



Structured Data Streams: Peta-scale I/O and Storage



Collaborative Research

**CERCS Research Center
College of Computing
Georgia Institute of Technology**

**Computer Science Department
University of New Mexico**

Karsten Schwan
Arthur Maccabe
Patrick Bridges
Greg Eisenhauer
Patrick Widener
Matthew Wolf

and

(Ron A. Oldfield, Sandia National Laboratories)

(Scott Klasky, Oak Ridge National Laboratories)



Remote
Sensors

Storage
Engine

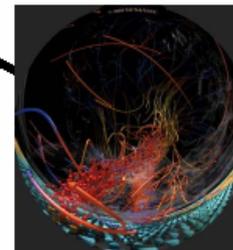
Local MPP
(Compute and
I/O Nodes)



Remote
Clients



Visualization



Initial
Project
Focus

Problem: Large-scale Data Movement to/from MPP

- **Many petascale-headed codes have I/O needs at the order of TBs/hr**
 - **GTC (gyrokinetic fusion code) (ORNL/Jaguar)**
multiple, simultaneous data feeds: e.g., diagnostics, analysis, restart controlled data drain to limit perturbation
 - **Chimera (supernova simulation) (ORNL/Jaguar)**
 - **35,000 cores, 550kb/core/sec => ~18 GB/sec**
or: **TBs/hour, for 6 days (!)**



Solution: Structured Data Streams



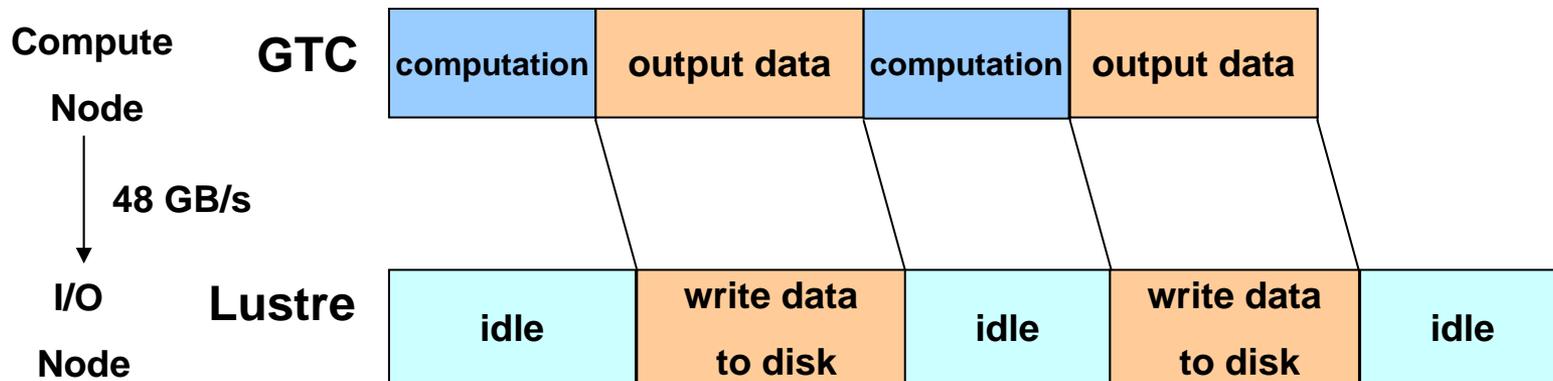
Future high end applications moving Terabytes/output cycle make it necessary to manipulate data 'in transit'.

'Structured data streams' is an innovative approach to I/O, performing runtime data annotation to enable efficient data manipulation, synchronously and asynchronously with data movement.

