All about the money:
Cost modeling and optimization of cloud applications

Dr. Sebastian Baltes
empirical-software.engineering
A pot of gold at the end of the cloud rainbow?

“Successfully migrating workloads to the cloud means delivering maximum benefits with minimal effort and costs.” Gartner, 2022
Often, there’s rather an unexpectedly high bill
Still confused as hell? Get help at duckbillgroup.com
Good news for GI: A lot of (cost) optimization potential!
Personal background
My current role(s)

Principal Expert ESE
SAP SE
Walldorf, Germany

Adjunct Lecturer
University of Adelaide
Adelaide, Australia
Software Engineering Research Beyond Disciplinary and Institutionalized Boundaries
Disciplinary Boundaries of Software Engineering

1968 NATO Software Engineering Conference, Garmisch, Germany
Disciplinary Boundaries of Software Engineering

• With a traditional view emphasizing software engineering’s roots in computer and systems engineering many questions of modern software development cannot be answered.

• Examples:
  • How can we develop visual programming environments without knowledge of cognition?
  • How can we fully grasp the implications of online code reuse without understanding copyright legislation and software licenses?
  • How can we systematically compare and optimize cloud application costs across vendors and abstractions without knowledge about workload and cost modeling?
Personal Observation

• Many of the problems **relevant** in the **software industry** are rooted in software engineering but often have an **interdisciplinary** angle.

• To be able to impact industry, academia needs to provide **actionable recommendations** addressing **problems rooted in practitioners’ actual needs**.

• **Empirical research methods are essential** for identifying the above-mentioned problems (**problem space**) and corroborating recommendations/proposed solutions with empirical evidence (**solution space**).
"If you were using MDE for building your mobile app, you’d see huge quality improvements, see this paper."

“Institutionalized Boundaries

“Have you heard about things like time-to-market and quickly responding to customer feedback? We’re not building safety-critical software.”

Research

Practice
Implications for researchers:

1) Strong understanding of state of practice is essential.

2) To reach this understanding, we need to utilize diverse empirical research methods and learn from other disciplines.

3) To advance evidence-based practice, we need to invest effort into communicating findings back to practitioners.
Empirical Software Engineering

Software Engineering
Stakeholders, Processes, Tools

1. Empirically understanding problem space
2. Developing solution
3. Empirically validating solution

Empirical Research

Interdisciplinarity

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Why do companies move workloads to the cloud?
Why move to the cloud?

• **Cost transparency** and/or **cost reduction**.

• **Capacity** is the maximum *workload* a cloud layer can handle.

• **Scalability**: Ability of a cloud layer to *increase its capacity* by expanding its quantity of consumed lower-level services.

• **Elasticity**: Degree a cloud layer autonomously adapts capacity to workload over time.

(definitions by Lehrig et al. 2015)
### Top cloud initiatives by cloud usage for all organizations

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Light</th>
<th>Moderate</th>
<th>Heavy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizing our existing use of cloud (cost savings)</td>
<td>55%</td>
<td>61%</td>
<td>71%</td>
</tr>
<tr>
<td>Migrating more workloads to cloud</td>
<td>59%</td>
<td>61%</td>
<td>43%</td>
</tr>
<tr>
<td>Implementing automated policies for cloud governance</td>
<td>39%</td>
<td>44%</td>
<td>50%</td>
</tr>
<tr>
<td>Better financial reporting on cloud costs (showback/chargeback)</td>
<td>41%</td>
<td>41%</td>
<td>48%</td>
</tr>
<tr>
<td>Progressing on a cloud-first strategy</td>
<td>46%</td>
<td>43%</td>
<td>35%</td>
</tr>
<tr>
<td>Moving from on-premises to SaaS versions of software applications</td>
<td>48%</td>
<td>47%</td>
<td>25%</td>
</tr>
<tr>
<td>Expanding use of containers</td>
<td>33%</td>
<td>38%</td>
<td>41%</td>
</tr>
<tr>
<td>Managing software licenses used in the cloud</td>
<td>38%</td>
<td>37%</td>
<td>33%</td>
</tr>
</tbody>
</table>

N=750
Source: Flexera 2023 State of the Cloud Report

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Cost transparency in the cloud is a problem

Reducing BigQuery Costs: How We Fixed A $1 Million Query
by Calvin Zhou · Data Science & Engineering
Nov 3, 2022 · 3 minute read

Introducing AWS Cost Anomaly Detection (Preview)
Posted On: Sep 25, 2020

https://news.ycombinator.com/item?id=31907374
Companies moving away from the cloud...

