

Формализованный выбор open-source СУБД для импортозамещения MES-систем: архитектурный подход и критерии промышленной эксплуатации

Д. Е. Усков, И. А. Гурин, В. В. Лавров

Аннотация — В статье рассмотрена задача импортозамещения подсистемы хранения данных MES-уровня промышленного предприятия за счет перехода от проприетарной СУБД к open-source решению. Обоснована актуальность такого перехода в условиях роста объемов технологических данных, повышения требований к отказоустойчивости, наблюдаемости и интеграции с современными инструментами промышленной аналитики. Предложен формализованный подход к выбору СУБД, основанный на совокупности технических и эксплуатационных критериев: транзакционная надежность, отказоустойчивость, резервное копирование и восстановление, масштабируемость, эффективность работы с временными рядами, управляемость, совместимость с корпоративной ИТ-архитектурой и предсказуемость поведения под нагрузкой. На примере PostgreSQL показано, что импортозамещение не должно сводиться к механической замене программного продукта; оно требует пересмотра архитектурных принципов, выделения контуров разработки, тестирования и промышленной эксплуатации, внедрения репликации, WAL-архивирования, секционирования данных и средств мониторинга. Практическая значимость работы состоит в возможности применения предложенных критериев при модернизации действующих MES-систем и проектировании новых промышленных информационных платформ на основе открытых технологий.

Ключевые слова — MES, СУБД, PostgreSQL, open-source, импортозамещение, промышленная информационная система, временные ряды, репликация, резервное копирование, отказоустойчивость, наблюдаемость, промышленная аналитика.

I. ВВЕДЕНИЕ

Системы уровня MES (Manufacturing Execution System) являются одним из центральных элементов

цифровой инфраструктуры промышленного предприятия. Они обеспечивают регистрацию производственных событий, сбор технологических параметров, хранение истории выполнения операций, подготовку данных для диспетчеризации, отчетности, анализа эффективности и последующего построения аналитических или информационно-моделирующих систем. Поэтому подсистема хранения данных MES-уровня фактически определяет надежность, производительность и технологическую ценность всей информационной системы [1-3].

В течение длительного времени такие подсистемы строились преимущественно на основе коммерческих СУБД, прежде всего Oracle Database и Microsoft SQL Server. Эти продукты обладают развитым функционалом, зрелыми средствами администрирования и большим опытом промышленного применения [4, 5]. Однако в современных условиях их использование сопровождается рядом ограничений: высокой совокупной стоимостью владения, зависимостью от вендора, ограниченной гибкостью масштабирования, сложностью интеграции с open-source инструментами обработки данных, машинного обучения и наблюдаемости [8, 9, 16].

Для промышленных предприятий указанные ограничения имеют не только экономический, но и технологический характер. MES-система работает с большими массивами временных рядов, событиями производственного цикла, результатами лабораторного контроля и данными смежных АСУ ТП. Рост частоты измерений, расширение состава контролируемых параметров, развитие промышленной аналитики и цифровых двойников повышают требования к доступности, скорости обработки и воспроизводимости данных. В этих условиях задача выбора СУБД становится архитектурной задачей, а не только вопросом лицензирования [1, 9, 12].

Актуальность работы обусловлена необходимостью создания воспроизводимого подхода к выбору open-source СУБД для MES-уровня, который учитывает специфику промышленной эксплуатации. Ошибка при выборе технологии может привести к снижению надежности, росту затрат на сопровождение, потере управляемости и усложнению развития системы. Следовательно, предприятие должно иметь не набор

Статья получена 18 июня 2026.

Усков Даниил Евгеньевич – аспирант Уральского Федерального Университета, Института Новых Материалов и Технологий, кафедры «Теплофизики и информатики в металлургии». (электронная почта: u_danila@mail.ru)

Гурин Иван Александрович – доцент, кандидат технических наук Уральского Федерального Университета, Института Новых Материалов и Технологий, кафедры «Теплофизики и информатики в металлургии». (электронная почта: ivan.gurin@urfu.ru)

Лавров Владислав Васильевич – профессор, доктор технических наук, доцент Уральского Федерального Университета, Института Новых Материалов и Технологий, кафедры «Теплофизики и информатики в металлургии». (электронная почта: v.v.lavrov@urfu.ru)

субъективных аргументов в пользу популярного продукта, а формализованные критерии оценки его пригодности [8, 9].

Цель работы заключалась в создании и обосновании формализованного подхода к выбору СУБД при импортозамещении подсистемы хранения данных MES-уровня, а также в практической демонстрации применения такого подхода на примере перехода к PostgreSQL.

