Google

Machine Learning Fleet Efficiency

Analyzing and Optimizing Large-Scale Google TPU Systems with ML Productivity Goodput

Arissa Wongpanich, *Tayo Oguntebi[*], Jose Baiocchi Paredes, Emma Wang, Mangpo Phothilimthana[*], Ritwika Mitra, Zongwei Zhou[*], Naveen Kumar, Vijay Janapa Reddi[*]*

[*] *Work done while at Google*

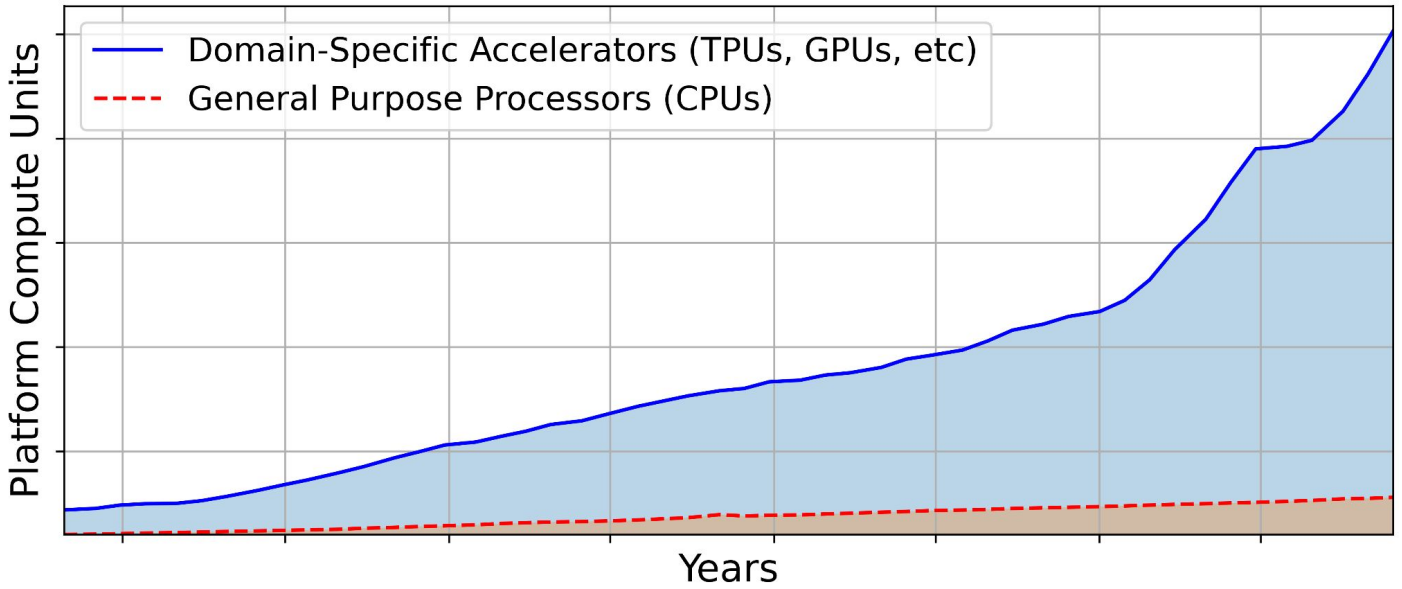
MLSys Industry Track, May 2026

How do we **measure** and **improve** the efficiency of a Machine Learning Fleet?

Challenge #1: Hardware heterogeneity

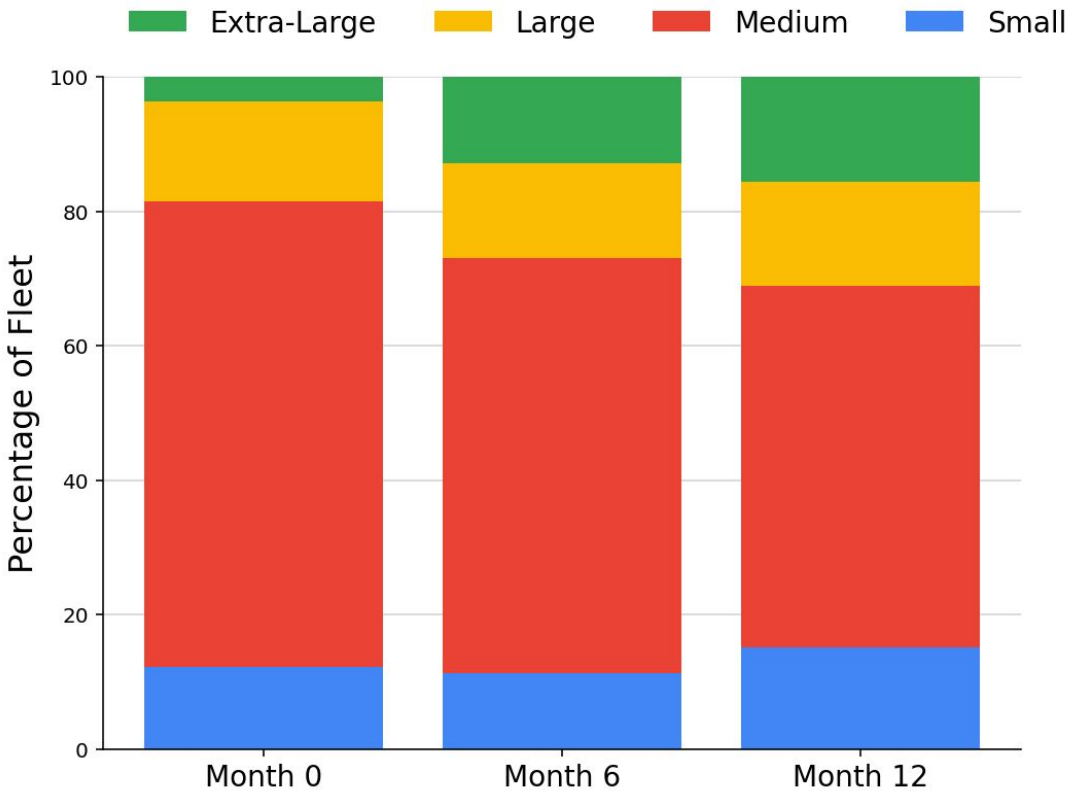
"The next decade will see a Cambrian explosion of novel computer architectures..."

- Hennessy & Patterson, "A new golden age for computer architecture", Communications of the ACM, Feb. 2019.



*Data obtained from Google's production fleet. Axes are obfuscated to preserve confidentiality.

Challenge #2: Workload heterogeneity



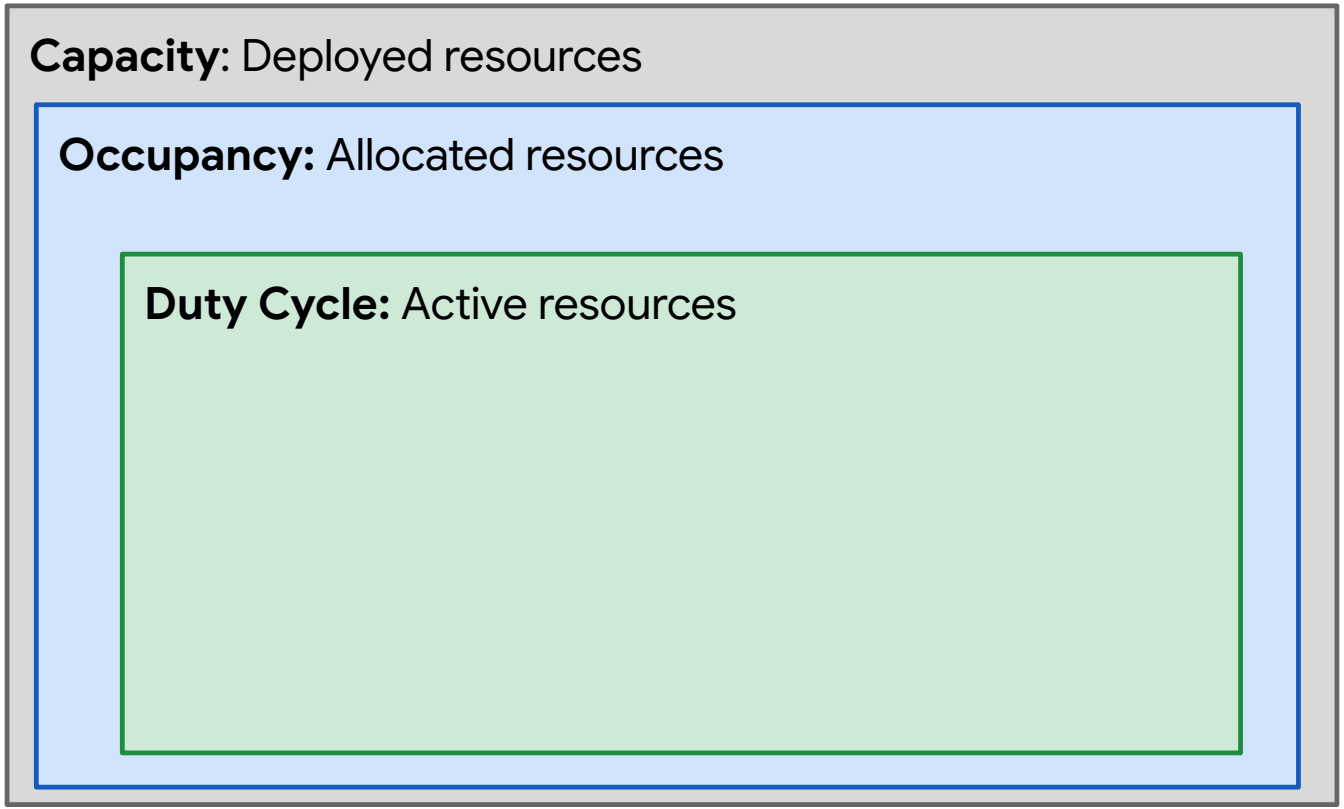
Fleet breakdown by workload topology size