Technical Elements

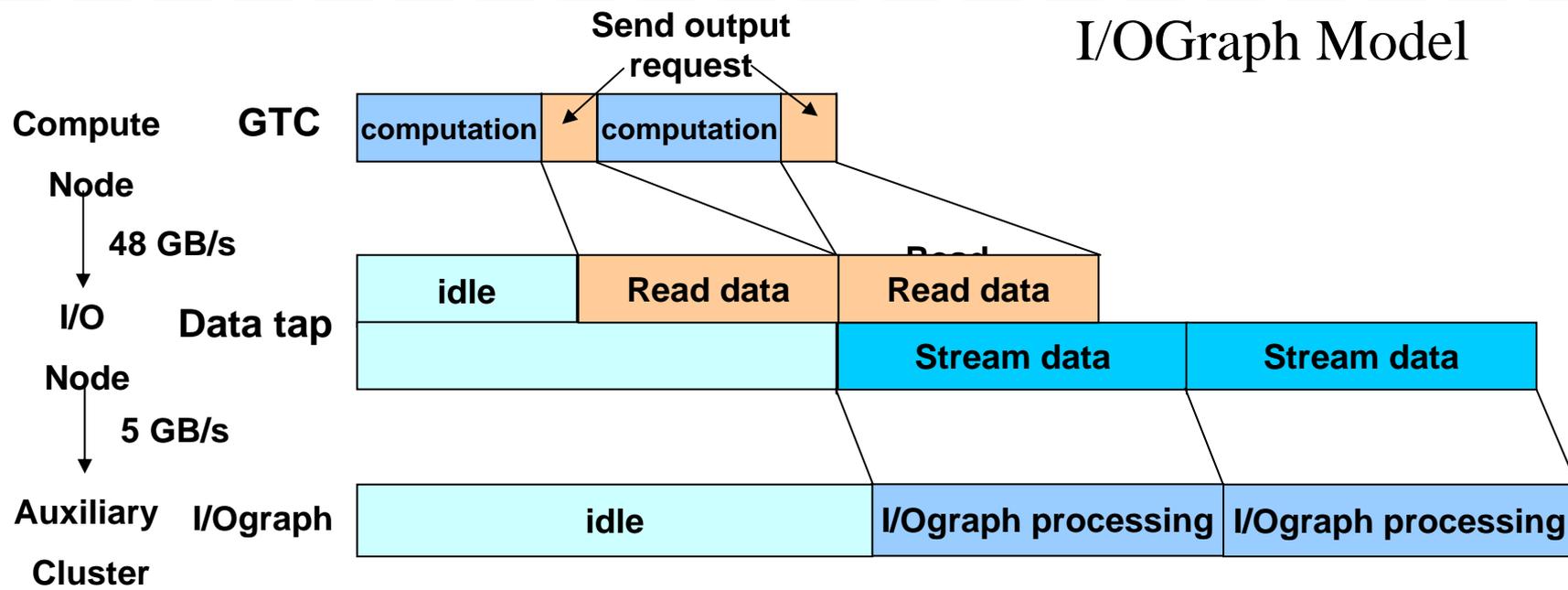
- **Data tap:** asynchronous structured data capture, uniform API
- **I/O graph:** graph-based, scheduled data buffering, forwarding, and synch. or asynch. data manipulation
- **Metabots:** asynchronous data and metadata processing