Для достижения цели были решены следующие задачи: выявление ограничений исходной проприетарной архитектуры; формулировка требований к СУБД промышленного MES-уровня; определение критериев выбора open-source решения; проектирование целевой архитектуры хранения данных на базе PostgreSQL; выполнение сравнительного анализа PostgreSQL, MySQL и ClickHouse с позиции промышленной применимости; определение практической значимости предложенного подхода для ИТ-отрасли и промышленной цифровизации.

II. ИСХОДНАЯ АРХИТЕКТУРА И ПРИЧИНЫ ЕЕ МОДЕРНИЗАЦИИ

До миграции подсистема хранения данных MES-уровня имела централизованный характер. В качестве основных компонентов использовались коммерческие СУБД и связанные с ними инструменты обработки данных. В отдельных контурах применялись Microsoft SQL Server, Oracle Database, а также вспомогательные средства хранения и подготовки данных, включая Excel и Microsoft Access. Такое построение типично для корпоративных информационных систем, развивавшихся эволюционно: первоначально оно обеспечивает приемлемую управляемость, но при росте объемов данных и усложнении требований начинает ограничивать развитие системы [3-5].

Особенностью исходной архитектуры было то, что СУБД выполняла не только функцию долговременного хранения, но и значительную часть вычислительной и прикладной логики. Внутри базы данных размещались хранимые процедуры, процедуры агрегации, элементы бизнес-логики, механизмы подготовки данных для отчетности. В результате формировалась высокая связанность схемы данных, SQL-кода и прикладных модулей. Такая связанность усложняла сопровождение, повышала риск ошибок при изменениях и снижала возможность независимого развития отдельных компонентов [8, 9].

Ключевой тип данных MES-системы представлен временными рядами технологических параметров. В исходной реализации они хранились в реляционных таблицах, а ускорение выборок обеспечивалось индексами и SQL-агрегациями. При умеренных объемах данных такой подход является работоспособным, однако при росте частоты измерений и глубины исторического архива возникают проблемы: увеличивается время выполнения запросов, усложняется обслуживание индексов, возрастает нагрузка на центральный узел, а работа с высокочастотными

данными становится менее предсказуемой [2, 9].

Загрузка и преобразование данных выполнялись преимущественно средствами, связанными с проприетарным стеком, в частности Microsoft SQL Server Integration Services (SSIS) и Oracle Data Integrator (ODI). Эти инструменты позволяют решать задачи интеграции, но усиливают зависимость от конкретного вендора, усложняют перенос логики и ограничивают гибкость изменения потоков данных [6, 7, 10]. Для MES-системы, которая должна быстро адаптироваться к новым источникам информации, новым форматам и новым аналитическим задачам, такая зависимость становится существенным ограничением.

Масштабирование исходной системы осуществлялось преимущественно вертикально: за счет увеличения ресурсов сервера, расширения дисковой подсистемы, добавления памяти и оптимизации запросов. Однако вертикальное масштабирование имеет объективные пределы. Кроме того, в проприетарных СУБД рост вычислительной мощности и применение расширенных механизмов отказоустойчивости часто сопровождаются дополнительными лицензионными затратами. Это повышает стоимость развития системы и снижает экономическую гибкость ИТ-инфраструктуры [4, 5, 9]. Таким образом, необходимость модернизации определялась не только задачей импортозамещения, но и совокупностью архитектурных причин: ростом объемов технологических данных, потребностью в горизонтальном распределении нагрузки, необходимостью повышения наблюдаемости, снижением зависимости от поставщика и переходом к более модульной архитектуре промышленной информационной системы [8, 9, 16].

III. ТРЕБОВАНИЯ К СУБД MES-УРОВНЯ

СУБД, используемая в MES-системе, должна оцениваться не как универсальное хранилище данных, а как критически важный компонент промышленной ИТ-инфраструктуры. Ее отказ, деградация производительности или потеря данных могут повлиять на корректность производственной отчетности, оперативное управление, анализ простоев, контроль качества и принятие технологических решений [1, 3].

С учетом этого требования к СУБД целесообразно разделить на технические и эксплуатационные. Технические требования характеризуют внутренние возможности системы, а эксплуатационные - способность устойчиво работать в промышленной среде, сопровождаться ИТ-службой и интегрироваться с существующей инфраструктурой предприятия [8, 9].

