# 賀恋棚 (が れんほう)



Java バックエンド開発エンジニア
2026 年卒。Alibaba Cloud 専有クラウドのセキュリティ製品でバックエンド実習を経験。
Java 業務開発、タスク配信、Redis 並行処理と本番障害調査を重視。
重慶
採用担当の方は、関心をお持ちいただけましたらメールでご連絡ください。·メール· lysf15520112973@163.com ·GitHub ·DavidHLP ·知識庫

## 主な強み

- **タスクの信頼性**：UltiCode で Outbox、Redis Streams、SQL CAS により配信漏れ・重複実行・再判定競合に対応。実 MySQL とブラウザー/Judge の業務検証を CI に組み込む。
- **キャッシュと並行処理**：ResiCache で同一キーの要求集約、分散ロード調整、バージョン検査、ロールバック可能な移行を実装。実 Redis/Cluster と並行テストで障害時の動作を検証。
- **OSS への貢献**：codebase-memory-mcp のクライアント連携と Linux インストーラー互換性修正の 2 件がマージ済み。インストール回帰とユーザーデータ保持も検証。

## 実習経験

**博彦科技（Beyondsoft）**
2026.01 - 2026.06
バックエンド開発インターン｜ Alibaba Cloud 専有クラウド・Aegis ホストセキュリティ

- **リソース連携**：Alibaba Cloud 専有クラウドの Aegis セキュリティ製品のバックエンド開発に参加。プラットフォーム/ECS/BMS/BMCP と連携し、GPU 情報の取り込み・集約を独力で実装して統計・絞り込み・関連検索を支援。
- **AI 資産識別**：機能開発に参加し、OpenClaw/Ollama のテスト用コンテナーを独力で構築。コンテナー状態、GPU 関連、脆弱性データから資産状態・関連付け・リスク表示を実装。
- **検索最適化と障害修正**：SQL 実行計画、インデックス、条件、対象フィールドを調整し、一覧/統計検索の時間を短縮。Arthas `watch` / `trace` / `stack` で経路・時間・例外・引数を調べ、不具合修正と回帰検証を独力で実施。
- **イメージ信頼チェーン調査**：containerd による Harbor プライベートイメージ取得時の TLS 障害を特定。CA、`certs.d`、`hosts.toml`、`imagePullSecrets`、ノードキャッシュを調べ、診断スクリプトとチェックリストを整備。

## プロジェクト経験

**オンラインプログラミング評価システム UltiCode GitHub**
2025.10 - 現在
技術：Java 17 / Spring Boot 4 / Spring Security / MyBatis-Plus / MySQL / Redis Streams / Dubbo / Flyway / Docker
練習、コンテスト、コード提出、サンドボックスでの判定、管理機能を備えた個人開発の OSS。Submission サービスが提出データを保持し、独立した Judge Worker が判定を実行。タスク配信の失敗、重複実行、再判定結果の競合を重点的に扱う。

- **確実な配信と障害回収**：提出と Outbox を同一トランザクションで保存し、配信状態更新とバックオフ再試行を実装。Lua で重複排除と Redis Streams 追加を原子化し、マーカーだけが残る障害に対応。`XPENDING` / `XCLAIM` で未確認タスクを回収し、容量・回数制限とデッドレターで無効な再配信を抑制。
- **再判定結果の競合制御**：`generation` + `attemptId` で有効な実行を識別し、リースとハートビートによる延長を併用。結果の書き戻しでは両条件を検査する SQL CAS により、期限切れの実行が新しい判定結果を上書きするのを防ぐ。
- **信頼できないコードの実行制約**：Docker のネットワーク無効化、非 root 実行、読み取り専用のルートファイルシステム、Capabilities の削減、Seccomp を設定。CPU・メモリー・PID・ファイルハンドルの上限と組み合わせ、コードの権限とリソース使用量を制限。
- **リフレッシュトークンの再利用防止**：SHA-256 ハッシュのみを DB に保存。用途・所有者・有効期限を検査し、条件付き更新で旧トークンの消費を一度に制限。同一トランザクション内で新トークンを発行し、同時更新の競合に対応。
- **自動受け入れ検証とリリース検査**：実 MySQL の新規導入・アップグレード移行とブラウザー・Judge の業務検証を CI に組み込み、ログイン、AC/WA、再判定、順位表を確認。9 イメージの署名、SBOM、SLSA 来歴、タグのダイジェストを照合し、不完全・競合する成果物を拒否。検証と本番デプロイは区別。

