Convex Allocations under I/O Constraints

New Challenges in Scheduling Theory, Aussois

Raphaël Bleuse, Giorgio Lucarelli, Denis Trystram

March 31, 2016

Outline

Bluewaters' Architecture

System Summary

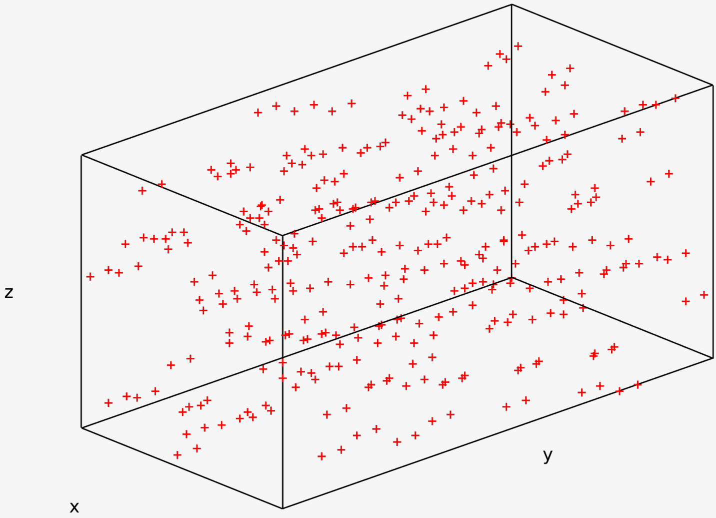
- 288 cabinets
- 26,868 compute nodes / 396,032 cores
- 672 I/O nodes

Network Topology

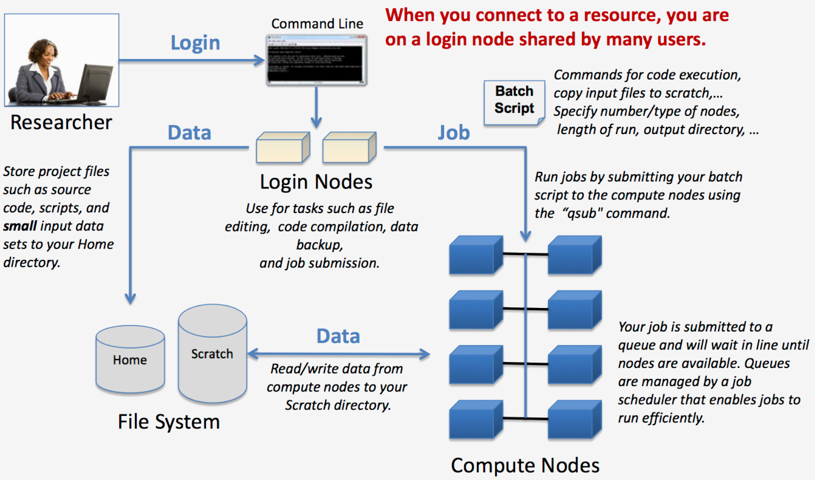
- 3D Torus: $24 \times 24 \times 24$
- single, multi-purpose network
- 2 nodes per interconnection
- static routing (x, then y, then z)

Bluewaters' Architecture

I/O nodes

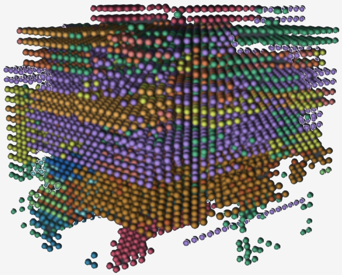


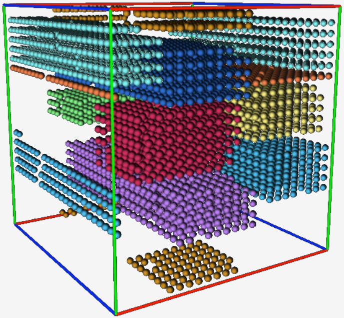
User Interaction



Allocation example

snapshot of the 10 biggest jobs





[Enos et al., 2014]

- fragmentation hinders performances
- convexity improves overall performances
- convexity helps reducing variability in run time

Properties recap.

- unique & multi-purpose interconnection network
- heterogeneous nodes (compute & others)
- convexity (w.r.t. topology) to reduce interferences

Formalization of **ConvexIO**

machine
- interconnection topology
- 2 independent sets of nodes: $\mathcal{S}_{\mathrm{C}}$ & $\mathcal{S}_{\mathrm{I/O}}$
- distribution: mapping of the nodes on the topology

tasks
- independent
- p_j: processing time
- q_j: number of requested nodes in $\mathcal{S}_{\mathrm{C}}$
- κ_j: requested nodes in $\mathcal{S}_{\mathrm{I/O}}$

objective minimizing $C_{\max}$

constraints convex allocations

Further insight on κ_j

κ_j: number of requested nodes in $\mathcal{S}_{\mathrm{I/O}}$

User/System do not care which nodes are allocated:

- spare nodes
- in-situ analysis nodes

κ_j: subset of $\mathcal{S}_{\mathrm{I/O}}$

User/System do care about allocated nodes:

- I/O placement (data)

$\mathcal{NP}$-completeness

Theorem

ConvexIO *is $\mathcal{NP}$-complete.*

Proof.

by reduction to **2-Part**

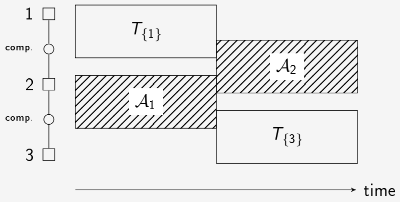
2-Part Given $\{\alpha_j\}_j$ such that $\sum \alpha_j = A$.
Find $\mathcal{A}_1, \mathcal{A}_2$ such that $\sum_{\alpha_j \in \mathcal{A}_1} \alpha_j = \sum_{\alpha_j \in \mathcal{A}_2} \alpha_j = \frac{A}{2}$.

