CherryPick: Adaptively Unearthing the Best Cloud Configurations for Big Data Analytics

O. Alipourfard et al.
Overview

- Background
- Prior work
- CherryPick
- Evaluation
- Criticism
- Recent work
- Conclusions
- Questions
Background
Background

Opportunities:

- Cloud computing
- Big data analytics
- Cost savings
Background

Challenges:

- Complex performance model
- Cost model tradeoffs
- Heterogeneous applications
- Limited number of samples (from a large configuration space)
Prior Work

- Ernest
- Coordinate descent
- Exhaustive search
- Random search
CherryPick
CherryPick

- Uses Bayesian Optimisation to build performance models
- Finds optimal/near-optimal configurations in only a few test runs
- Uses the acquisition function to draw samples
CherryPick

Initial:

\[
\begin{align*}
\text{minimize} & \quad C(\bar{x}) = P(\bar{x}) \times T(\bar{x}) \\
\text{subject to} & \quad T(\bar{x}) \leq \mathcal{T}_{\text{max}}
\end{align*}
\]

Modified:

\[
\begin{align*}
\log \tilde{C}(\bar{x}) &= \log C(\bar{x}) + \log (1 + \varepsilon_c) \\
\text{subject to} & \quad \log T(\bar{x}) \leq \log \mathcal{T}_{\text{max}}
\end{align*}
\]
CherryPick Workflow

Step-1: Start with initial cloud configs.
Step-2: Update perf. model (re-compute confidence interval with BO)
Step-3: Select and run a new config (select next sample with the best gain estimated by BO)
Step-4: Confident that we find the best configuration?
   - No
   - Yes → End
Step-5: End
CherryPick Implementation

- Search Controller
- Cloud Monitor
- Bayesian Optimisation Engine
- Cloud Controller
Evaluation
Evaluation

- Applications: TPC-DS, TPC-H, TeraSort, SparkReg, SparkKm
- 66 cloud configurations
- Objective: reduce cost of execution under runtime constraint
- Compared with:
  - Exhaustive search
  - Coordinate Descent
  - Random Search (with a budget)
  - Ernest
Evaluation

- Metric 1: the expense to run a job with the selected configuration
- Metric 2: the expense to run all sampled configurations
- 20 independent runs
- 10th, 50th and 90th percentiles computed
Evaluation

(a) Running cost

(b) Search cost
Evaluation

CherryPick (El=10%)
- Rand-1x cost
- Rand-2x cost
- Rand-4x cost

Relative running cost

Experiment

CherryPick (El=10%)
- Ernest

Running Cost
Search Cost
Search Time
Evaluation

- Investigated parameter tuning
- Investigated performance behaviour
Evaluation

- Handling workload variation

![Graph showing handling of workload variation](image)
Criticism
“With 4x cost, random search can find similar configurations to CherryPick on the median”
• 3/4 comparison tasks are easy to beat (nothing to compare with)

• Not using available information efficiently
Recent Work
Recent Work

- PARIS
- Scout
- Arrow
- Micky
Conclusions
Conclusions

- Introduced CherryPick
- Compared to existing systems
- Presented evaluation results
- Criticism
Questions?