A proxy for workload type (serving vs. training vs. bulk inference)

Extra-large (O(1k) chips) and small jobs becoming more prevalent

*How do we **measure** ML fleet efficiency?*

Traditional fleet efficiency metrics: Capacity-based



Pitfalls of traditional efficiency metrics

High capacity ≠ high resource availability.	**High occupancy ≠ productivity.**	**Duty cycle ≠ useful work.**
Capacity ignores scheduling constraints. → Goal: **Scheduling Goodput**	Occupancy resources aren't necessarily utilized. → Goal: **Runtime Goodput**	DC measures **activity**, not **productivity**. → Goal: **Program Goodput**

Idea: Decompose metric into 3 factors

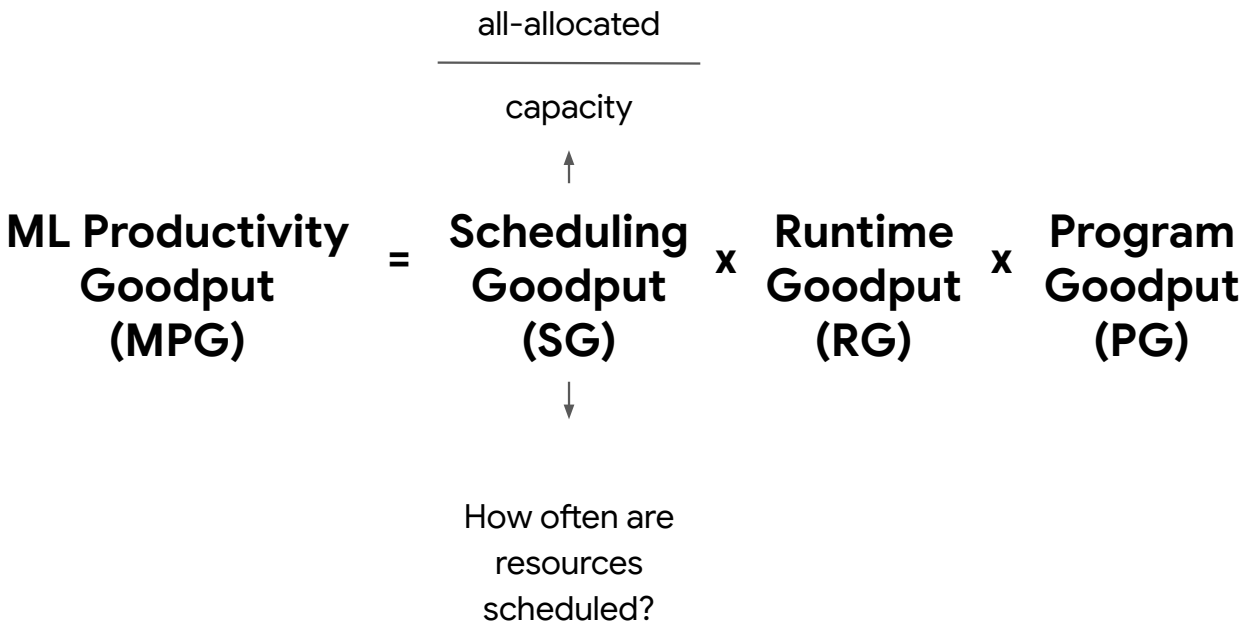
ML Productivity Goodput (MPG) = **Scheduling Goodput (SG)** x **Runtime Goodput (RG)** x **Program Goodput (PG)**

Idea: Decompose metric into 3 factors

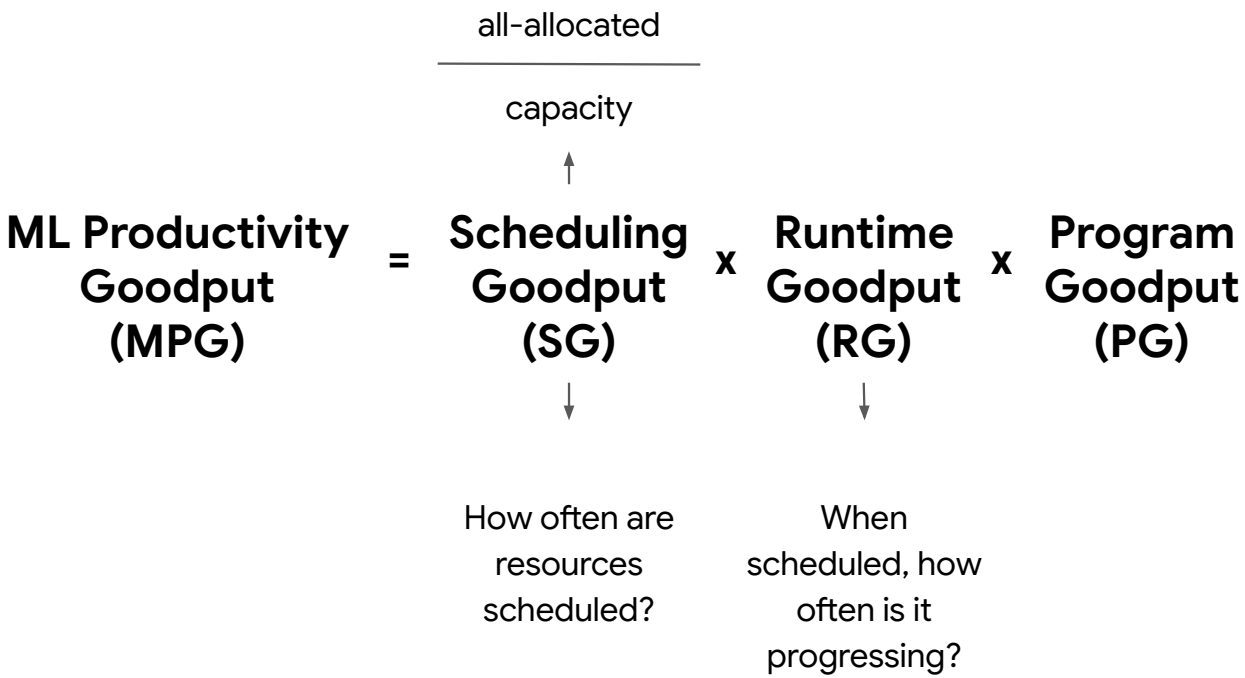
ML Productivity Goodput (MPG) = **Scheduling Goodput (SG)** x **Runtime Goodput (RG)** x **Program Goodput (PG)**

How often are resources scheduled?