Our cloud spend in 2022

Since we published why we’re leaving the cloud, we’ve received a lot of questions about our actual spending. We’re happy to share, both where we currently are and where we’re going.

Fernando Álvarez
SRE, Ops

...or moving their cloud applications to more traditional architectures

Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%

The move from a distributed microservices architecture to a monolith application helped achieve higher scale, resilience, and reduce costs.

Hype-driven software engineering...
Observations

• A lot of **confusion** and **hype-driven discussions/decisions**.

• Great **opportunity for research** to step in and objectify the discussion.

• Spoiler: While **(operations) cost** is considered in related disciplines, it is an **essential but often overlooked non-functional property** in software engineering research.
“The cloud” and its billing models
### Cloud Services: Who manages what?

<table>
<thead>
<tr>
<th></th>
<th>Traditional IT</th>
<th>IaaS</th>
<th>PaaS</th>
<th>SaaS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Runtime</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middleware</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virtualization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Servers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Networking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **You manage**
- **Provider manages**

[IBM Cloud](https://www.ibm.com/cloud/learn/iaas-paas-saas)

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
### The Cloud Stack

**SaaS**
- (e.g., Microsoft 365, SAP Concur, Google Docs)

**PaaS**
- (e.g., Cloud Foundry, Google App Engine, Managed K8s)

**CaaS**
- (e.g., Google Cloud Run, Azure Container Instances)

**Serverless**
- **Compute** (FaaS like AWS Lambda)
- **Storage** (Serverl. DBs like Aurora)
- **Network** (e.g., AWS API Gateway)

**IaaS**
- (e.g., AWS EC2, Azure VMs)

<table>
<thead>
<tr>
<th>Compute</th>
<th>Storage</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>(provisioned VM instances)</td>
<td>(block/file/object storage)</td>
<td>(software-defined networking)</td>
</tr>
</tbody>
</table>

**Physical Infrastructure**
- (in data centers)

Can be provisioned **declaratively** via IaC files or **interactively** via web portals.
Pricing approaches in the cloud

• **Usage-based billing:**
  (aka consumption-based billing, pay-per-use, pay-as-you-go)
  Customers pay for what they use and/or how long they use a resource (by the hour/second). Billing usually monthly.

• **Subscription-based billing:**
  (aka reserved instances)
  Customers pay a recurring fee for a period of time, flat rate regardless of usage, for a specific configuration. Discounts often available for longer commitments, e.g., 1-3 years.

• **Hybrid approaches:**
  E.g., fixed monthly rate plus usage-based component.

• **Special offers:**
  E.g., free tiers, transient/spot instances (unused capacity) offered at a discount (can be reclaimed if provider needs capacity)
Function-as-a-Service (FaaS)

- Cloud-computing service that allows to **execute code in response to events**, without managing complex infrastructure.
- “Serverless” offering

```java
public class LambdaRequestHandler
    implements RequestHandler<String, String> {
    public String handleRequest(String input, Context context) {
        context.getLogger().log("Input: " + input);
        return "Hello World - " + input;
    }
}
```

https://www.baeldung.com/java-aws-lambda

https://aws.amazon.com/blogs/architecture/
field-notes-optimize-your-java-application-for-aws-lambda-with-quarkus/

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Usage-based billing: AWS Lambda

- **Duration** a function was executed (rounded up to **ms**).
- **Price** depends on the **amount of memory** allocated to function.
- **CPU** power and other resources **proportionally allocated**.