Model execution cartoons

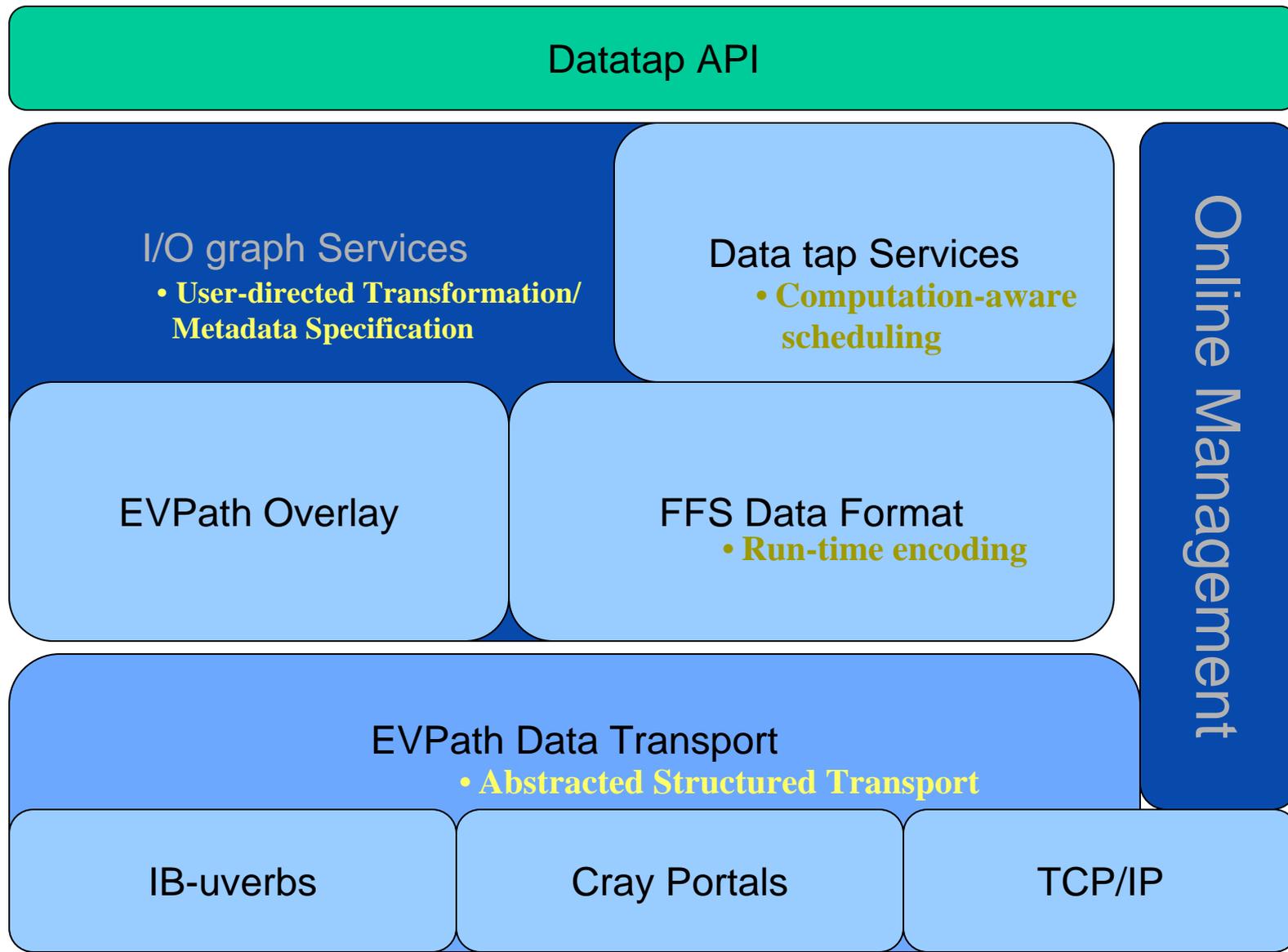