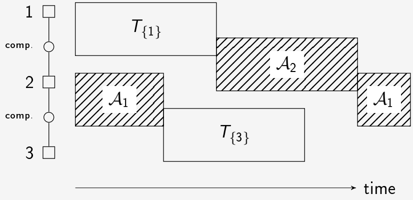
ConvexIO

- 2 compute nodes + 3 I/O nodes.
- $\alpha_j \mapsto T_j : q_j = 1, \kappa_j = \{2\}, p_j = \alpha_j$.
- for io $\in \{1, 3\}$, add $T_{\{\text{io}\}} : q_j = 1, \kappa_j = \{\text{io}\}, p_j = \frac{A}{2}$.

□

$\mathcal{NP}$-completeness (*bis*)



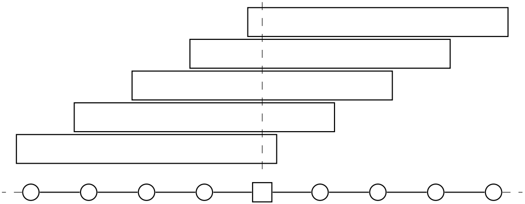


Approximation Algorithm on 1D topology

Setup

- 1D line/toric topology.
- Relates to $P|fix_j|C_{\max}$ and $P|set_j|C_{\max}$.

Eligible allocations of a task



Approximation Algorithm on 1D topology

1. determine an allocation of nodes minimizing the overall load
2. derive a feasible schedule using Gergov's DSA algorithm [Gergov, 1999]

Approximation Algorithm on 1D topology

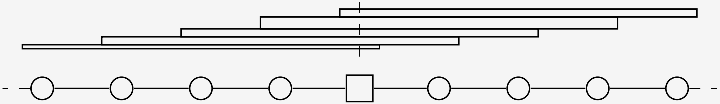
Allocation step: linear program

$$
\begin{aligned}
\min\ & \mathcal{L} \\
\text{s.t.}\ & \mathcal{L} \geq L_i \quad \forall i \\
& L_i \geq \sum_j x_{j,s} \mathbf{1}_{i \in [s, s+q_j]} p_j \quad \forall i \\
& \sum_s x_{j,s} = 1 \quad \forall j \\
& x_{j,s} \quad \forall j, s \\
& + \text{I/O pinning constraints}
\end{aligned}
$$

Approximation Algorithm on 1D topology

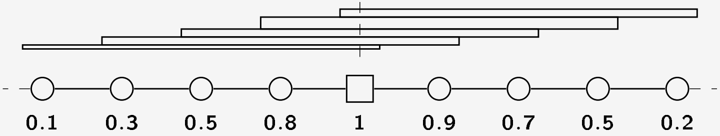
Allocation step: linear program

$$
\begin{aligned}
\min\ & \mathcal{L} \\
\text{s.t.}\ & \mathcal{L} \geq L_i \quad \forall i \\
& L_i \geq \sum_j x_{j,s} \mathbf{1}_{i \in [s, s+q_j]} p_j \quad \forall i \\
& \sum_s x_{j,s} = 1 \quad \forall j \\
& x_{j,s} \quad \forall j, s \\
& + \text{I/O pinning constraints}
\end{aligned}
$$



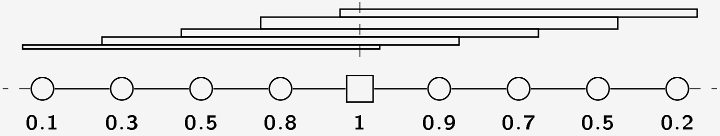
Approximation Algorithm on 1D topology

Allocation step: rounding

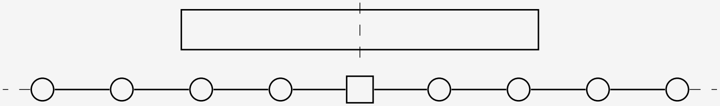


Approximation Algorithm on 1D topology

Allocation step: rounding



⇓



Contributions

- model for HPC infrastructure
- approximation algorithm for **ConvexIO** on 1D topology

Directions

- improve approximation ratio
- formulation without linear programing
- extend topology dimension

Questions?

References I

Enos, J., Bauer, G., Brunner, R., Islam, S., Steed, M., Jackson, D., and Fiedler, R. (2014).
Topology-Aware Job Scheduling Strategies for Torus Networks.
Cray User Group.

Gergov, J. (1999).
Algorithms for Compile-time Memory Optimization.
In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '99, pages 907–908, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.