К основным требованиям относятся:

- 1) транзакционная надежность и поддержка свойств ACID (Atomicity, Consistency, Isolation, Durability) при одновременной работе большого числа пользователей, сервисов и интеграционных потоков [2, 9, 12, 13];
- 2) отказоустойчивость, включая репликацию, наличие резервных узлов и возможность переключения при отказе [12, 13];
- 3) резервное копирование и восстановление, в том

числе возможность восстановления состояния базы данных на заданный момент времени [12];

4) масштабируемость, позволяющая разделять операции чтения и записи, выносить аналитические запросы на реплики и расширять инфраструктуру без полной перестройки системы [9, 12];

5) эффективная работа с временными рядами, включая секционирование по времени, оптимизацию индексов, выполнение агрегирующих запросов и управляемое хранение исторических данных [12, 14, 16];

6) наблюдаемость и администрируемость, позволяющие контролировать метрики состояния, выявлять узкие места, анализировать производительность и выполнять профилактическое обслуживание [16];

7) совместимость с корпоративной архитектурой: стандартные интерфейсы, поддержка драйверов, интеграция с ETL/ELT-инструментами (Extract, Transform, Load / Extract, Load, Transform), аналитическими платформами и средствами мониторинга [10, 15, 16];

8) предсказуемость поведения под нагрузкой, поскольку для промышленной эксплуатации важна не только максимальная производительность, но и стабильность времени отклика [8, 9].

При таком подходе выбор СУБД становится задачей многокритериальной инженерной оценки. На первый план выходит не отдельная характеристика, например, скорость аналитических запросов, а сбалансированное соответствие требованиям MES-уровня. Именно это отличает промышленный выбор технологии от выбора инструмента для локального проекта или экспериментального прототипа.

IV. НОВИЗНА ПРЕДЛАГАЕМОГО ПОДХОДА

Новизна рассматриваемого подхода состоит в переходе от интуитивного или административно мотивированного выбора СУБД к формализованной оценке ее промышленной пригодности. В практике импортозамещения часто используется упрощенная

Таблица 1 – Критерии выбора СУБД для MES-уровня

логика: проприетарный продукт заменяется на open-source аналог, обладающий сходным набором функций. Однако для MES-уровня такая логика недостаточна, поскольку надежность системы определяется не только функциональным перечнем возможностей, но и архитектурой их применения [8, 9].

Предлагаемый подход включает три взаимосвязанных элемента. Первый элемент - выделение критериев, непосредственно связанных с эксплуатацией промышленной системы: отказоустойчивость, восстановление, транзакционная целостность, масштабируемость, работа с временными рядами, наблюдаемость и интеграционная совместимость (таблица 1). Второй элемент - оценка альтернативных СУБД не по популярности, а по степени соответствия этим критериям. Третий элемент - связь выбора СУБД с целевой архитектурой MES-системы, включая разделение ответственности между базой данных, интеграционным слоем, аналитическими сервисами и средствами мониторинга [9, 12–16].

Разработана риск-ориентированная многокритериальная модель выбора open-source СУБД для MES-систем, учитывающая вес критериев, экспертные оценки и уровень неопределенности принимаемого решения.

Таким образом, научно-практическая новизна заключается не в утверждении, что PostgreSQL может заменить коммерческую СУБД, а в обосновании методики, позволяющей определить, при каких архитектурных условиях open-source СУБД становится пригодной для промышленного MES-уровня. Это особенно важно для предприятий, где импортозамещение должно одновременно решать задачи технологической независимости, надежности и развития цифровой аналитики.

Критерий	Содержание критерия	Значение для MES-уровня
Отказоустойчивость	Репликация, резервные узлы, переключение при отказе	Снижение риска простоев и потери доступности производственных данных
Восстановление данных	Резервное копирование, WAL (Write-Ahead Logging), восстановление на момент времени	Сохранение исторических данных и воспроизводимость состояния системы
Транзакционная надежность	Поддержка ACID, согласованность данных	Корректная работа при параллельных производственных и аналитических операциях
Масштабируемость	Разделение чтения и записи, реплики, секционирование	Поддержка роста объема данных и числа запросов
Временные ряды	Хранение, индексация и агрегация данных по времени	Эффективная обработка технологических параметров и архивов
Наблюдаемость	Метрики, журналы, диагностика, мониторинг	Предупреждение отказов и управляемое сопровождение системы
Совместимость	Стандартные протоколы, драйверы, интеграция с ETL/ELT и BI (Business Intelligence)	Снижение затрат на внедрение и интеграцию в ИТ-ландшафт предприятия

А. Формализованная модель оценки пригодности СУБД

Предлагаемая методика основана на риск-скорректированной многокритериальной оценке. Для каждого варианта СУБД v и каждого критерия k эксперт задает балл соответствия X_{vk} в диапазоне от 0 до 10 и уровень уверенности P_{vk} в диапазоне от 0 до 100 %. Дополнительно для критерия задаются вес W_k , порог T_k и признак критичности. Наличие уровня уверенности принципиально важно: высокий экспертный балл при низком подтверждении не должен давать такое же основание для промышленного выбора, как балл, подтвержденный опытом эксплуатации, документацией или нагрузочным испытанием.