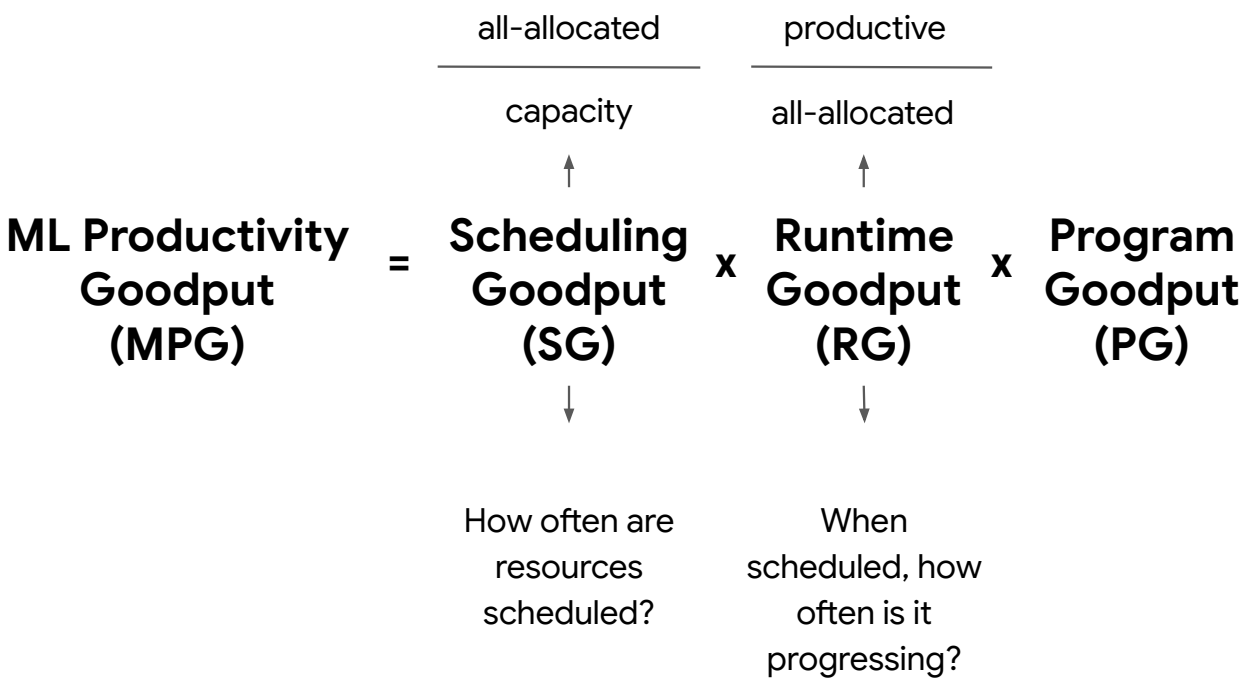
Idea: Decompose metric into 3 factors



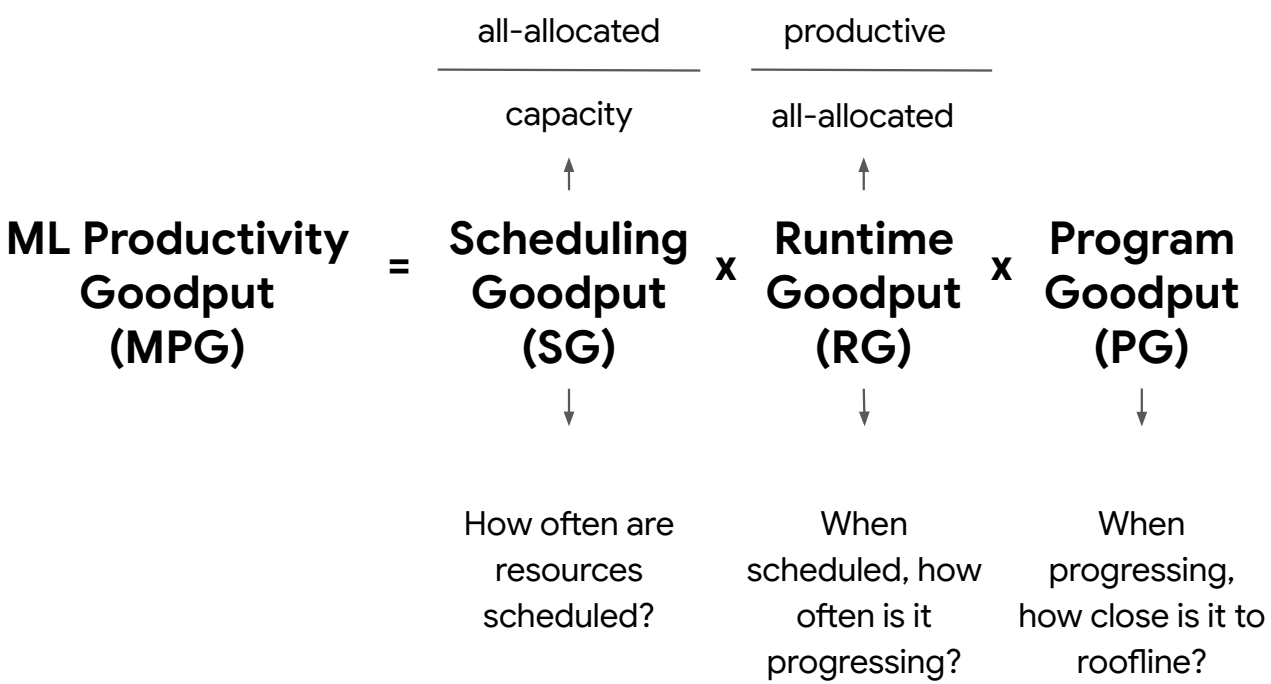
Idea: Decompose metric into 3 factors



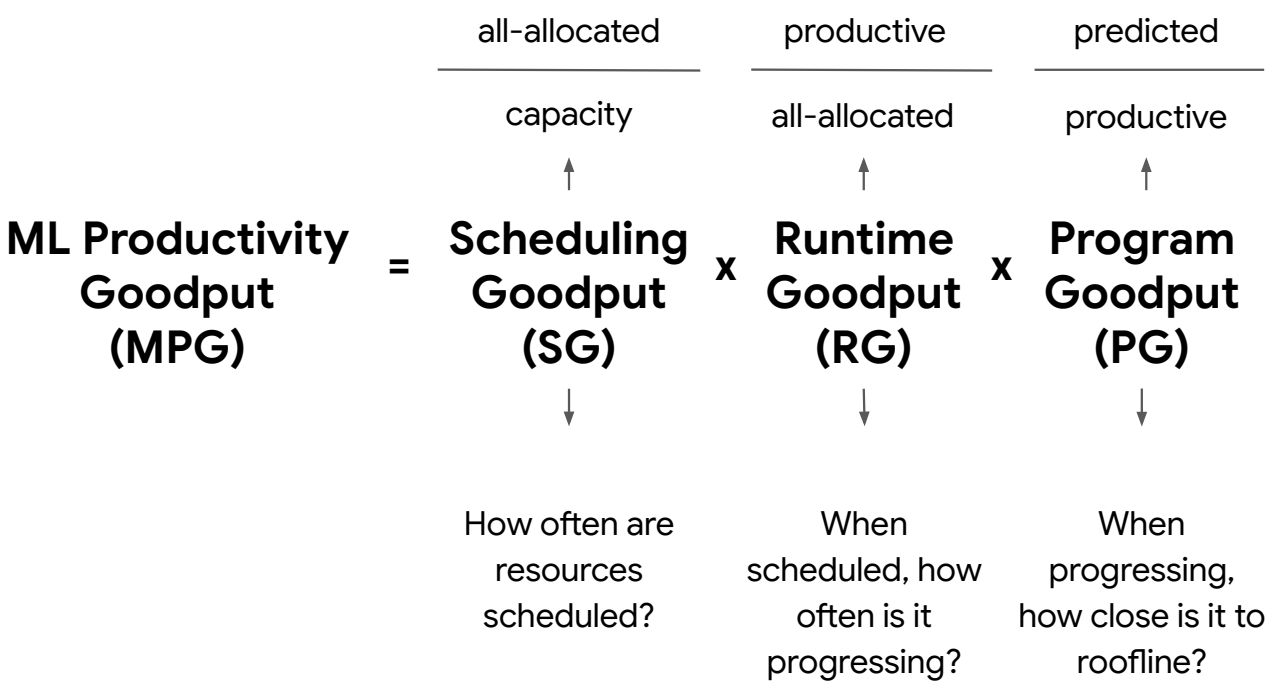
Idea: Decompose metric into 3 factors



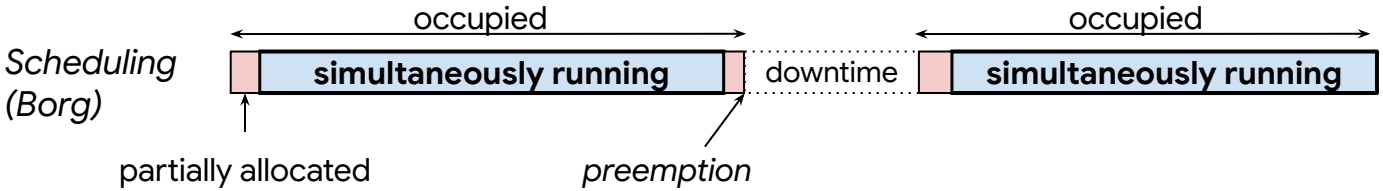
Idea: Decompose metric into 3 factors



Idea: Decompose metric into 3 factors

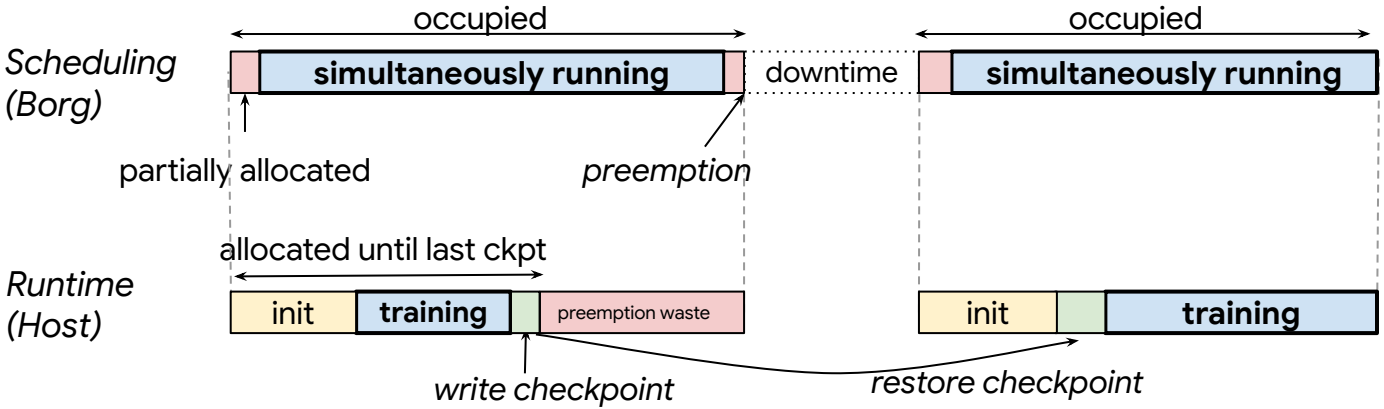


