
Inhalt

Vorwort	IX
1 Einführung	1
Eine kurze Geschichte der System-Entwicklung	1
Eine kurze Geschichte von Mustern in der Software-Entwicklung	3
Formalisierung der algorithmischen Programmierung	3
Muster für die objektorientierte Programmierung	3
Der Aufstieg von Open-Source-Software.	4
Der Wert von Mustern, Praktiken und Komponenten	4
Auf den Schultern von Riesen stehen	5
Eine gemeinsame Sprache für Gespräche über unsere Praktiken ...	5
Gemeinsame Komponenten für einfache Wiederverwendung	6
Zusammenfassung	7

Teil I: Single-Node-Muster

Motivation	9
Zusammenfassung	11
2 Sidecar	13
Ein Beispiel-Sidecar: HTTPS für einen alten Service	13
Dynamische Konfiguration mit Sidecars	14
Modulare Anwendungs-Container	16
Aus der Praxis: Den topz-Container deployen	17
Mit Sidecars eine einfache PaaS bauen	17
Modularität und Wiederverwendbarkeit durch Sidecars	19
Container parametrisieren.	19
Die API jedes Containers definieren	20
Dokumentieren Sie Ihre Container	21
Zusammenfassung	22

3	Ambassador	23
	Mit einem Ambassador einen Service per Sharding aufteilen	24
	Aus der Praxis: Einen Sharded Redis implementieren	25
	Einen Ambassador zum Service Brokering einsetzen	27
	Mit einem Ambassador experimentieren oder Requests splitten	28
	Aus der Praxis: 10%-Experimente umsetzen	29
4	Adapter	33
	Monitoring	34
	Aus der Praxis: Prometheus für das Monitoring einsetzen	35
	Logging	36
	Aus der Praxis: Verschiedene Logging-Formate mit fluentd normalisieren	37
	Einen Health-Monitor ergänzen	38
	Aus der Praxis: Ausführliches Health Monitoring für MySQL	39

Teil II: Serving-Muster

	Einführung in Microservices	43
5	Replizierte lastverteilte Services	47
	Zustandslose Services	47
	Readiness-Prüfungen für das Load Balancing	48
	Aus der Praxis: Einen replizierten Service in Kubernetes erstellen ..	49
	Services mit Session Tracking	50
	Replizierte Services auf Anwendungsebene	51
	Einziehen einer Caching-Schicht	51
	Ihren Cache deployen	52
	Aus der Praxis: Die Caching-Schicht deployen	53
	Die Cache-Schicht erweitern	55
	Rate Limiting und Denial-of-Service-Verteidigung	55
	SSL Termination	56
	Aus der Praxis: nginx und SSL Termination deployen	57
	Zusammenfassung	59
6	Sharded Services	61
	Sharded Caching	61
	Warum Sie einen Sharded Cache brauchen könnten	62
	Die Rolle des Cache für die System-Performance	63

Replizierte Sharded Caches	64
Aus der Praxis: Einen Ambassador und Memcache für einen Sharded Cache deployen	65
Sharding-Funktionen	69
Auswahl eines Schlüssels	70
Konsistente Hashing-Funktionen	71
Aus der Praxis: Einen konsistenten HTTP Sharding Proxy bauen . .	71
Repliziertes Sharded Serving	72
Hot Sharding Systems	73
7 Scatter/Gather	75
Scatter/Gather mit Root-Verteilung	76
Aus der Praxis: Verteilte Dokumentensuche	76
Scatter/Gather mit Leaf Sharding	78
Aus der Praxis: Sharded Dokumentensuche	78
Die richtige Anzahl an Blättern wählen	79
Scatter/Gather für mehr Zuverlässigkeit und bessere Skalierbarkeit replizieren	81
8 Funktionen und ereignisgesteuerte Verarbeitung	83
Herausfinden, ob FaaS sinnvoll ist	84
Die Vorteile von FaaS	84
Die Herausforderungen von FaaS	84
Der Bedarf für Hintergrundprozesse	85
Der Bedarf für das Halten von Daten im Speicher.	86
Die Kosten einer dauerhaften Request-basierten Verarbeitung.	86
Muster für FaaS	87
Das Decorator-Muster.	87
Aus der Praxis: Ein Request-Defaulting vor die Request-Verarbeitung schalten	88
Umgang mit Ereignissen	89
Aus der Praxis: Zwei-Faktor-Authentifizierung implementieren. . . .	90
Ereignisbasierte Pipelines	91
Aus der Praxis: Eine Pipeline für das Registrieren neuer Benutzer implementieren.	92
9 Ownership Election	95
Herausfinden, ob Sie überhaupt eine Master Election benötigen	96
Grundlagen der Master Election	97
Aus der Praxis: etcd deployen	99
Locks implementieren	100

Aus der Praxis: Locks in etcd implementieren	103
Ownership implementieren	103
Aus der Praxis: Leases in etcd implementieren	104
Parallele Datenverarbeitung	105

Teil III: Batch-Verarbeitung

10 Work-Queue-Systeme	111
Ein generisches Work-Queue-System	111
Die Quell-Container-Schnittstelle	112
Die Worker-Container-Schnittstelle	114
Die Infrastruktur für die gemeinsame Work Queue	116
Aus der Praxis: Einen Video-Thumbnailer implementieren	118
Dynamisches Skalieren der Worker	119
Das Multi-Worker-Muster	121
11 Ereignisgesteuerte Batch-Verarbeitung	123
Muster für die ereignisgesteuerte Verarbeitung	124
Copier	124
Filter	125
Splitter	126
Sharder	127
Merger	128
Aus der Praxis: Einen ereignisgesteuerten Flow für das Registrieren neuer Benutzer bauen	129
Publisher/Subscriber-Infrastruktur	131
Aus der Praxis: Kafka deployen	132
12 Koordinierte Batch-Verarbeitung	135
Join (oder Barrier-Synchronization)	136
Reduce	137
Aus der Praxis: Count	138
Sum	139
Histogramm	139
Aus der Praxis: Eine Pipeline zum Taggen und Bearbeiten von Bildern	140
13 Zusammenfassung: Ein Neuanfang?	143
Index	145