Нормированный вес критерия рассчитывался как:

$$w_k = \frac{W_k}{\sum W_k} \quad (1)$$

Уровень уверенности приводился к диапазону:

$$p_{vk} = \frac{P_{vk}}{100} \quad (2)$$

Риск-скорректированная оценка определялась выражением:

$$E_{vk} = X_{vk} \cdot p_{vk} \quad (3)$$

Интегральный показатель пригодности технологии вычислялся по формуле:

$$S_v = 10 \sum (w_k E_{vk}) \quad (4)$$

где S_v характеризует степень соответствия технологии требованиям промышленной эксплуатации.

$$U_v = 100 \sum (w_k (1 - p_{vk})) \quad (5)$$

Дополнительно рассчитывался индекс неопределенности экспертной оценки, позволяющий учитывать степень подтвержденности исходных данных и уровень доверия к результатам выбора.

Если критерий является критичным и $X_{vk} < T_k$, фиксируется критическое нарушение. Если $E_{vk} < T_k$, фиксируется зона риска. Такой механизм предотвращает ситуацию, когда высокий средний балл скрывает недопустимое несоответствие по восстановлению данных, транзакционной целостности или отказоустойчивости.

Следует подчеркнуть, что индекс S не является прямым показателем производительности СУБД. Он характеризует расчетную промышленную пригодность в заданной архитектурной роли и при заданном наборе экспертных предположений. Окончательное решение должно подтверждаться пилотной эксплуатацией, нагрузочным тестированием, проверкой миграции, восстановлением данных, аудитом информационной безопасности и расчетом совокупной стоимости владения. Графическая демонстрация структуры представлена на рисунке 1.



Рисунок 1 – Структура риск-ориентированной экспертной оценки open-source СУБД для MES-систем

V. ЦЕЛЕВАЯ АРХИТЕКТУРА НА ОСНОВЕ POSTGRESQL

В соответствии с результатами формализованной оценки PostgreSQL получила наибольшее значение интегрального показателя пригодности среди рассмотренных open-source решений. Поэтому именно данная СУБД была выбрана в качестве основы целевой архитектуры. Она обеспечивает сбалансированное сочетание транзакционной надежности, развитых средств репликации, резервного копирования, секционирования и интеграции с современными инструментами обработки данных [12, 17].

Принципиальным архитектурным изменением является пересмотр роли СУБД. В исходной системе база данных была одновременно хранилищем, вычислительным ядром и носителем значительной части прикладной логики. В целевой архитектуре СУБД должна выполнять прежде всего функции надежного хранения, обеспечения транзакционной целостности, доступа к данным и поддержки их восстановления. Вычислительная и трансформационная логика выносятся в прикладные сервисы, интеграционные процессы и аналитические компоненты. Это снижает связанность, повышает модульность и упрощает развитие системы [8, 9].

Отказоустойчивость в целевой архитектуре обеспечивается за счет потоковой репликации PostgreSQL, использования резервных узлов и регламентированных сценариев переключения. При этом важно подчеркнуть: сама по себе репликация не является полным решением задачи высокой доступности. Она должна сопровождаться процедурами мониторинга состояния узлов, проверкой задержки репликации, регламентом переключения, тестированием восстановления и контролем целостности данных [12, 16].

Подсистема резервного копирования должна строиться на сочетании базовых резервных копий и архивирования журнала транзакций WAL. Такой подход позволяет не только восстановить базу после отказа, но и вернуться к состоянию на определенный момент времени. Для MES-системы это критично, поскольку исторические технологические данные используются для анализа производственных режимов, расследования

причин отклонений, подготовки отчетности и обучения аналитических моделей [1, 12].

Работа с временными рядами должна быть организована на уровне проектирования схемы данных. Практически значимым решением является секционирование таблиц по временным интервалам, например, по суткам, месяцам или другим периодам, зависящим от интенсивности данных. Секционирование в сочетании с продуманной индексной стратегией позволяет управлять размером активных таблиц, ускорять выборки за определенный период и выполнять архивирование исторических данных без разрушения общей структуры системы [12].

Важным элементом целевой архитектуры является разделение контуров разработки, тестирования и промышленной эксплуатации. Такое разделение позволяет проверять изменения схемы данных, миграционные сценарии, процедуры восстановления и новые запросы до их внедрения в промышленную среду. Для ИТ-службы предприятия это означает переход от эпизодического сопровождения к управляемой инженерной практике, близкой к принципам DevOps и DataOps [8, 10].