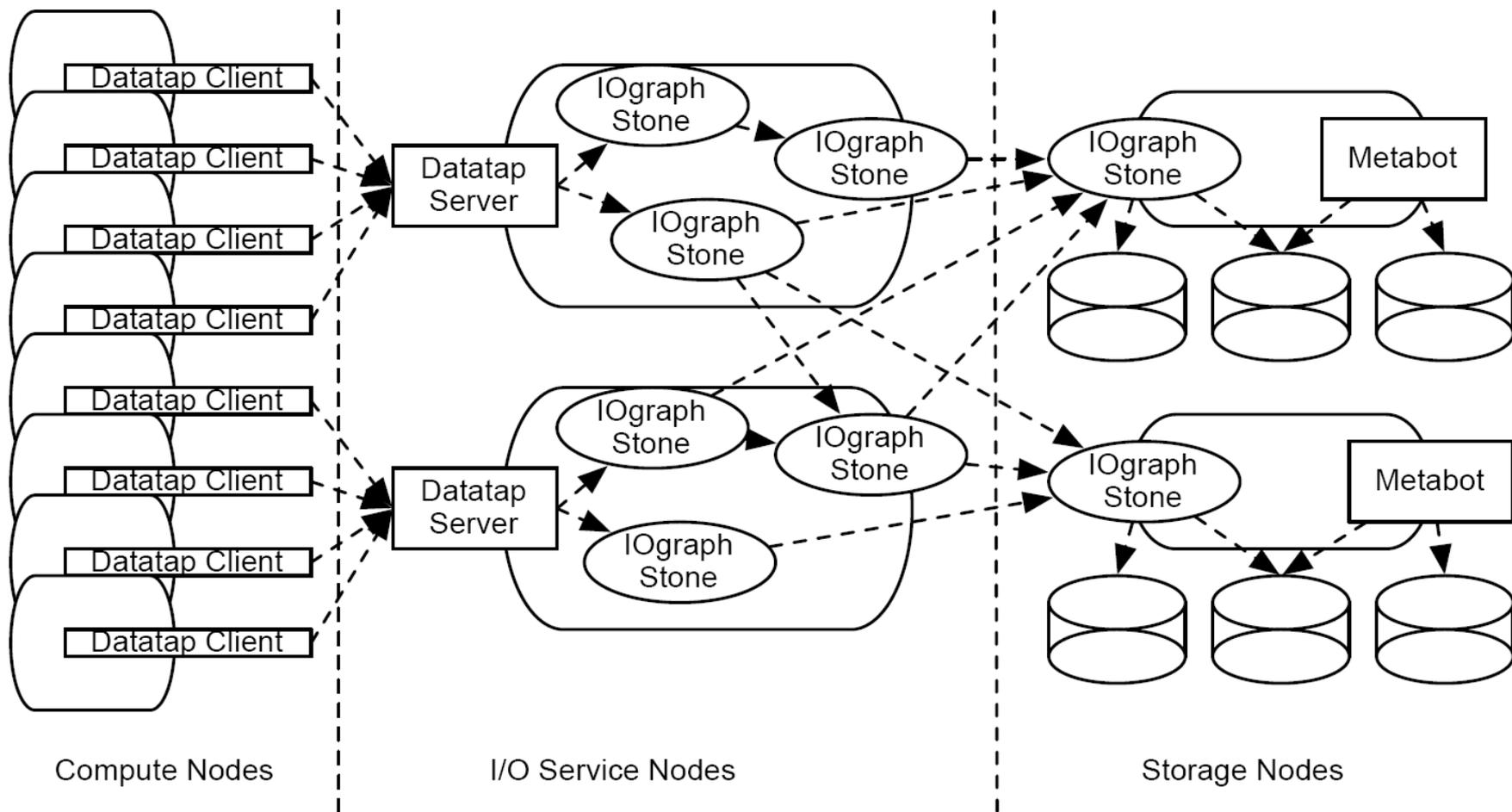
Synchronous Model

I/OGraph Model