ML Productivity Goodput: Life of a job



Scheduling Goodput
Have all necessary resources?

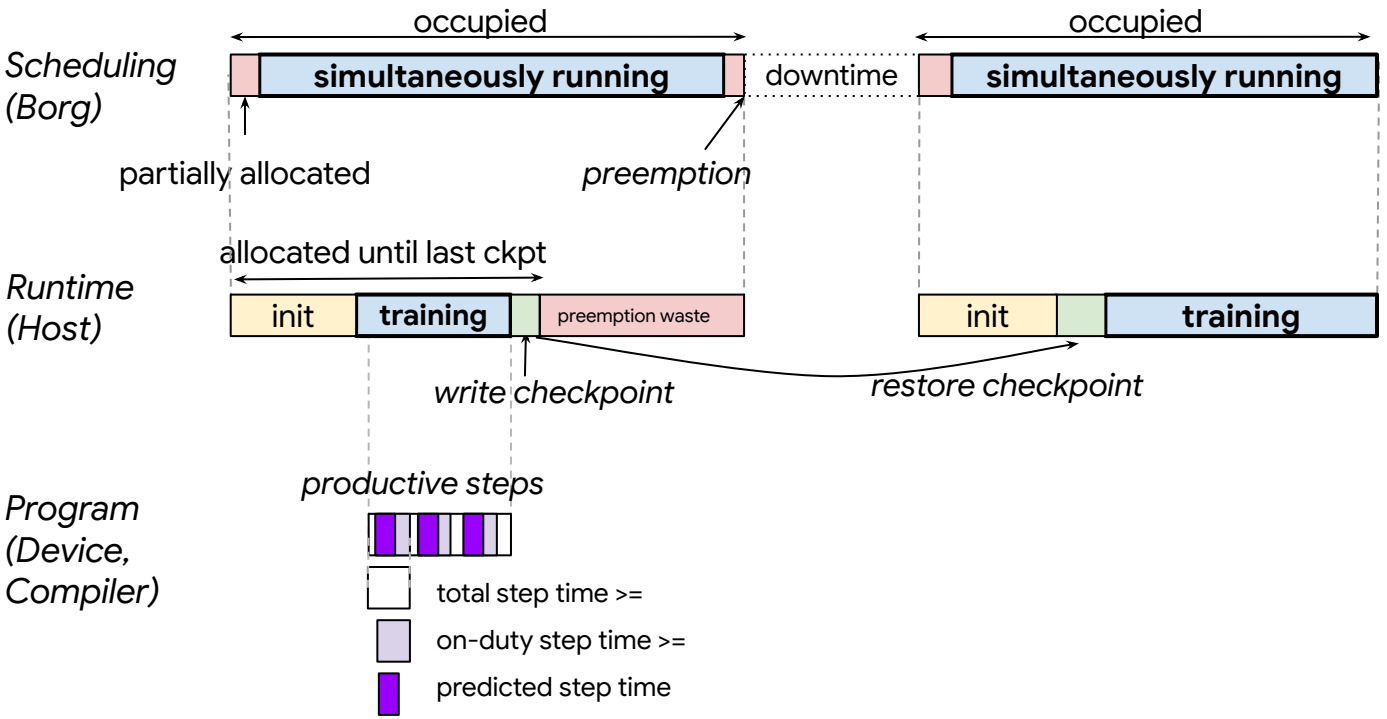
ML Productivity Goodput: Life of a job



Scheduling Goodput
Have all necessary resources?

Runtime Goodput
Making progress?

ML Productivity Goodput: Life of a job



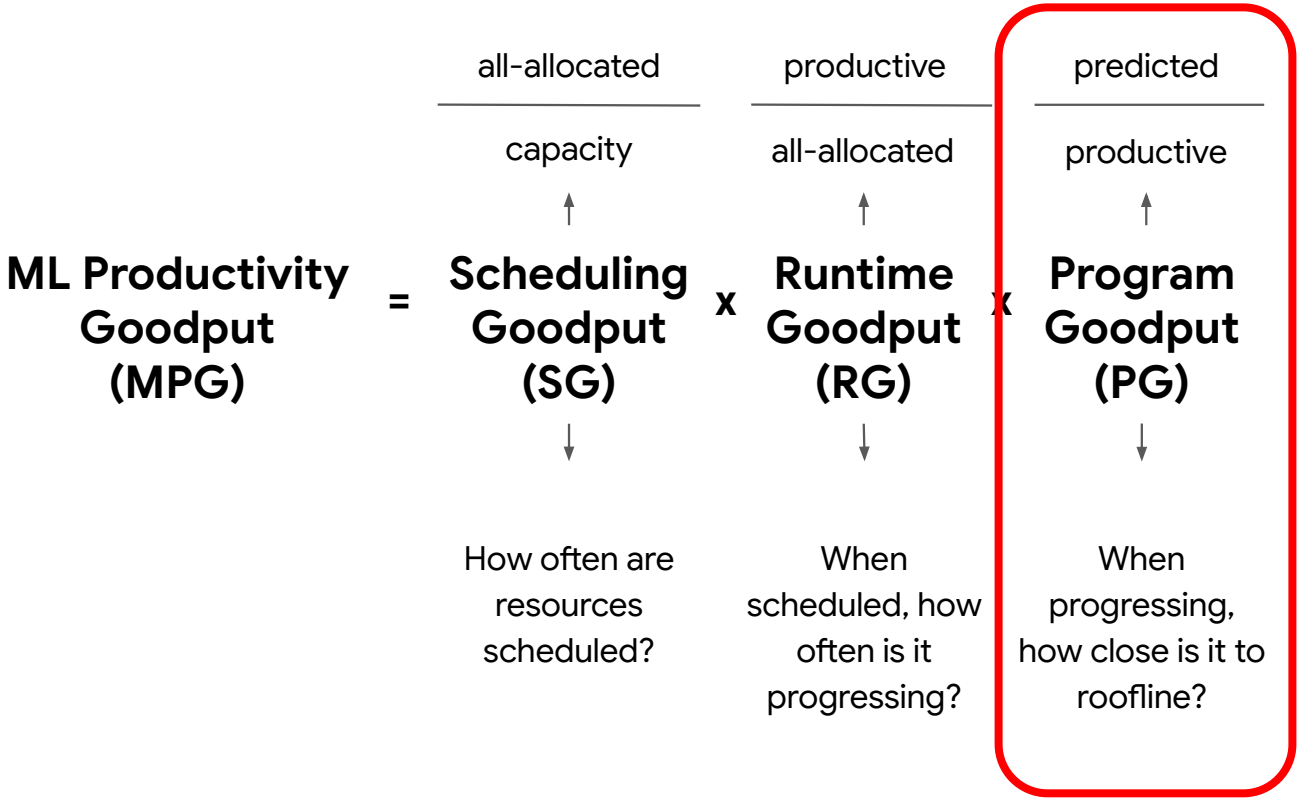
Scheduling Goodput
Have all necessary resources?