Интеграционный слой может включать Apache NiFi и аналогичные инструменты потоковой или пакетной обработки данных. Их целесообразно использовать для маршрутизации, первичной подготовки, контроля происхождения данных и прототипирования интеграционных сценариев [15]. Однако сложная бизнес-логика и критически важные вычисления должны проектироваться так, чтобы их можно было тестировать, версионировать и сопровождать независимо от конкретного инструмента интеграции.

Отдельное значение имеет наблюдаемость. Для промышленной эксплуатации недостаточно знать, что СУБД «работает». Необходимо контролировать задержки запросов, нагрузку на диски, активность транзакций, задержку репликации, объем WAL, частоту блокировок, состояние резервного копирования и доступность сервисов. Интеграция с системами мониторинга и визуализации метрик повышает управляемость ИТ-инфраструктуры и позволяет выявлять проблемы до того, как они отразятся на производственном контуре [12, 16].

VI. СРАВНИТЕЛЬНЫЙ АНАЛИЗ PostgreSQL, MySQL И ClickHouse

Для проверки применимости предложенных критериев рассмотрены три распространенных open-source решения: PostgreSQL, MySQL и ClickHouse

Таблица 2 – Сравнение СУБД по критериям промышленного MES-уровня

Критерий	PostgreSQL	MySQL	ClickHouse
Транзакционная надежность	Высокая: полноценная поддержка ACID	Зависит от выбранного движка и настроек	Ограничена, система ориентирована на аналитические сценарии
Отказоустойчивость	Потоковая репликация, резервные узлы, зрелые сценарии восстановления	Репликация доступна, но гибкость архитектурных сценариев ниже	Репликация и шардинг возможны, но эксплуатационная сложность выше
Масштабируемость	Реплики чтения, секционирование, возможность распределения нагрузок	Хороша для типовых веб-сценариев, но менее универсальна для MES	Очень высока для аналитических чтений
Работа с временными рядами	Эффективна при правильном секционировании и индексировании	Средняя, требует дополнительной оптимизации	Очень высокая для аналитических временных рядов
Аналитическая производительность	Высокая для смешанных сценариев при корректной архитектуре	Средняя	Очень высокая для OLAP-нагрузок
Универсальность для MES	Высокая: поддерживает транзакционные и аналитические сценарии	Средняя	Низкая как основная транзакционная СУБД, высокая как аналитический контур
Эксплуатационная сложность	Средняя, хорошо документирована и поддерживается сообществом	Низкая для типовых задач	Высокая при промышленной эксплуатации

Для демонстрации работы модели была сформирована экспертная матрица оценок по критериям таблицы 2. Значения баллов и уровней уверенности задавались на основе анализа документации рассматриваемых СУБД, опубликованных архитектурных рекомендаций и результатов пилотной апробации.

Таблица 3 – Сравнение СУБД по интегральным показателям

СУБД	S	U	Критические нарушения
PostgreSQL	65.66	18.55	0
MySQL	48.28	28.00	0
ClickHouse	51.01	26.85	2

MySQL является зрелой и широко распространенной СУБД, удобной для большого числа прикладных задач. Ее преимущества связаны с относительной простотой эксплуатации, большим количеством специалистов и развитой экосистемой. Однако для MES-уровня этого недостаточно. Приоритетными становятся строгая транзакционная согласованность, предсказуемость при интенсивных потоках технологических данных, гибкость масштабирования и развитые механизмы

(таблицы 2-3). Эти системы относятся к разным архитектурным классам и поэтому позволяют показать принципиальную разницу между универсальной транзакционной СУБД, классической реляционной системой и аналитическим хранилищем, ориентированным на OLAP-нагрузки [12-14].

промышленного восстановления. По совокупности этих факторов MySQL может применяться в отдельных сервисах, но менее предпочтительна как основная СУБД критической MES-подсистемы [13].

ClickHouse, напротив, демонстрирует высокую эффективность в аналитических сценариях, особенно при обработке больших объемов исторических данных и временных рядов. Эта система может быть крайне полезной как специализированный аналитический контур, например, для витрин данных, отчетности и исследования больших архивов технологических параметров. Однако архитектурная ориентация ClickHouse на OLAP-нагрузки ограничивает ее применение в роли основной транзакционной СУБД MES-уровня [14].

PostgreSQL занимает промежуточную, но наиболее сбалансированную позицию. Она поддерживает строгую транзакционную модель, позволяет организовать репликацию, резервное копирование, восстановление, секционирование и интеграцию с широким набором open-source инструментов. Это делает PostgreSQL рациональным выбором для подсистемы хранения данных MES-уровня, где требуется совместить транзакционные операции, исторический архив, технологическую отчетность и подготовку данных для аналитики [12, 17].

