Mars On-site Shared Analytics Information and Computing


MOSAIC Project Overview

We consider:

- a heterogeneous network
- with time-varying communication links
- a task network
  - some tasks are pinned to some nodes
  - some tasks are optional
  - tasks have dependencies

We study the optimal distribution of processing tasks/power across a time-varying network of agents

Bottom line

- We present a Mixed Integer Linear Program that is used to generate a schedule containing computation and communication tasks for each agent
- Tested in a distributed system
Heterogeneous Network Example
High Performance Spaceflight Computing (HPSC)

- Lander CPU support (laptop level)
- Puffer scaled-down (pi level)
- Semi-autonomous network

PUFFERs (Pop-Up Flat-Folding Explorer Robots)

http://www.kiss.caltech.edu/lectures/2019_PUFFER.html
Notional Mission Concept
Small / Multi-robot systems with intermittent connectivity

- CPU support nodes
- Network failures
- *Intentional* disconnects

http://www.kiss.caltech.edu/lectures/2019_PUFFER.html
Time-varying communication links

Contact Plan

• Mobile autonomous agents can share *intended* motion to allow motion-aware relay (data mules)
• Delay Tolerant Networking

[Diagram showing communication links]

https://www.nasa.gov/content/dtn
Task/Software Network Models

(a) Housekeeping and (b) Science task chains
• “Archive” is optional, but rewarding
Scheduling Problem
Centralized version

Network of Assets

Task/software Networks

Communication Network Topology (Contact Plan)

Map to

Goal: Schedule the tasks and the data communication across the network
Reward \[ \sum_{i=1}^{N} \sum_{T \in \mathbb{R}} \sum_{c=1}^{C_d^* - \tau_i(T)} r(T)X(i, T, c) \]

Agent, task, time

Max. reward of optional tasks completed

\[ \sum_{i=1}^{N} \sum_{T \in \mathbb{R}} \sum_{c=1}^{C_d^* - \tau_i(T)} X(i, T, c) \leq 1 \quad \forall T \in \mathbb{R} \]  \hspace{1cm} (2a)

\[ \sum_{i=1}^{N} \sum_{T \in \mathbb{T} \setminus \mathbb{R}} \sum_{c=1}^{C_d^* - \tau_i(T)} X(i, T, c) = 1 \quad \forall T \in \mathbb{T} \setminus \mathbb{R} \]  \hspace{1cm} (2b)

\[ X(i, T, c) \leq D(i, L, c) \]
\[ \forall i \in [1, \ldots, N], T \in [1, \ldots, M], L \in P_T, c \in [1, \ldots, C_d^*] \]  \hspace{1cm} (2c)

\[ \sum_{T=1}^{M} \left[ \sum_{j=1}^{N} \left( C(i, j, T, c) + C(j, i, T, c) \right) + \sum_{\hat{c} = \max(1, c - \tau_i(T))}^{c} X(i, T, \hat{c}) \right] \leq 1 \]
\[ \forall i \in [1, \ldots, N], c \in [1, \ldots, C_d^*] \]  \hspace{1cm} (2d)

\[ D(i, T, c + 1) - D(i, T, c) \leq \sum_{\tau=1}^{c} \sum_{j=1}^{N} \frac{r_{ji}(c)}{s(T)} C(j, i, T, c) + \sum_{\tau=1}^{c - \tau_i(T)} X(i, T, c) \]
\[ \forall i \in [1, \ldots, N], T \in [1, \ldots, M], c \in [1, \ldots, C_d^* - 1] \]  \hspace{1cm} (2e)

\[ C(i, j, T, c) \leq D(i, T, c) \quad \forall i, j \in [1, \ldots, N], T \in [1, \ldots, M], c \in [1, \ldots, T] \]  \hspace{1cm} (2f)

\[ D(i, T, 1) = 0 \quad \forall i \in [1, \ldots, N], T \in [1, \ldots, M] \]  \hspace{1cm} (2g)

All required tasks completed once

Optional tasks are performed at most once

Only start a task once you have req inputs

One thing at a time (cpu, coms, or idle)

Only have data by calculating or receiving

Only communicate what you have

Start with no data
Centralized vs Distributed

MILP solves centralized version

Each agent has its own task list
Centralized vs Distributed

- Gossip, Plan, Act
- Caveat: consensus is hard, and we do require it
- Mostly static task sets are best

Every agent knows all tasks and network states
Experimental Results

• Setup
  • Simulation (15 puffers + base station)
  • Field Test (JPL Mars Yard)

• Interesting cases observed:
  • Data Mulling (Simulation only)
  • Science Clusters
  • Data Relay
  • Assembly Line
Simulation: Example Solution
Data Mule
Simulation: Example Solution

Data Mule

Base station

00:00
Field Tests (Video Preview)

Fully Distributed Systems
Field Tests

System operated for 3 hours
Example code / solutions at:
https://github.com/nasa/mosaic
Field Tests (Callout)

Please stop by the ICAPS demo on Saturday, July 13th to see this live!
• Session B, Desk 9
Bottom Line

We present a Mixed Integer Linear Program that is used to generate schedules containing computation and communication tasks for

- on a heterogeneous network
- with time-varying communication links
- a task network (plan) per agent
  - some tasks are pinned to some nodes
  - some tasks are optional
  - tasks have dependencies
- tested in a distributed system

Showed
- Emergent data mules, assembly lines in distributed system

Please stop by the ICAPS demo on Saturday, July 13th to see this live!
- Session B, Desk 9
Stack Overview

1. Pluggable Distributed Resource Allocator
2. Resource Registry: What tasks, battery, cpu, storage for each nearby node
3. (Ideally) Software Tasks for other nodes
4. All sync’d over the network