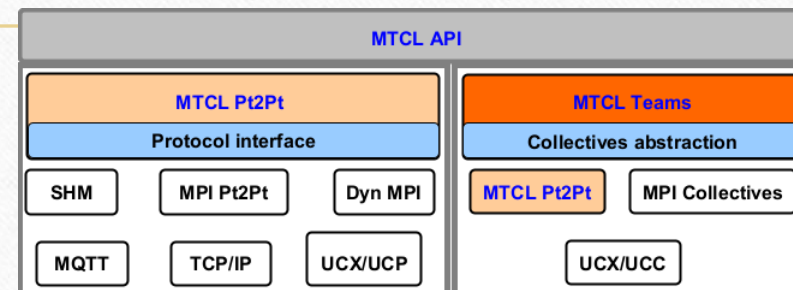
The application's streaming network



A **dgroup** is a plain shared-memory FastFlow application with an enhanced runtime system for message routing, serialization, and communication.

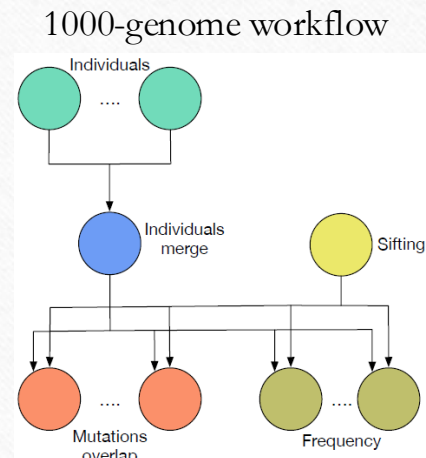
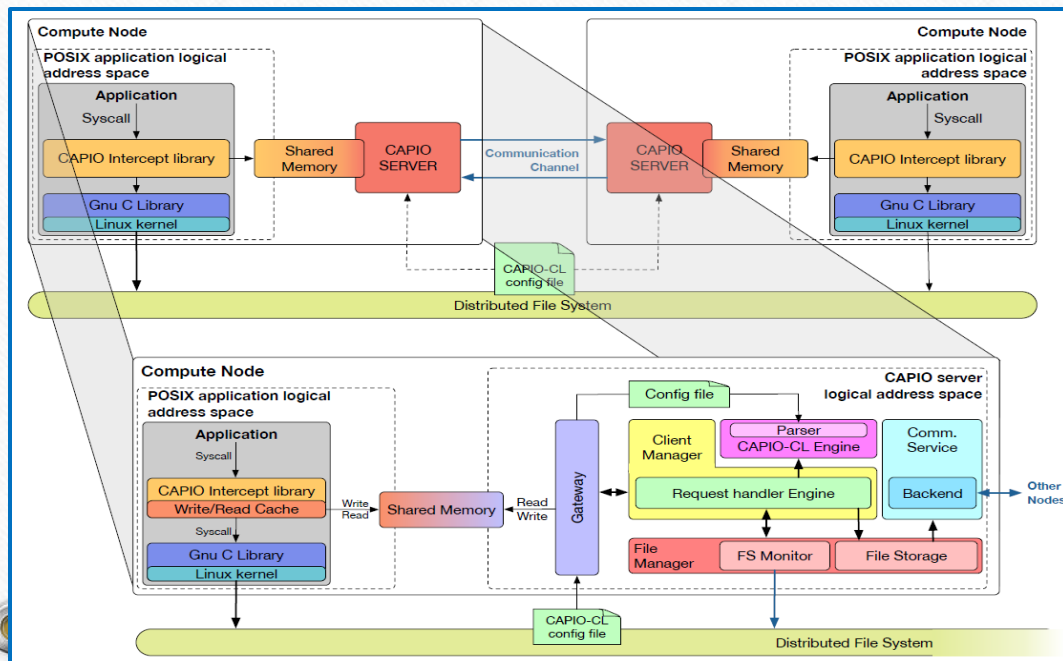
MTCL – Versatile Communication Library