Важно отметить, что результат сравнительного анализа не означает универсального превосходства PostgreSQL над всеми альтернативами. Более

корректный вывод состоит в том, что PostgreSQL является наиболее сбалансированным решением для роли основной СУБД в рассматриваемой архитектуре. ClickHouse при этом может дополнять PostgreSQL как аналитическое хранилище, а MySQL может использоваться в некритичных или вспомогательных сервисах. Такой вывод соответствует современному архитектурному принципу: для каждого класса задач должна выбираться технология, наиболее соответствующая ее эксплуатационной роли [9, 12-14].

VII. СОВРЕМЕННЫЕ ИТ-ТЕХНОЛОГИИ И ПОЛЬЗА ДЛЯ ОТРАСЛИ

Рассматриваемая работа имеет значение не только для отдельного промышленного предприятия, но и для ИТ-отрасли в целом. Она показывает, что импортозамещение корпоративных информационных систем должно рассматриваться как инженерная модернизация архитектуры, а не как формальная замена одного программного продукта другим. Такой подход повышает качество проектирования, формирует компетенции сопровождения open-source решений и снижает технологическую зависимость от закрытых платформ [8, 9].

В технологическом отношении статья опирается на современные направления развития ИТ-инфраструктуры: open-source СУБД, модульную архитектуру, репликацию и отказоустойчивость, секционирование больших таблиц, управление временными рядами, резервное копирование с восстановлением на момент времени, разделение контуров разработки и эксплуатации, интеграционные инструменты обработки потоков данных, средства мониторинга и наблюдаемости. Совокупность этих технологий формирует основу надежной промышленной платформы данных [10-12, 15, 16].

Для ИТ-служб предприятий практическая польза состоит в наличии набора критериев, который можно использовать при архитектурном аудите действующих систем, подготовке технических требований, сравнении альтернативных open-source решений и планировании этапов миграции. Это снижает риск субъективного выбора технологии и позволяет аргументировать решения перед техническим руководством, службой эксплуатации и заказчиками производственных информационных систем.

Для разработчиков и интеграторов MES-систем предложенный подход важен тем, что задает более строгую рамку проектирования. База данных рассматривается не изолированно, а в связи с интеграционным слоем, сервисами обработки данных, аналитическими модулями, резервным копированием, мониторингом и процедурами сопровождения. Это способствует развитию зрелых инженерных практик, повышает надежность внедряемых решений и делает open-source стек более приемлемым для промышленных заказчиков [8-10].

Для отрасли промышленной цифровизации значение работы состоит в том, что она поддерживает переход к

открытым, воспроизводимым и расширяемым платформам данных. Такие платформы могут служить основой для дальнейшего развития промышленной аналитики, предиктивной диагностики, оптимизации производственных режимов и построения цифровых двойников. При этом качество будущих аналитических решений напрямую зависит от надежности и структурированности базового слоя данных [1, 12, 16].

VIII. ПРАКТИЧЕСКАЯ ЗНАЧИМОСТЬ И ОГРАНИЧЕНИЯ ПРИМЕНЕНИЯ

Практическая значимость предложенного подхода заключается в возможности его применения при модернизации действующих MES-систем и проектировании новых решений на основе открытого программного обеспечения. Предприятие может использовать предложенные критерии как основу для технического задания, карты миграции, сравнительной оценки СУБД и проверки готовности целевой архитектуры к промышленной эксплуатации.

При этом внедрение open-source СУБД не следует рассматривать как автоматическое снижение рисков. Напротив, часть ответственности, ранее возлагавшаяся на поставщика проприетарного решения, переходит к команде предприятия или интегратора. Поэтому критически важны квалификация администраторов, регламенты резервного копирования, тестирование восстановления, контроль производительности, документирование архитектуры, управление версиями схемы данных и регулярная проверка отказоустойчивости [8, 12, 16].

Ограничением настоящей работы является качественный характер сравнительной оценки. Для принятия окончательного решения в конкретной организации необходимо дополнить предложенный подход нагрузочным тестированием, анализом структуры реальных данных, оценкой миграционной трудоемкости, проверкой совместимости SQL-кода, расчетом совокупной стоимости владения и анализом требований информационной безопасности. Тем не менее даже на качественном уровне формализованные критерии позволяют существенно повысить обоснованность выбора технологии [9, 12–14].

IX. ЗАКЛЮЧЕНИЕ

В статье рассмотрена задача импортозамещения подсистемы хранения данных MES-уровня промышленного предприятия. Показано, что актуальность данной задачи определяется не только экономическими и организационными факторами, но и технологическими требованиями: ростом объемов временных рядов, потребностью в отказоустойчивости, масштабируемости, наблюдаемости и интеграции с современными инструментами промышленной аналитики.