Runtime Goodput
Making progress?

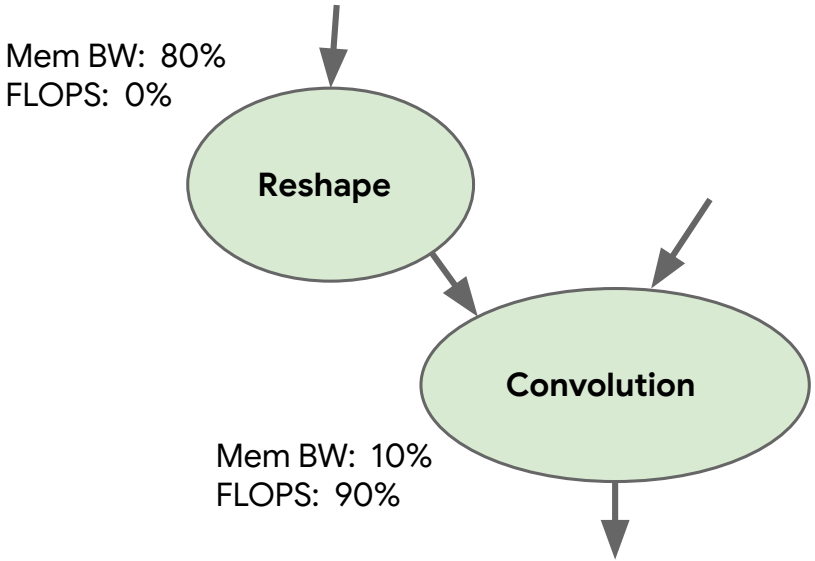
Program Goodput
When busy, how close to roofline?

*How do we **improve** ML fleet efficiency?*

ML Productivity Goodput

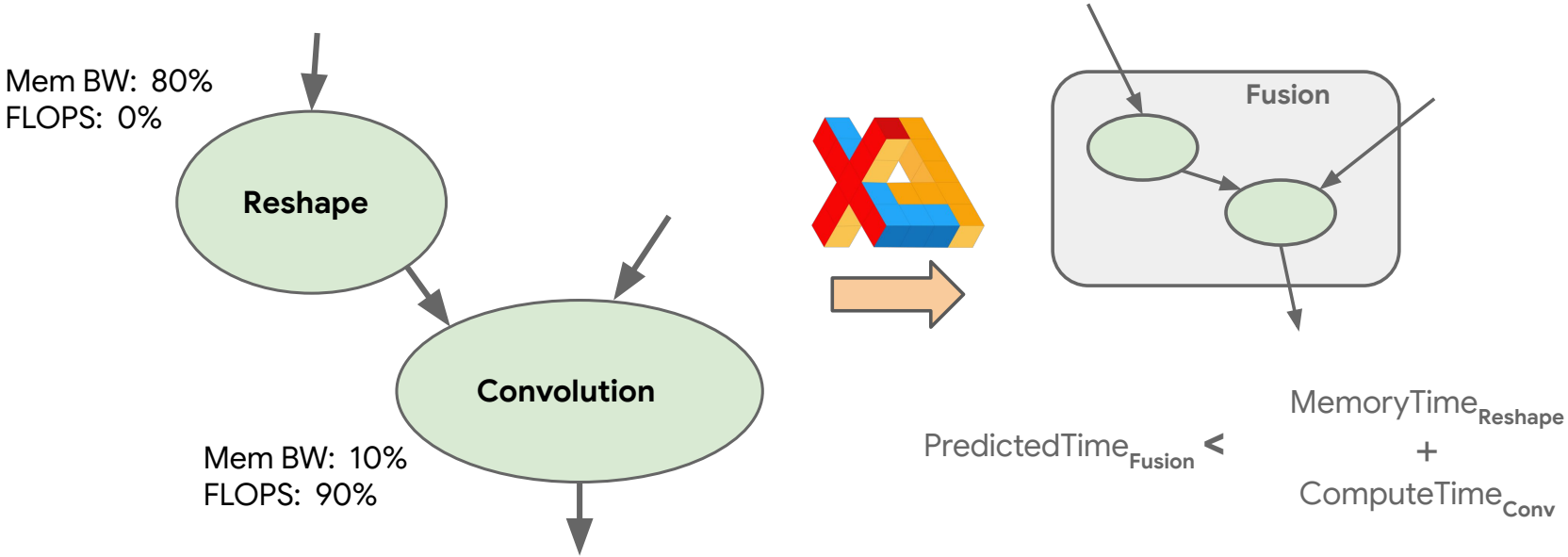


Program Goodput: Why not roofline efficiency?



This subgraph would have a high roofline efficiency score.

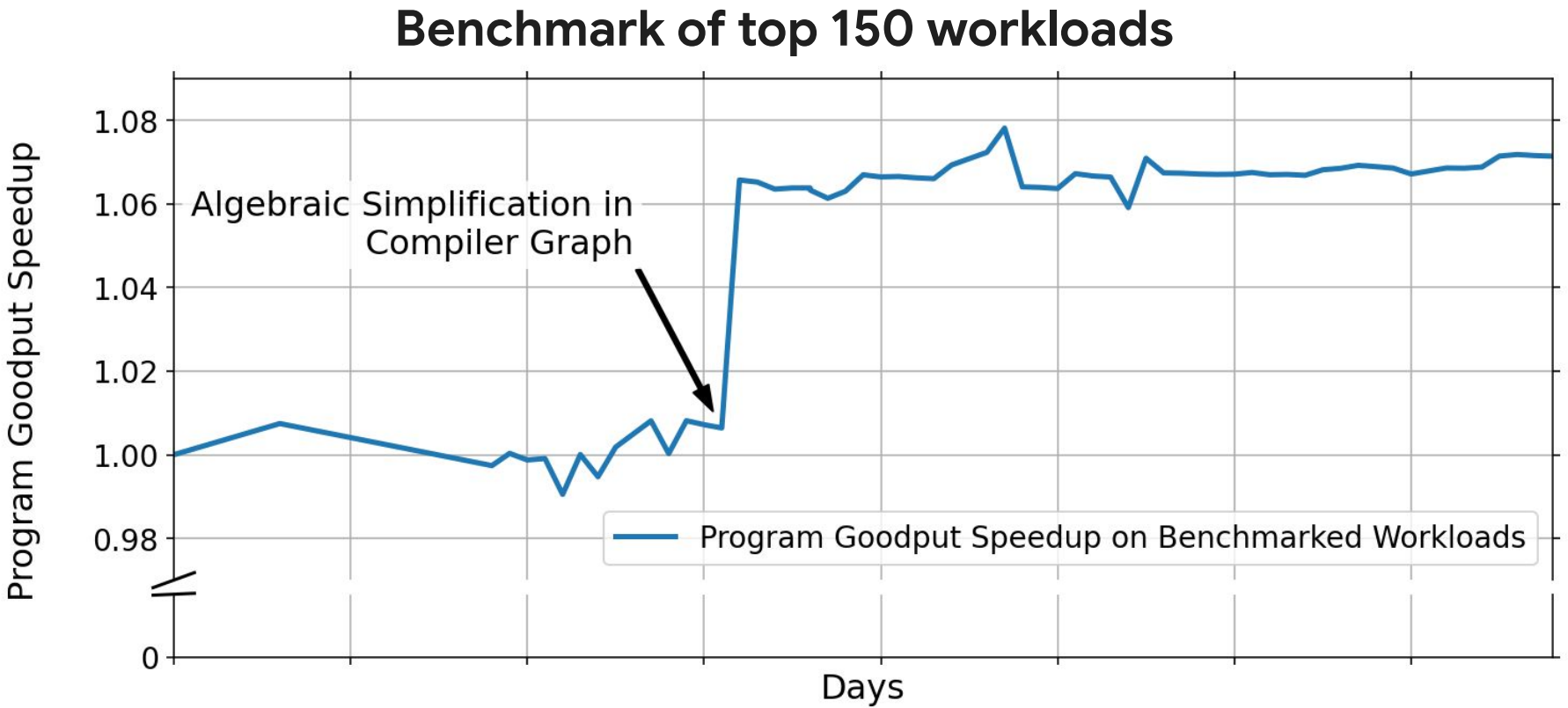
Program Goodput: Why not roofline efficiency?