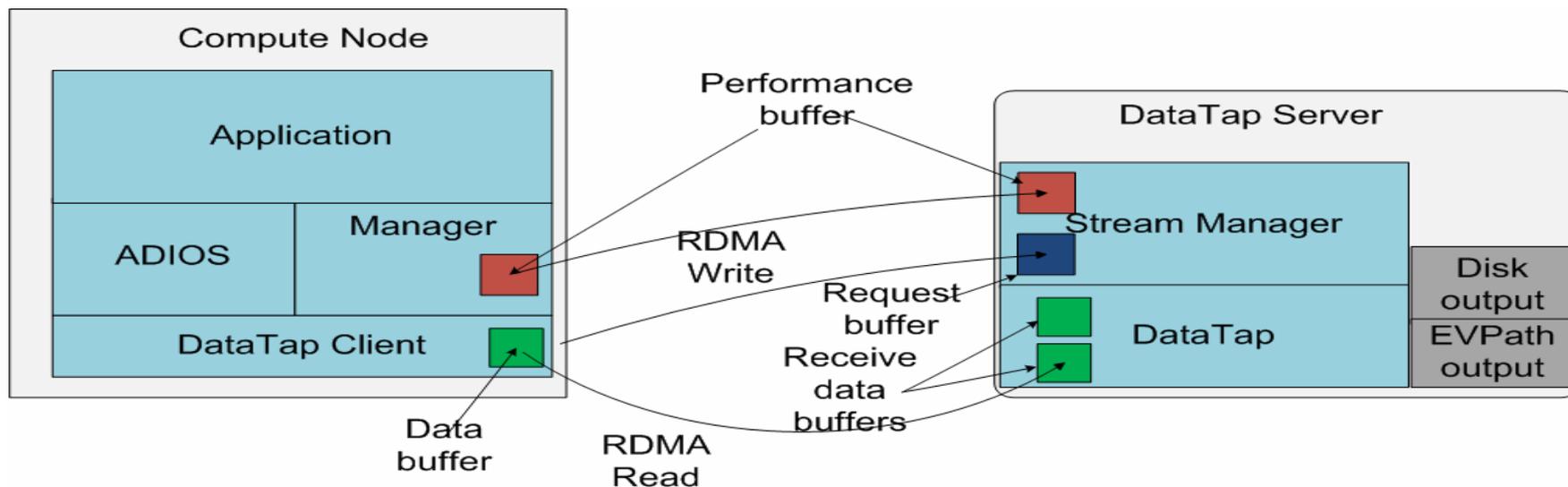


Structured Streams: Architecture and