Выявлено, что исходная проприетарная архитектура, основанная на централизованной СУБД и высокой связанности логики обработки с базой данных, ограничивает гибкость развития системы. Основными

проблемами являются зависимость от вендора, высокая стоимость масштабирования, сложность сопровождения, ограниченная модульность и недостаточная управляемость при росте объемов данных.

Предложен формализованный подход к выбору СУБД для MES-уровня, основанный на оценке отказоустойчивости, резервного копирования и восстановления, транзакционной надежности, масштабируемости, эффективности работы с временными рядами, наблюдаемости, совместимости и предсказуемости поведения под нагрузкой. Такой подход позволяет перейти от субъективного выбора программного продукта к инженерно обоснованному выбору архитектурного компонента.

На основе сравнительного анализа PostgreSQL, MySQL и ClickHouse обосновано, что PostgreSQL обладает наиболее сбалансированным сочетанием характеристик для роли основной СУБД MES-подсистемы. При этом специализированные решения, такие как ClickHouse, целесообразно рассматривать не как прямую замену транзакционной СУБД, а как возможное дополнение для аналитического контура.

Показано, что применение формализованной многокритериальной оценки позволяет снизить субъективность выбора СУБД и обеспечить воспроизводимость архитектурных решений при модернизации промышленных информационных систем.

Практическая ценность работы состоит в том, что предложенные критерии и архитектурные принципы могут быть использованы ИТ-службами, разработчиками и интеграторами при модернизации промышленных информационных систем. Ключевой вывод заключается в следующем: успешное импортозамещение MES-уровня достигается не механической заменой СУБД, а комплексным пересмотром архитектуры хранения, интеграции, мониторинга и сопровождения данных.

БЛАГОДАРНОСТИ

Авторы благодарны сотрудникам кафедры «Теплофизики и информатики в металлургии» за обсуждение вопросов проектирования промышленных информационных систем и архитектурных аспектов импортозамещения программного обеспечения. Полученные замечания и рекомендации были использованы при разработке формализованной модели выбора open-source СУБД для MES-систем.

БИБЛИОГРАФИЯ

- [1] IEC, *IEC 62264-1:2013 Enterprise-Control System Integration – Part 1: Models and Terminology*. Geneva, Switzerland: International Electrotechnical Commission, 2013.
- [2] S. D. Kuznetsov, *Osnovy bazdannykh* [Database Fundamentals]. Moscow, Russia: Binom. Laboratoriya znaniy, 2017. (in Russian).
- [3] D. Talia, P. Trunfio, F. Marozzo, L. Belcastro, R. Cantini, and A. Orsino, *Bol'shie dannye. Sovremennye freymvorki i razrabotka prilozheniy* [Big Data. Modern Frameworks and Application Development]. Moscow, Russia: DMK Press, 2025. (in Russian).
- [4] Oracle Corporation. Oracle Database Documentation [Online]. Available: <https://docs.oracle.com/en/database/oracle/oracle-database/>

- [5] Microsoft. SQL Server Technical Documentation [Online]. Available: <https://learn.microsoft.com/en-us/sql/sql-server/>
- [6] Microsoft. SQL Server Integration Services (SSIS) Documentation [Online]. Available: <https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services>
- [7] Oracle Corporation. Oracle Data Integrator Documentation [Online]. Available: <https://docs.oracle.com/en/middleware/fusion-middleware/data-integrator/14.1.2/>
- [8] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA: Pearson, 2016.
- [9] M. Stonebraker, J. M. Hellerstein, and J. Hamilton, "Architecture of a Database System," *Foundations and Trends in Databases*, vol. 1, no. 2, pp. 141–259, 2007.
- [10] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Indianapolis, IN: Wiley, 2004.
- [11] J. Kreps, *I Heart Logs: Event Data, Stream Processing, and Data Integration*. Sebastopol, CA: O'Reilly Media, 2014.
- [12] J. Densmore, *Konveyery dannykh. Karmannyi spravochnik* [Data Pipelines Pocket Reference]. Astana, Kazakhstan: ALIST, 2024. (in Russian).
- [13] Oracle Corporation. MySQL 8.4 Reference Manual [Online]. Available: <https://dev.mysql.com/doc/refman/8.4/en/>
- [14] ClickHouse Inc. ClickHouse Documentation [Online]. Available: <https://clickhouse.com/docs/>
- [15] Apache Software Foundation. Apache NiFi Documentation [Online]. Available: <https://nifi.apache.org/docs.html>
- [16] J. Pivetto and B. Brazil, *Zapuskaem Prometheus. Monitoring infrastruktury i prilozheniy* [Prometheus Up & Running. Infrastructure and Application Monitoring]. Moscow, Russia: DMK Press, 2023. (in Russian).
- [17] E. V. Rogov, *PostgreSQL 17 iznutri* [PostgreSQL 17 Internals]. Moscow, Russia: DMK Press, 2025. (in Russian).