- **Multi-transport** (connection-oriented) communication library
 - Transports/Protocols: MPI, UCX, TCP, MQTT, ...
 - Supports blocking/non-blocking point-to-point, and collectives communications
- **MTCL Proxy** to transparently switch protocols across domains



CAPIO – Transparent I/O Streaming

- JSON-based I/O coordination language + Distributed Middleware
- Transparently intercepts POSIX I/O syscalls and enables data streaming between workflow components, driven by simple language rules



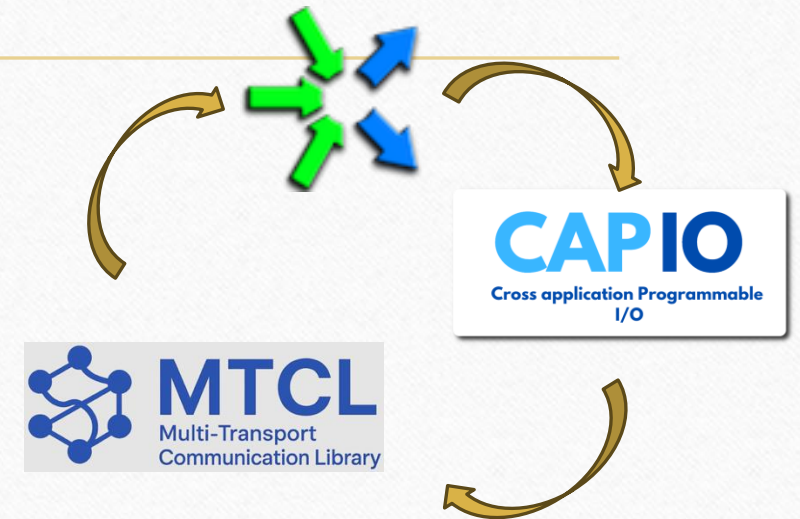
```
{
  "name": "1000_genome",
  "IO_Graph": [{
    "name": "individuals",
    "streaming": [{
      "name": ["capio_dir/chr1n-1-250000/*"],
      "committed": "on_close",
      "mode": "no_update"
    }]
  }],
  "name": "individuals_merge",
  "streaming": [{
    "dirname": ["capio_dir/chr1n/*"],
    "committed": "on_close",
    "n_files": 2504
  }]
}, {
  "name": "sifting",
  "streaming": [{
    "name": ["capio_dir/sifted.SIFT.chr1.txt"],
    "committed": "on_close",
    "mode": "no_update"
  }]
}, {
  "name": "mutation_overlap",
  "name": "frequency"
}
}
```

CAPIO-CL config file

Aim: Integrated Ecosystem

- Integration

- FastFlow** provides the parallel programming models and needed mechanisms to design complex workflows
- MTCL** ensures flexible and efficient communication across diverse protocols
- CAPIO** enhances I/O performance and enables the integration of black-box components in the application workflow



- Deployment Scenarios

- HPC + Cloud + Edge** computing environments
- Support for scientific simulations (HPC), Real-Time data processing and Federated Learning (Cloud + Edge)

FastFlow: <https://github.com/fastflow/fastflow>

MTCL: <https://github.com/ParaGroup/MTCL>

CAPIO: <https://github.com/High-Performance-IO/capio>

Research Opportunities

- **Integration** in a coherent distributed ecosystem is challenging. Research challenges:
 - **System software engineering** (efficiency, flexibility, easy-of-use)
 - **Performance optimization** across heterogeneous platforms and application domains
 - **Deployment, Orchestration and Coordination** in HPC-Cloud-Edge infrastructure
- Defining **Domain Specific Languages (DSLs)** on top of the distributed ecosystems
 - Real-time data streaming (through **WindFlow** API) 
 - Federated Learning environment (e.g., integration with **Apache TVM**, **LibTorch**)
 - ... other application domains?

Feel free to get in touch if you're interested!