Implementation

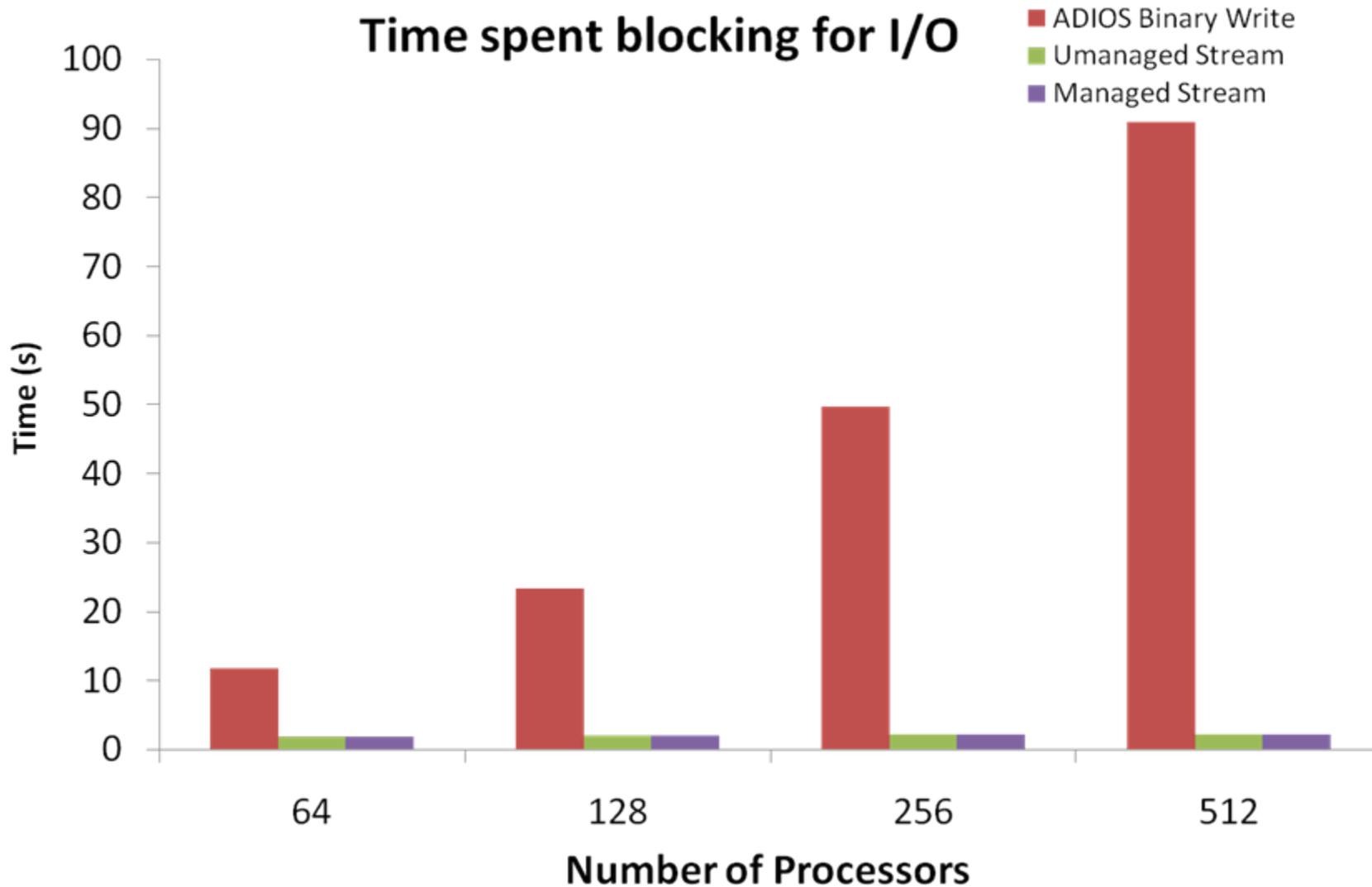