**ResiCache キャッシュ保護ミドルウェア GitHub**
2025.01 - 現在
技術（現開発系列）：Java 21 / Spring Boot 4 / Spring Cache / Redis / Redisson / Micrometer / Testcontainers
拡張アノテーションと責任の連鎖を組み合わせた、個人開発の Spring Cache 保護拡張。同一キーのリクエスト集約、分散環境でのロード調整、バージョン付き早期失効、安全なデシリアライズを重点的に実装。

- **同一キーの並行処理と非同期重複排除**：`ConcurrentHashMap` + `CompletableFuture` で JVM 要求を集約。

Leader が分散ロック内で再確認・ロードし、Follower は結果を共有。キー別 in-flight Future、有界キュー、`CallerRunsPolicy` で滞留と完了/登録時の後処理競合に対応。
- **階層型 Bloom フィルター**：ローカルのヒット時はリモート検査を省略。ミス時は Redis を確認して反映し、ローカルのミスだけで有効なロードを拒否しない。
- **バージョン付き早期失効**：Redis Lua でキャッシュ値のバージョンを原子的に照合して TTL を短縮し、古い非同期タスクが新しい値の有効期限を変更するのを防ぐ。失効後のリクエストがロードを開始し、更新・削除との競合に対応。
- **デシリアライズ保護**：Jackson のストリーミングで型許可リストを検査してからバージョン付き形式を解析。悪意ある型・メタデータ往復を回帰検証し、検査時に JSON ツリー全体を構築しない。
- **失敗時の挙動と検証**：業務ロードの例外とキャッシュ書き戻しの失敗を区別し、ロード済みなら書き戻しに失敗しても業務値を返す。並行テストと実 Redis の統合テストで、結果共有、例外伝播、新しい値の TTL 保護、削除後に値が復活しないことを確認。
- **クラスターと移行の安全性**：実際の 3 マスター Redis Cluster でキャッシュキーとロックキーの同一スロット配置を検証。シリアライズ形式のシャドー読み取り、二重書き込み、切り替え、ロールバックを実装し、TTL を保持。CAS で並行更新された値の上書きを拒否し、部分失敗後の再開に対応。

## 専門スキル

### Java バックエンドとデータ処理

- Java、Spring Boot、Spring MVC、Spring Security、MyBatis-Plus に習熟し、API 設計、認証・認可、例外処理の実践を持つ。
- MySQL のインデックス設計、実行計画分析、スロークエリの最適化に習熟。Redis のキャッシュ保護、分散ロック、並行ロードの実践経験。
- Dubbo、Redis Streams、Outbox の実装経験があり、トランザクション境界、重複消費、再試行、結果整合性を理解。

### 障害診断とエンジニアリング検証

- Arthas、Linux のプロセス情報、コンテナーログ、SQL 実行計画を組み合わせ、問題の特定・修正・回帰検証ができる。
- Docker イメージ、ネットワーク、リソース制限、基本的な隔離機構に習熟。containerd、Harbor、プライベートイメージの TLS 障害診断経験。
- Git、Maven、Docker Compose、Flyway、Micrometer に習熟。Testcontainers と並行テストで障害と競合を検証。

## 学歴・受賞

### 学歴

**洛陽師範学院**
データ科学とビッグデータ技術・学士課程
GPA: 3.6 / 4.0 (2021.09 - 2026.07)

### 受賞

- 2025 藍橋杯 Java ソフトウェア設計 B 組・**全国三等賞**
- 2024 テディ杯データ分析技能競技・**全国三等賞**
- 2023 全国大学コンピューター能力挑戦・**全国三等賞**

## OSS 貢献と技術記録

### OSS 貢献

**codebase-memory-mcp**：Oh My Pi クライアント接続、MCP 登録、インストール・アンインストールと利用者設定の保護に貢献。Linux の信頼できるシステムパスとの互換性を修正し、回帰テストを追加。

### 代表的な技術知識頁

- Java 性能障害：現象から証拠への最小判断木：プロセス、スレッドスタック、GC、Arthas の観測をつなぎ、影響の抑制・証拠保全・問題特定・復旧の手順を整理。
- UltiCode Outbox と Redis Streams：判定タスクの配送を復旧可能な状態にする：トランザクション Outbox、原子的な配送、再試行、デッドレターと未確認メッセージの回収を整理。