### AWS Lambda Pricing

#### x86 Price
- First 6 Billion GB-seconds / month: $0.0000166667 for every GB-second, $0.20 per 1M requests
- Next 9 Billion GB-seconds / month: $0.000015 for every GB-second, $0.20 per 1M requests
- Over 15 Billion GB-seconds / month: $0.0000133334 for every GB-second, $0.20 per 1M requests

#### Arm Price
- First 7.5 Billion GB-seconds / month: $0.0000133334 for every GB-second, $0.20 per 1M requests
- Next 11.25 Billion GB-seconds / month: $0.0000120001 for every GB-second, $0.20 per 1M requests
- Over 18.75 Billion GB-seconds / month: $0.0000106667 for every GB-second, $0.20 per 1M requests

### Memory (MB) vs Price per 1ms

<table>
<thead>
<tr>
<th>Memory (MB)</th>
<th>Price per 1ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>$0.0000000021</td>
</tr>
<tr>
<td>512</td>
<td>$0.0000000083</td>
</tr>
<tr>
<td>1024</td>
<td>$0.0000000167</td>
</tr>
<tr>
<td>1536</td>
<td>$0.0000000250</td>
</tr>
<tr>
<td>2048</td>
<td>$0.0000000333</td>
</tr>
<tr>
<td>3072</td>
<td>$0.0000000500</td>
</tr>
<tr>
<td>4096</td>
<td>$0.0000000667</td>
</tr>
<tr>
<td>5120</td>
<td>$0.0000000833</td>
</tr>
<tr>
<td>6144</td>
<td>$0.0000001000</td>
</tr>
<tr>
<td>7168</td>
<td>$0.0000001167</td>
</tr>
<tr>
<td>8192</td>
<td>$0.0000001333</td>
</tr>
<tr>
<td>9216</td>
<td>$0.0000001500</td>
</tr>
<tr>
<td>10240</td>
<td>$0.0000001667</td>
</tr>
</tbody>
</table>

[https://aws.amazon.com/lambda/pricing/](https://aws.amazon.com/lambda/pricing/)

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Subscription-based billing: Amazon EC2

https://calculator.aws/#/addService/ec2-enhancement

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Provisioning via Web UI: Google Cloud

To create a VM instance, select one of the options:

- New VM instance
  Create a single VM instance from scratch

- New VM instance from template
  Create a single VM instance from an existing template

- New VM instance from machine image
  Create a single VM instance from an existing machine image

- Marketplace
  Deploy a ready-to-go solution onto a VM instance

Name: Instance-1

Labels

Region: us-central1 (Iowa)

Zone: us-central1-a

Pricing summary

Monthly estimate: $126.83
That's about $0.17 hourly

Pay for what you use: no upfront costs and per second billing

<table>
<thead>
<tr>
<th>Item</th>
<th>Monthly estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 vCPU + 8 GB memory</td>
<td>$125.83</td>
</tr>
<tr>
<td>10 GB balanced persistent disk</td>
<td>$1.00</td>
</tr>
<tr>
<td>Total</td>
<td>$126.83</td>
</tr>
</tbody>
</table>

Machine configuration

Machine types for common workloads, optimized for cost and flexibility

Series: C3 (Public Preview)

Powered by Intel Sapphire Rapids CPU platform

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads.

- c3-highcpu-4 (4 vCPU, 8 GB memory)
Infrastructure-as-Code (IaC): Terraform

Example Usage

```hcl
resource "google_service_account" "default" {
  account_id   = "service_account_id"
  display_name = "Service Account"
}

resource "google_compute_instance" "default" {
  name       = "test"
  machine_type = "e2-medium"
  zone       = "us-central1-a"

  tags = ["foo", "bar"]

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-11"
      labels = {
        my_label = "value"
      }
    }
  }
}
```

https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/compute_instance

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
**Goal:** Achieving the following properties for a (usually Kubernetes-based) GitOps-managed system:

1. **Declaratively** defined desired state.
2. **Versioned and immutable** desired state.
3. Software agents **automatically pull** desired state declarations from source.
4. Software agents **continuously observe** actual system state and **attempt to apply** desired state.
GitOps: ArgoCD


Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
“Great, resources are automatically provisioned after I update the IaC files!”
Cost transparency in the cloud is a problem

https://news.ycombinator.com/item?id=31907374

Tell HN: I DDoSed myself using CloudFront and Lambda Edge and got a $4.5k bill
274 points by hunksley 5 months ago | hide | past | favorite | 333 comments

@donkersgoed@hachyderm.io
@donkersgood

How a single-line bug cost us $2000 in AWS spend...
We recently refactored a Lambda Function. We extensively tested its functionality and released it into production. And everything still worked as expected. But then the billing alarm went off..

https://twitter.com/donkersgoed/status/1635244161778737152

Reducing BigQuery Costs: How We Fixed A $1 Million Query
by Calvin Zhou · Data Science & Engineering
Nov 3, 2022 · 3 minute read

https://shopify.engineering/reducing-bigquery-costs

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Mitigations: Infracost

- Supports over 1,100 Terraform resources across AWS, Azure and Google (no other IaC formats)
- Focuses rather on guardrails and policies than on supporting architecture decision making (e.g., “With certain workload assumptions, when will the decision to use serverless backfire?”)

https://github.com/infracost/infracost

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Mitigations: AWS Lambda Power Tuning

• AWS Lambda Power Tuning helps **optimize Lambda functions for cost and/or performance** in a data-driven way.

• **Invokes a given Lambda function with multiple configuration**, then **analyzes execution logs**, suggests best configuration minimizing cost and/or maximizing performance.

• Limitations:
  • "Please note that the input function will be executed in your AWS account."
  • Focus on on individual functions (local vs. global optima)
Mitigations: AWS Lambda Power Tuning

https://github.com/alexcasalboni/aws-lambda-power-tuning

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Mitigation: OpenCost

- Vendor-neutral open source project for **measuring and allocating infrastructure and container costs in real time.**
- "**OpenCost shines a light into the black box of Kubernetes spend.**"
- "**Real-time cost allocation, broken down by Kubernetes concepts down to the container level.**"

→ More fine-grained reporting for K8s, reduce reporting delay.
Infrastructure-from-Code (IfC)

• “[…] logical evolution of cloud. Instead of writing low-level, control-plane specific instructions, IfC infers requirements from application logic and provisions the optimal cloud infrastructure.” - infrastructurefromcode.com

• “Programming languages and cloud infrastructure will converge in a single paradigm: where all resources required will be automatically provisioned, and optimized by the environment that runs it.” - Shawn “swyx” Wang
Unleash developer productivity with infrastructure from code

Jeremy Daly (he/him)
CEO & Founder
Ampt Web Services, Inc.

AWS re:Invent 2022 - Unleash developer productivity with infrastructure from code [COM301]

https://www.youtube.com/watch?v=RmwKBPCo7o4

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Infrastructure-from-Code (IfC)

For example, the following sample IfC implementation...

```javascript
import { api, data, events } from '@some-ifc-sdk'

api.post('/users', async (req, res) => {
  const { email, name } = req.body;
  const newUser = await data.set(`user:${email}`, { email, name });
  res.send({ user: newUser });
});

data.on('created:user', (item) => {
  console.log(`New user created!`);
  events.publish('user.created', { after: '1 day' }, item);
});

events.on('user.created', (event) => {
  console.log('user.created event received!');
  // Send a follow up email, call an API, etc.
})();
```

Automatically provisions and configures Amazon API Gateway

...when deployed on AWS, would automatically provision and configure the following resources...

...including mapping IAM permissions between services.

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Cost-aware architecture decision making for cloud applications
Cost-aware cloud architecture decisions

- Cloud-native developers frequently **modify** IaC configs within editors/IDEs.
- **Cost monitoring/estimation** tools available in web portals, mainly considered **downstream** task.
- Cost considerations need to be moved closer to **software architecture decision making**.
- Related topic: **Cloud resource demand management**.
Cost-aware cloud architecture decisions

### A vendor-agnostic cost model for predicting compute and storage costs helping to reason about tradeoffs.
Cost-aware cloud architecture decisions

Potential questions:

- For a given expected workload, is it cheaper to utilize usage-based serverless offering or a subscription-based IaaS offering?
- Is a specific FaaS offering cheaper at AWS compared to Azure for a given workload?
Minimal information required for a cost model

- Description/operationalization of **modeled resources**, e.g.,
  - Compute
  - Storage
  - Network

- Description of a **workload**
  - Database: Query, Dataset
  - Serverless: Function inputs (e.g., JSON), abstract description of runtime properties of function(s)
  - PaaS/CaaS/IaaS offerings: Much more complicated

- **Evolution of the workload** over time
  - Short-term peaks
  - Long-term development
The company perspective

• **Scenario:** A company wants to offer a novel database system aaS.