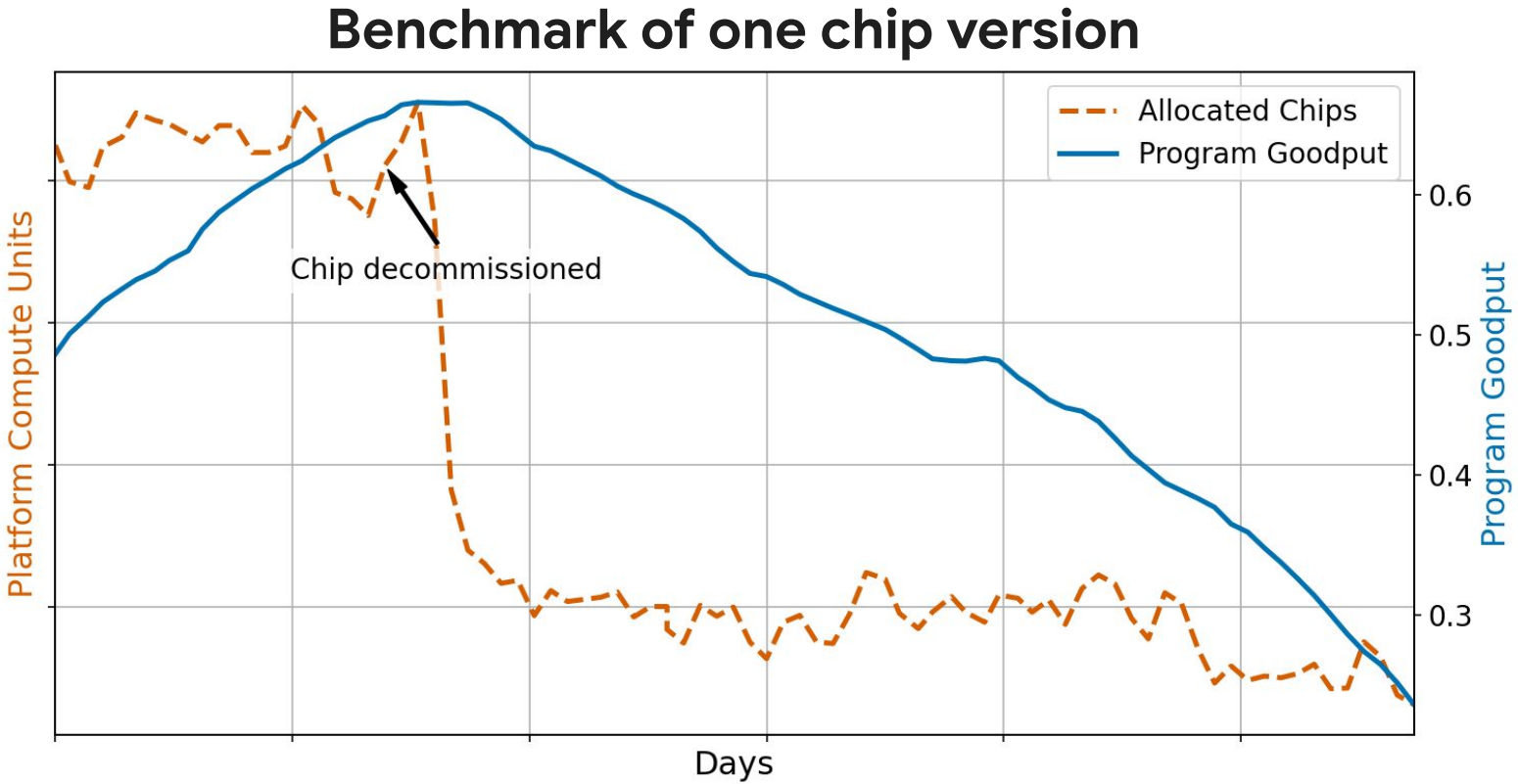
This subgraph would have a high roofline efficiency score.

But fusing/overlapping would be faster overall.

Tracking compiler improvements with Program Goodput

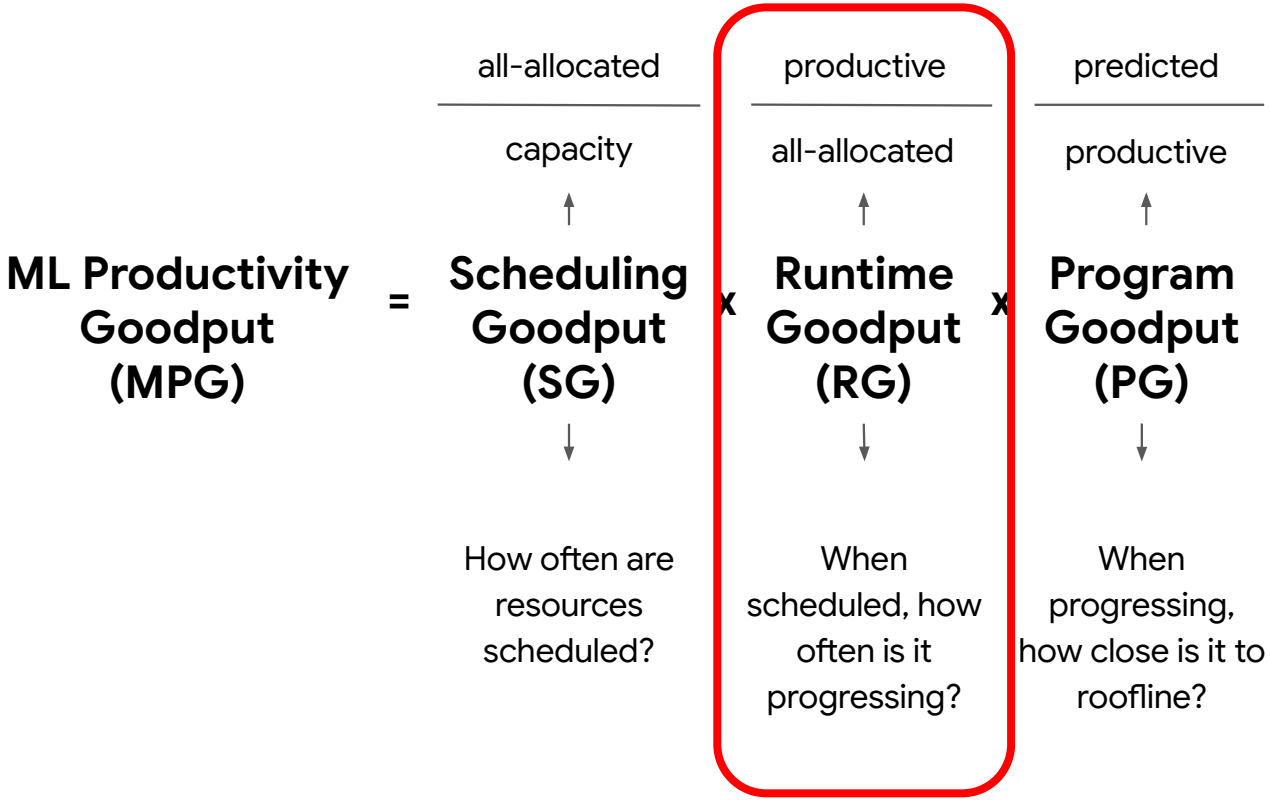


Tracking hardware / workload dynamics with Program Goodput

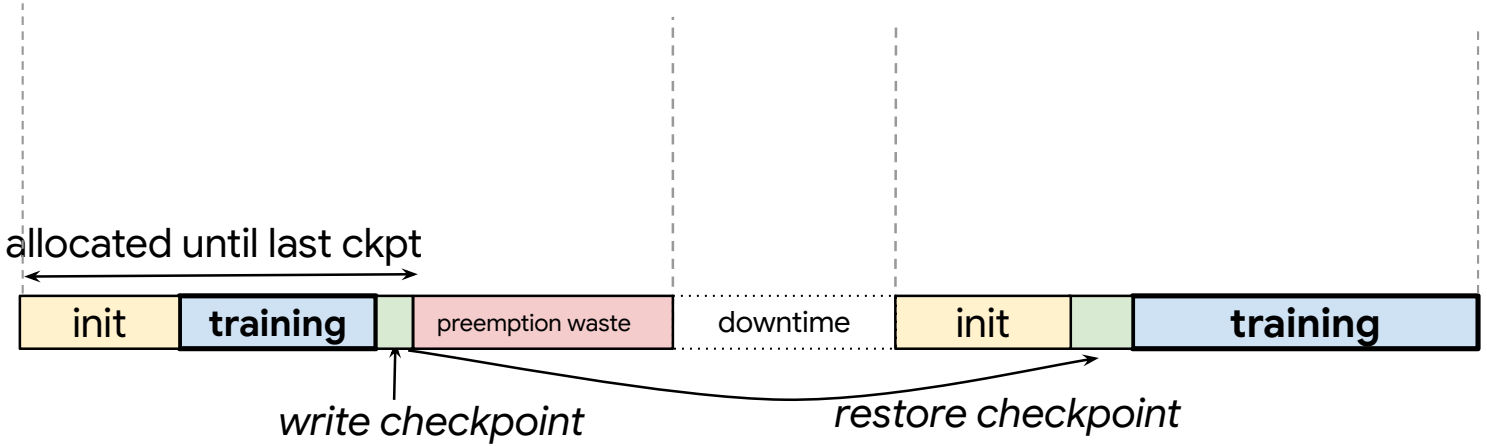


*Data obtained from Google's production fleet. Axes are obfuscated to preserve confidentiality.