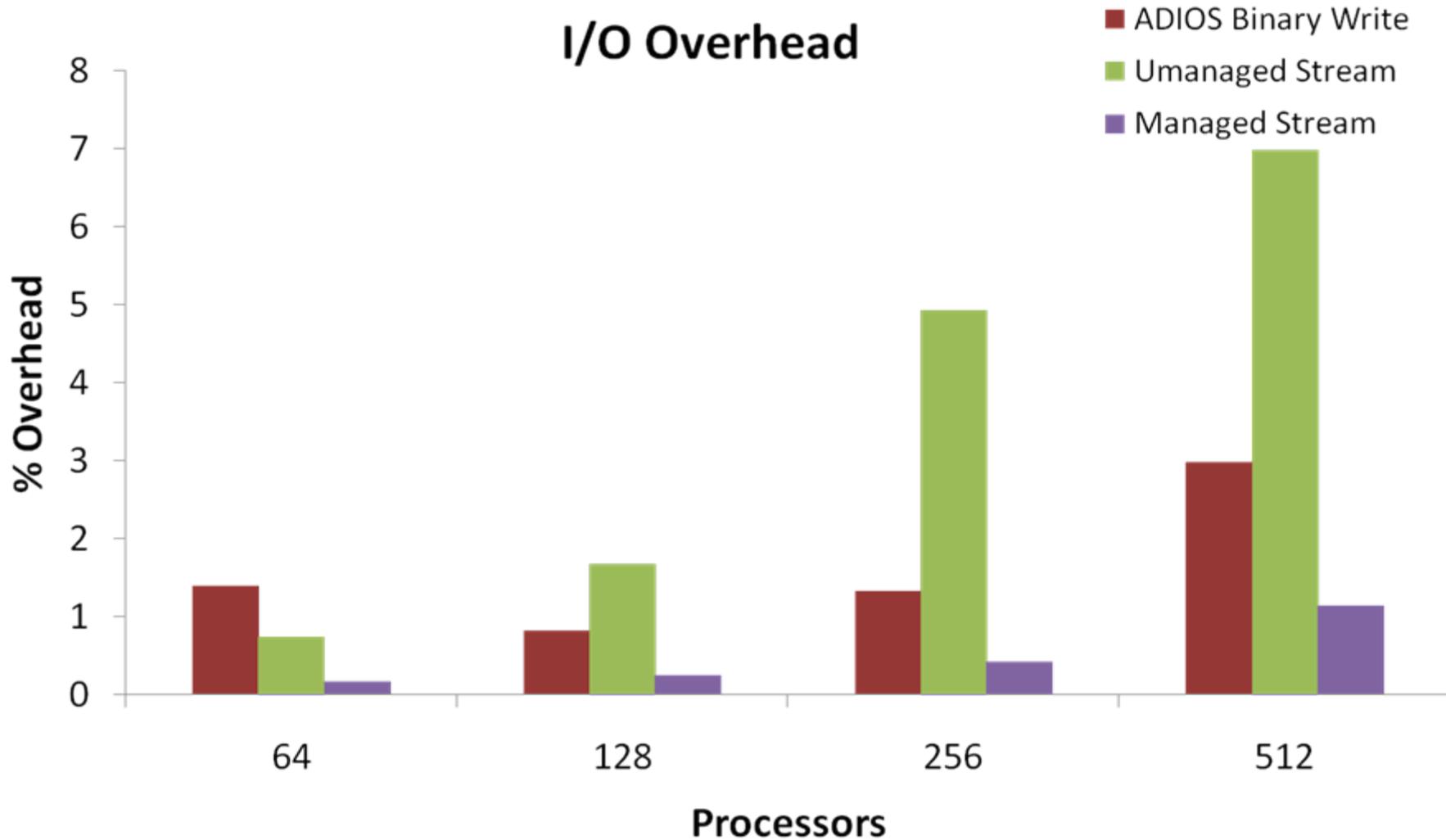
👉 Thanks to Patrick Bridges for the diagram.