• Given a set of **benchmark workloads**, how to determine which cloud provider’s IaaS setup is **cheaper in which scenarios without executing** (all of) the workloads?

• Once the system is live: When optimizing queries, there might be cases where a slight decrease in **performance** leads to significant **cost savings**.

• Input for cost model: query and dataset properties.
The research perspective
Software Engineering (SE)

• SE research focuses on effort estimation rather than monitoring/modeling/optimizing operation cost.
• However, since DevOps emerged, operations-related costs moved closer to the daily work of developers.
Services/Cloud Computing

2022 IEEE International Conference on Cloud Engineering (IC2E)

Streaming vs. Functions:
A Cost Perspective on Cloud Event Processing

Tobias Pfandzelter, Sören Henning, Trever Schirmer, Wilhelm Hasselbring, David Bermbach

1TU Berlin & ECDF, Mobile Cloud Computing Research Group
{eps,db}@mcc.tu-berlin.de

2Kiel University, Software Engineering Group
{soeren.henning,hasselbring}@email.uni-kiel.de

Using Parametric Models to Represent Private Cloud Workloads

Richard Wolski, Member, IEEE, and John Brevik

IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 7, NO. 4, OCTOBER-DECEMBER 2014

2009 IEEE International Conference on Cloud Computing

The Method and Tool of Cost Analysis for Cloud Computing

Xinhui Li, Ying Li, Tiancheng Liu, Jie Qiu, Fengchun Wang
IBM China Research Lab, BJ, 100193, China
{lixinhui, liying, liutc, qijfje, wangfc}@cn.ibm.com

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Example: Streaming vs. Functions

A Cost Perspective on Cloud Event Processing

Tobias Pfandzelter†, Sören Henning‡, Trever Schirmer†, Wilhelm Hasselbring‡, David Bermbach‡
†TU Berlin & ECDF Mobile Cloud Computing Research Group
{tp,ts,db}@mcc.tu-berlin.de
‡Kiel University, Software Engineering Group
{soeren.henning,hasselbring}@email.uni-kiel.de

- **UC1**: stateless storage use-case
- **UC1**: stateful sliding window aggregation use-case
Towards Cost-Optimal Query Processing in the Cloud

Viktor Leis  
viktor.leis@fau.de  
Friedrich-Alexander-Universität Erlangen-Nürnberg

Maximilian Kuschewski  
maximilian.kuschewski@fau.de  
Friedrich-Alexander-Universität Erlangen-Nürnberg

Figure 5: Prototype measurement vs. prediction on a 100 GB aggregation query

Sebastian Baltes - All about the money: Cost modeling and optimization of cloud applications
Potential next steps

Support engineers and organizations in choosing suitable cloud architectures, shifting cost transparency left using appropriate tooling.

Mining GitHub for typical IaC/IfC setups, retrieving workloads characteristics from observability data.

Existing work on cost modeling in other research communities.

Software Engineering Stakeholders, Processes, and Tools

Empirical Research

Interdisciplinarity
Takeaways

• **Cost transparency is a problem** for cloud applications.

• Research mainly focused on cost-optimizing **database** or **serverless** workflows.

• More research needed on **cost models** allowing reasoning between cloud layers and vendors, particularly on the **long run** ("lock-in").

• Cost transparency needs to be **integrated into tools** that modern software/platform engineers use ("shifting left").

• Cost **optimization** needs to consider other **non-functional requirements** such as performance, scalability, elasticity.
Interested in collaborating? Please reach out!

Dr. Sebastian Baltes

cost-aware cloud architecture

empirical-software.engineering