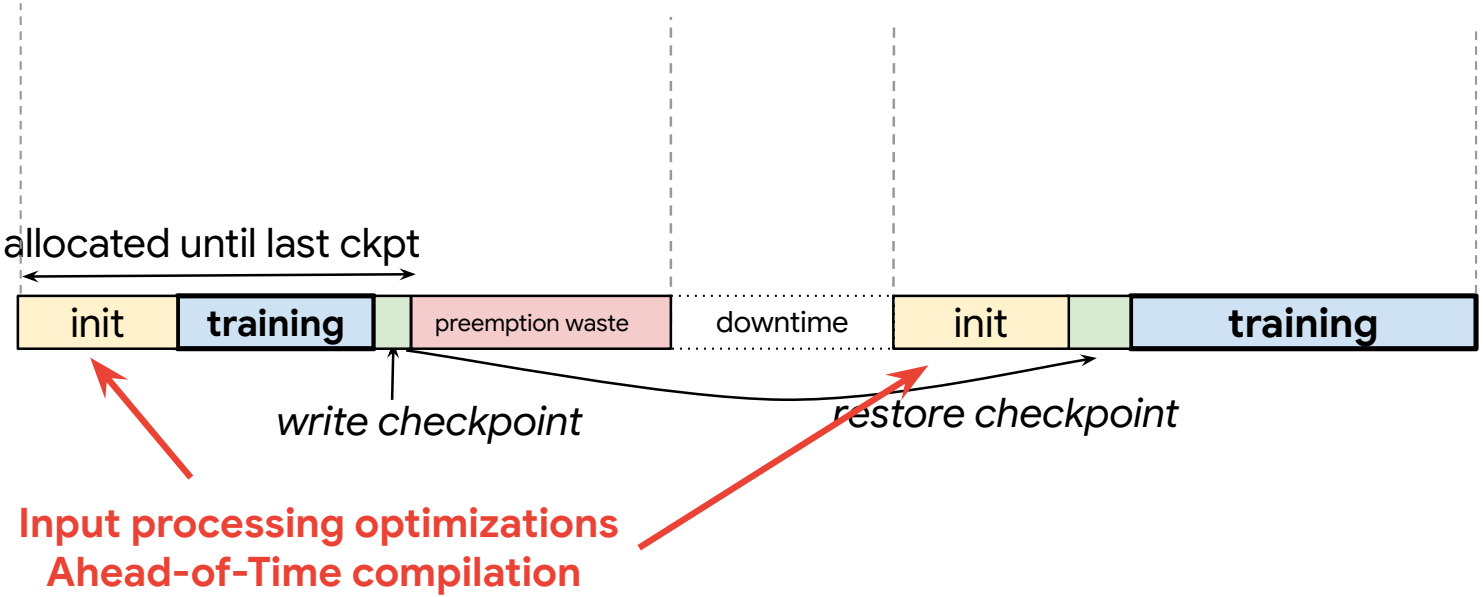
ML Productivity Goodput



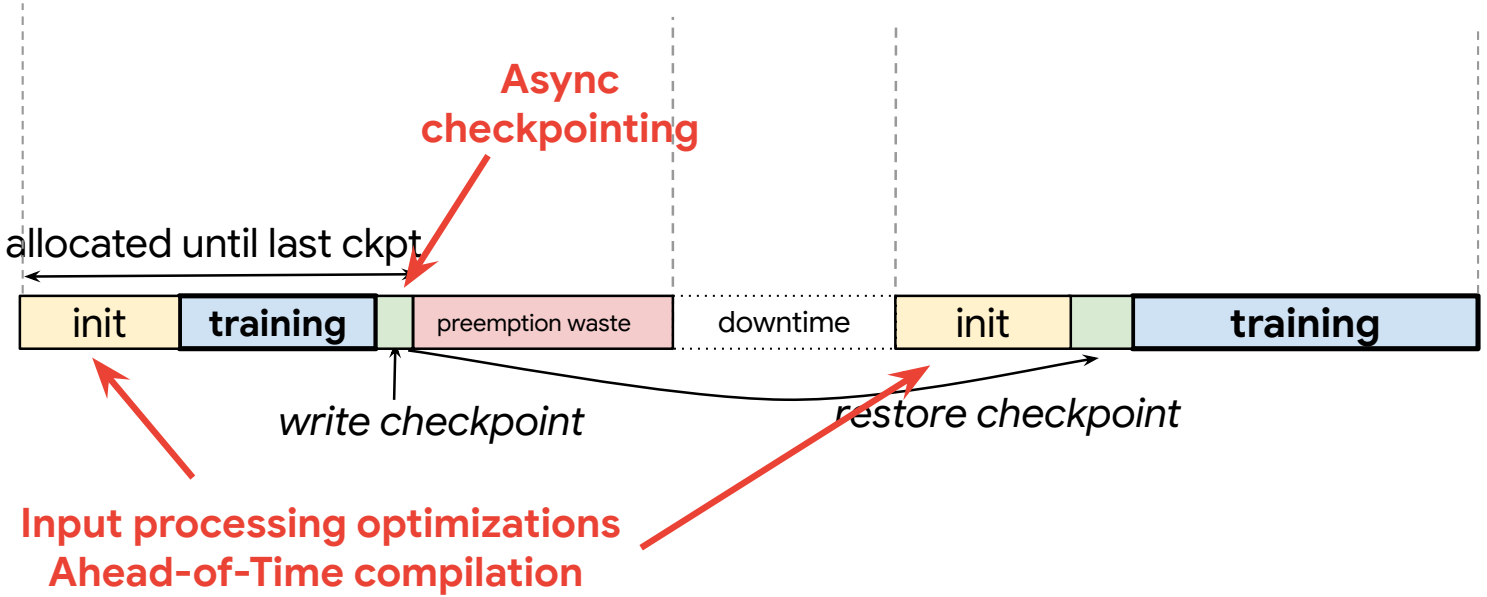
How do we improve runtime goodput?



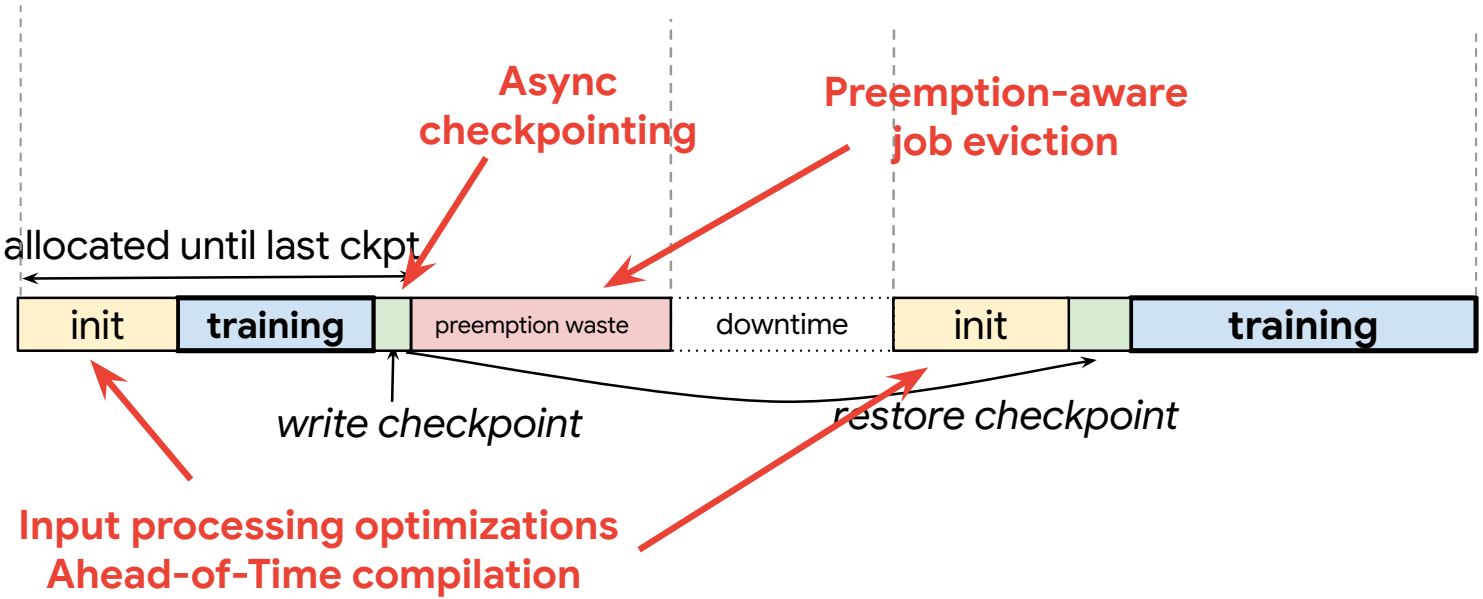
How do we improve runtime goodput?