- Client-side library for structured stream
- Asynchronous server directed I/O to maximize buffer utilization on DataTap servers
- Graph-structured monitoring and data flows
- Stream management to reduce adverse impact of network contention

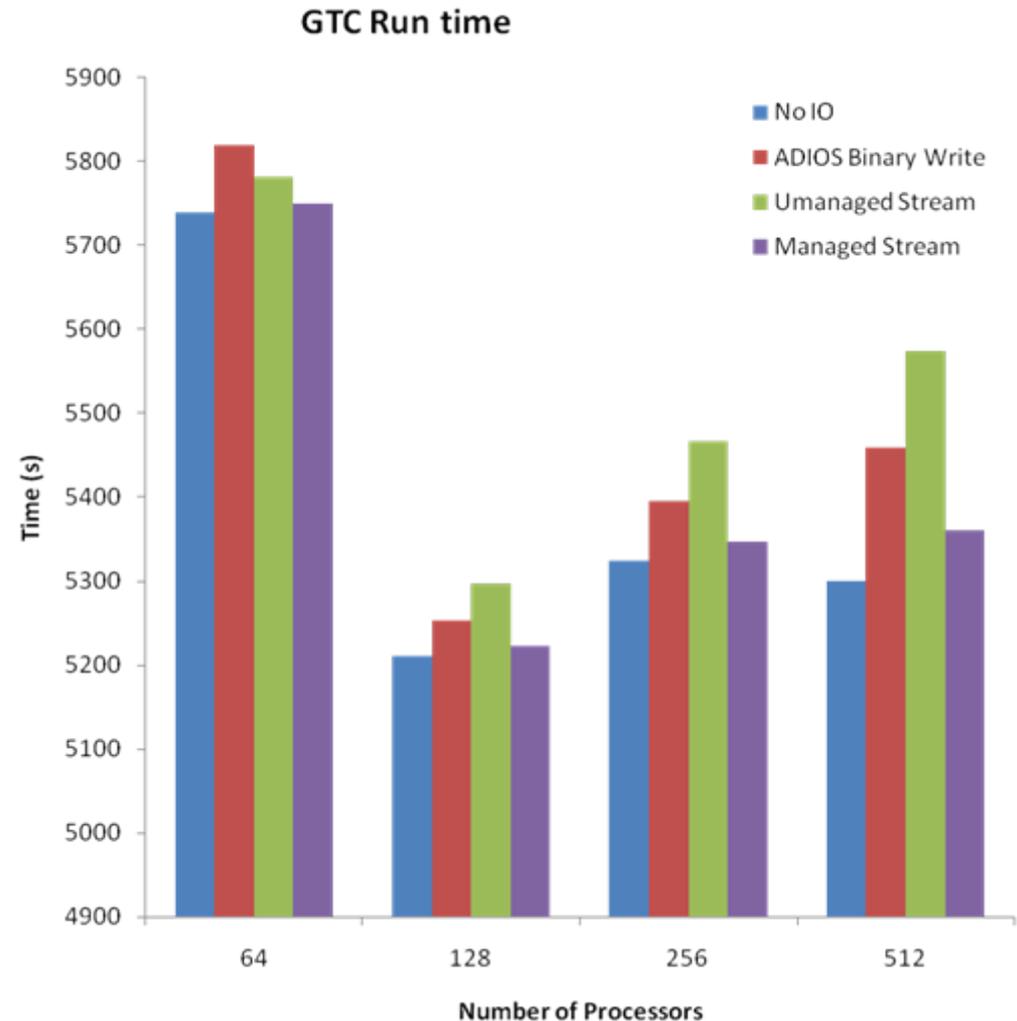


Overhead reduction with DataTap



Blocking time reduction with DataTap 1

- Reduction in blocking results in overall runtime reduction
- Compared to the case of “No IO” the performance of “Managed Stream” is acceptable
- Better stream management will result in a corresponding decrease in overhead





Shortening Application I/O Phases Using Metabots



- Decouple application operations in time/space from I/O phases
- Run eligible activities outside application as *metabots*
- Flexibility in location, timing
- Must consider co-scheduling, data consistency
- Reduce I/O phase duration, disk contention between writers, application memory footprint
- Example below is assembling multiple messages into a single sorted output file

	In-band processing (s)	Metabot processing (s)	Total (s)
Single, in-order writer/reorderer	2113.16	N/A	2113.16
2 storage nodes + metabot	250.91	526.71	777.62
4 storage nodes + metabot	216.52	526.71	743.23



Project Status



- Cray XT (catamount & CNL) **and** i86-Cluster Linux/IB implementations:
 - Cray portal and IB realizations; I/O Servers in Cray CNL
 - also runs on Power-based machines – full BlueGene port under way
- Representative I/O graphs: diagnostics, analysis, restart:
 - current focus on synch. vs. asynch. data annotation/manipulation
- Object data storage with Lustre and Ext3(for comparison)
 - integrating LWFS (UNM/Sandia) and GT software
- Representative Metabots
 - separating creation of directory structures from data I/O; data transformations
- Evaluation with representative petascale codes (fusion-GTC; astrophysics-Chimera; combustion-S3D) on leadership class machine (ORNL – Jaguar)
 - Also material physics (MD) code on Linux/IB.



Ongoing/Future Work



- **Data tap:**
 - Modularize server-directed I/O scheduler; provide user specification of policy
 - Continue generalization to other RDMA transports
- **I/O graphs:**
 - scheduled data movement and differentiated services for diverse data streams
 - automated graph creation/operator generation
- **Metabots:**
 - metabot API and control framework - location and (co-)scheduling
 - specification language for metabot activity (integrated with data specifications for I/O graphs)
- **Associated: ADIOS (joint ORNL/GT project)**
 - Light-weight componentization of application-level I/O interface
 - Enable easy testing and migration to I/OGraph platform for end users
- **Associated: CPA/industry projects: I/O and Platform Virtualization**