Formalized selection of open-source database management systems for MES import substitution: an architectural approach and industrial operation criteria

D. E. Uskov, I. A. Gurin, V. V. Lavrov

Abstract — This paper addresses the problem of import substitution of the data storage subsystem at the Manufacturing Execution System (MES) level of an industrial enterprise through migration from a proprietary database management system (DBMS) to an open-source solution. The relevance of such migration is substantiated by the increasing volume of production data and the growing requirements for fault tolerance, observability, and integration with modern industrial analytics tools.

A formalized approach to DBMS selection is proposed based on a set of technical and operational criteria, including transaction reliability, fault tolerance, backup and recovery capabilities, scalability, time-series data processing efficiency, manageability, compatibility with enterprise IT architecture, and predictable behavior under load. Using PostgreSQL as a case study, it is demonstrated that import substitution should not be reduced to a simple replacement of a software product. Instead, it requires a revision of architectural principles, separation of development, testing, and production environments, implementation of replication mechanisms, Write-Ahead Logging (WAL) archiving, data partitioning, and monitoring tools.

The practical significance of the study lies in the applicability of the proposed criteria and evaluation approach to the modernization of existing MES solutions and the design of new industrial information platforms based on open technologies.

Keywords — MES, DBMS, PostgreSQL, open-source, import substitution, industrial information system, time-series data, replication, backup and recovery, fault tolerance, observability, industrial analytics.

REFERENCES

- [1] IEC, *IEC 62264-1:2013 Enterprise-Control System Integration – Part 1: Models and Terminology*. Geneva, Switzerland: International Electrotechnical Commission, 2013.
- [2] S. D. Kuznetsov, *Osnovy bazdannykh* [Database Fundamentals]. Moscow, Russia: Binom. Laboratoriya znaniy, 2017. (in Russian).
- [3] D. Talia, P. Trunfio, F. Marozzo, L. Belcastro, R. Cantini, and A. Orsino, *Bol'shie dannye. Sovremennyye freymvorki i razrabotka prilozheniy* [Big Data. Modern Frameworks and Application Development]. Moscow, Russia: DMK Press, 2025. (in Russian).
- [4] Oracle Corporation. Oracle Database Documentation [Online]. Available: <https://docs.oracle.com/en/database/oracle/oracle-database/>
- [5] Microsoft. SQL Server Technical Documentation [Online]. Available: <https://learn.microsoft.com/en-us/sql/sql-server/>
- [6] Microsoft. SQL Server Integration Services (SSIS) Documentation [Online]. Available: <https://learn.microsoft.com/en-us/sql/integration-services/sql-server-integration-services>
- [7] Oracle Corporation. Oracle Data Integrator Documentation [Online]. Available: <https://docs.oracle.com/en/middleware/fusion-middleware/data-integrator/14.1.2/>
- [8] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA: Pearson, 2016.
- [9] M. Stonebraker, J. M. Hellerstein, and J. Hamilton, “Architecture of a Database System,” *Foundations and Trends in Databases*, vol. 1, no. 2, pp. 141–259, 2007.
- [10] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Indianapolis, IN: Wiley, 2004.
- [11] J. Kreps, *I Heart Logs: Event Data, Stream Processing, and Data Integration*. Sebastopol, CA: O'Reilly Media, 2014.
- [12] J. Densmore, *Konveyery dannykh. Karmannyi spravochnik* [Data Pipelines Pocket Reference]. Astana, Kazakhstan: ALIST, 2024. (in Russian).
- [13] Oracle Corporation. MySQL 8.4 Reference Manual [Online]. Available: <https://dev.mysql.com/doc/refman/8.4/en/>
- [14] ClickHouse Inc. ClickHouse Documentation [Online]. Available: <https://clickhouse.com/docs/>
- [15] Apache Software Foundation. Apache NiFi Documentation [Online]. Available: <https://nifi.apache.org/docs.html>
- [16] J. Pivetto and B. Brazil, *Zapuskaem Prometheus. Monitoring infrastruktury i prilozheniy* [Prometheus Up & Running. Infrastructure and Application Monitoring]. Moscow, Russia: DMK Press, 2023. (in Russian).
- E. V. Rogov, *PostgreSQL 17 iznutri* [PostgreSQL 17 Internals]. Moscow, Russia: DMK Press, 2025. (in Russian).