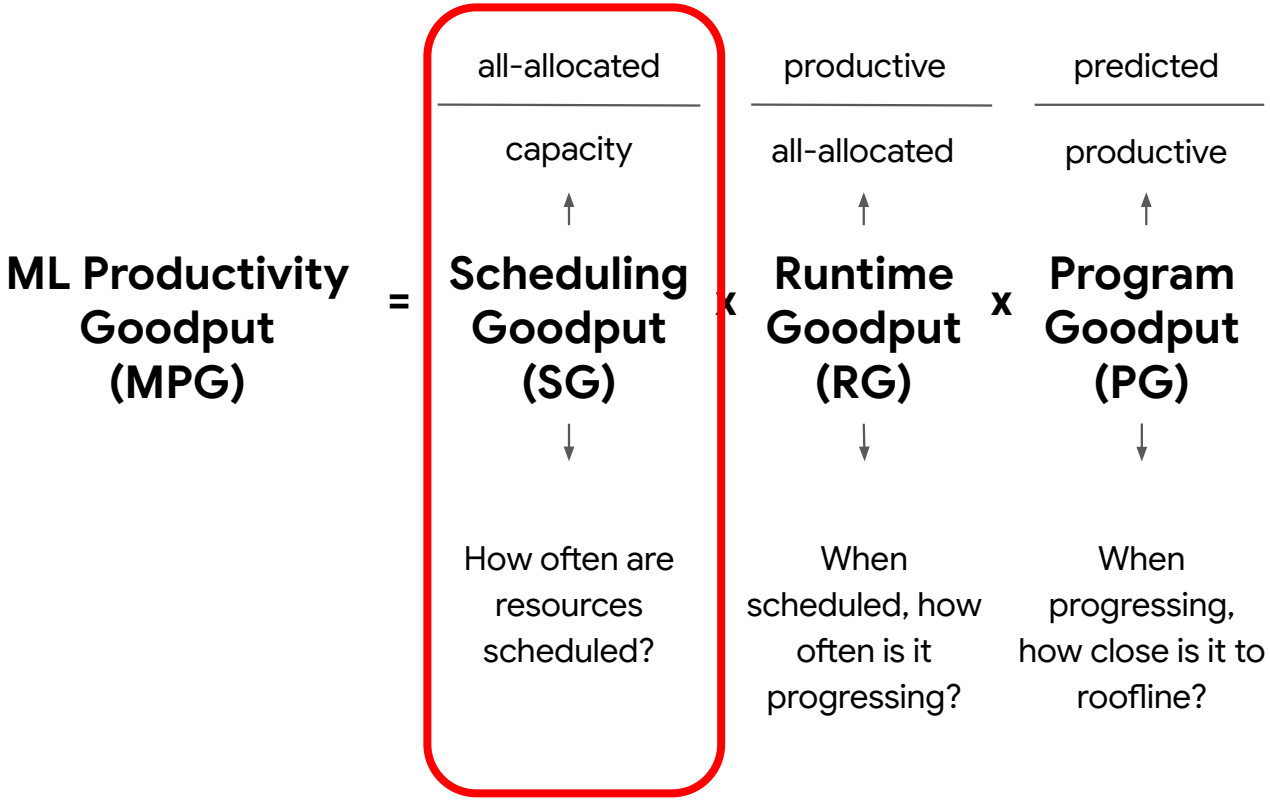
How do we improve runtime goodput?



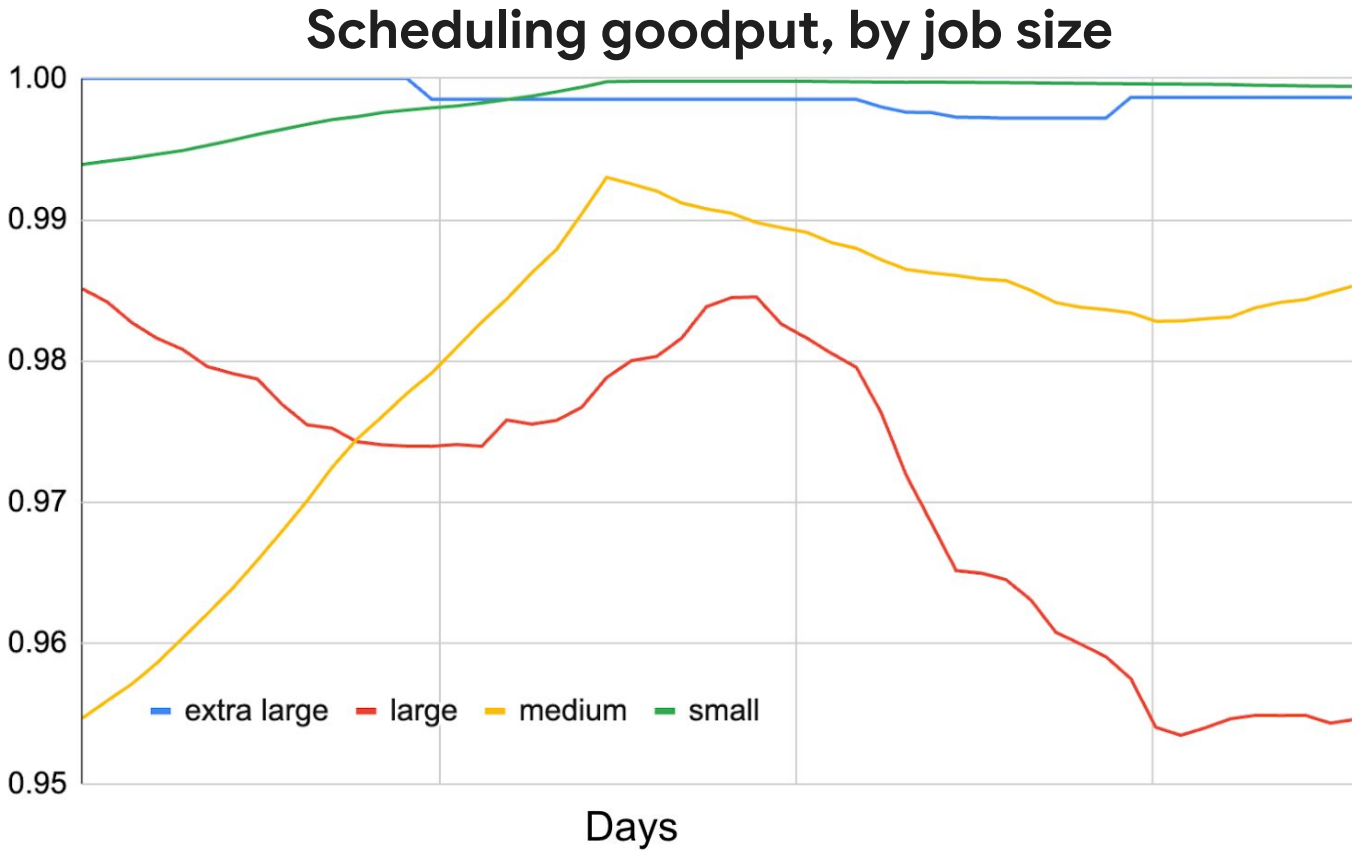
How do we improve runtime goodput?



ML Productivity Goodput



Tracking workload scheduling efficiency



SG is >95%

Extra-large (O(1k chips) and small jobs are scheduled more easily

Conclusion

- ML fleets are complex: heterogeneous and evolving
- ML Productivity Goodput enables performance reasoning
  - Segmented deep-dive views of data
  - View trends and diagnose regressions
  - Develop and deploy targeted optimizations

Google

Thank you!

Link to paper: