

УДК 330.4 JEL C02

UDC 330.4 JEL C02

5.2.2. Математические, статистические и инструментальные методы в экономике

5.2.2. Mathematical, statistical and instrumental methods in economics

ДОСТОИНСТВА ПЕРЕХОДА СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА МИКРОСЕРВИСНУЮ АРХИТЕКТУРУ

ADVANTAGES OF TRANSITIONING DECISION SUPPORT SYSTEMS TO MICROSERVICE ARCHITECTURE

Попазов Дмитрий
аспирант

Popazov Dmitry
graduate student

Попова Елена Витальевна
доктор экономических наук, профессор
ФГБОУ ВО «Кубанский государственный
аграрный университет имени И. Т. Трубилина»,
Краснодар, Россия

Popova Elena Vitalievna
Doctor of Economics, Professor
Kuban State Agrarian University named after I. T.
Trubilin, Krasnodar, Russia

Целью данной работы является демонстрация узких мест в выборе архитектуры программного обеспечения для систем поддержки принятия решений. Автором выделены и разобраны достоинства и сложности перехода информационных систем экономического прогнозирования в виде систем поддержки принятия решений на микросервисную архитектуру. Дан краткий обзор возможностей каждой из выбранных программных архитектур, с описанием их характерных особенностей и структур, с указанием достоинств и недостатков каждой из них. Проведено сравнение с противоположным подходом к организации архитектуры программного обеспечения — монолитной архитектурой, с выделением некоторых преимуществ последней. Приведены доводы, по которым монолитная архитектура в области экономического прогнозирования с использованием систем поддержки принятия решений является неэффективным решением, а также выделена сложность перехода с монолитной архитектуры на микросервисную для продуктов, для уже существующих программных продуктов. Рассмотрены экономические методы, которые гораздо эффективнее работают с микросервисной архитектурой, и учитывают её достоинства. Рассмотрены результаты сравнения производительности двух рассматриваемых архитектур при моделировании агентно-ориентированной модели. Обоснован выбор аппаратной архитектуры именно для программных средств экономического прогнозирования — систем поддержки принятия решений. Объяснена связь экономического прогнозирования с потребностями бизнеса и обосновано, почему монолитная архитектура не релевантна для современных бизнес-задач. Рассмотрено почему в настоящее время актуально задаваться вопросом оптимизации работы систем поддержки принятия решений, в связи с тенденциями на интеграцию искусственного

The purpose of this study is to demonstrate the bottlenecks in choosing a software architecture for decision support systems. The author highlights and analyzes the advantages and difficulties of the transition of economic forecasting information systems in the form of decision support systems to a microservice architecture. A brief overview of the capabilities of each of the selected software architectures is given, with a description of their characteristic features and structures, indicating the advantages and disadvantages of each. A comparison is made with the opposite approach to organizing software architecture - a monolithic architecture, highlighting some of the advantages of the latter. Arguments are given why a monolithic architecture in the field of economic forecasting using decision support systems is an ineffective solution, and the complexity of the transition from a monolithic to a microservice architecture for products for existing software products is highlighted. Economic methods are considered that work much more effectively with a microservice architecture and take into account its advantages. The results of comparing the performance of the two architectures under consideration when modeling an agent-oriented model are considered. The choice of hardware architecture specifically for economic forecasting software tools — decision support systems — is substantiated. The connection between economic forecasting and business needs is explained and it is substantiated why monolithic architecture is not relevant for modern business tasks. It is considered why it is currently relevant to ask the question of optimizing the operation of decision support systems, in connection with the trends of integrating artificial intelligence into software products in the field of economic forecasting

интеллекта в программные продукты в области
экономического прогнозирования

Ключевые слова: АРХИТЕКТУРА,
МОНОЛИТНАЯ АРХИТЕКТУРА,
МИКРОСЕРВИСНАЯ АРХИТЕКТУРА,
ЭКОНОМИЧЕСКОЕ ПРОГНОЗИРОВАНИЕ,
АБСТРАКЦИЯ, МОДЕЛЬ, МЕТОД, ДАННЫЕ,
ИНФОРМАЦИОННАЯ СИСТЕМА

Keywords: ARCHITECTURE, MONOLITHIC
ARCHITECTURE, MICROSERVICE
ARCHITECTURE, ECONOMIC FORECASTING,
ABSTRACTION, MODEL, METHOD, DATA,
INFORMATION SYSTEM

<http://dx.doi.org/10.21515/1990-4665-211-016>

Введение

Область экономического прогнозирования тесно связана с информационными технологиями, представляя из себя совокупность абстракций. Абстракции призваны упростить понимание сложносочиненных вещей, и в какой-то степени абстракцию можно считать метафорой, внутрь которой исследователь волен не укладывать мельчайшие подробности предмета абстрагирования, а представить его суть, лишенную деталей, которыми полезно было бы пренебречь.

Если взглянуть на математические модели методы, то мы можем с уверенностью сказать, что это абстракция, только математическая, выполненная в срезе чисел и переменных. Если смотреть глубже, и определить, как работает экономическое прогнозирование в принципе, то мы можем сделать вывод, что математические модели бесполезны, если нет системы, которая могла бы их применять. То есть над математическими моделями необходим ещё один уровень абстракции, который представляет собой информационную систему, отражающую те подходы для обработки «сырых» численных данных, которые в неё заложены.

Система поддержки принятия решений (далее СППР) является этой самой информационной системой. Она анализирует полученные данные, и выстраивает сценарий прогноза, на основании фактических данных, и данных собранных за прошлые отчётные периоды.

<http://ej.kubagro.ru/2025/07/pdf/16.pdf>

СППР, как и прочие информационные системы, строятся и функционируют согласно выбранной архитектуре. В данной статье речь пойдет о проблемах выбора архитектуры для разработки и поддержки СППР, о том почему до сих пор используют «устаревший» вид архитектуры, и о том, насколько важно именно в вопросах экономического прогнозирования опираться на современные подходы выбора архитектуры.

Теоретическая база

Архитектуры информационных систем можно разделить на две крупные категории.

Монолитная архитектура (монолит). Не вдаваясь в исторический контекст, монолит удобно представлять как единую систему, сосредоточенную в границах одного активного процесса. Для монолита характерно условное разбиение на компоненты, каждый из которых «знает» о существовании других, и тесно с ними связан. К тому же когда монолитное приложение небольшое по размеру и имеет всего несколько функций, он обладает своими сильными сторонами, например, простота разработки, тестирования, развертывания и масштабирования [1]. Применение монолита удобно в качестве «стартовой» архитектуры, поскольку подобного рода продукты можно быстро развить, и придать им ту форму с которой можно начать решать бизнес-проблемы. Но с учетом того, что с ростом числа пользователей, и расширением спектра бизнес-задач, которой системе предстоит решать, она будет разрастаться и здесь закрадывается главная проблема монолитной архитектуры — сложность масштабирования.

Можно поддерживать и развивать монолитное программное обеспечение, но в конечном итоге становится очевидным, что необходимо внести изменения в архитектуру всего приложения. Такая тенденция впервые была замечена в компаниях с большим трафиком, большим количеством разработчиков и большой кодовой базой. Такие компании,

как Amazon, Netflix, LinkedIn, SoundCloud и многие другие перешли на микросервисную архитектуру, потому что их существующее монолитное приложение было слишком сложно поддерживать и развивать [2].

Микросервисная архитектура. Микросервисная архитектура — это подход к созданию приложения, который предполагает отказ от единой, монолитной структуры. То есть вместо выполнения всех ограниченных контекстов приложения на сервере посредством внутривещественных взаимодействий используется множество небольших приложений, каждое из которых соответствует какому-то ограниченному контексту [3].

Микросервисы образуются из декомпозиции монолитных приложений. В монолите выделяется несколько абстрагированных бизнес-сущностей, каждая из которых выделяет строго отведенную ей роль. Они слабо связаны между собой, и могут функционировать по отдельности, что значительно упрощает тестирование отдельных компонентов, однако усложняет тестирование системы в целом, поскольку работа системы происходит в разных процессах. Но помимо всего прочего, микросервисные архитектуры значительно проще масштабируются.

Преимущества микросервисной архитектуры в области экономического прогнозирования.

Стоит понимать, что даже не смотря на простоту построения монолитной архитектуры, удобство тестирования «в одном потоке», СППР не может обладать настоящей гибкостью если оставить её «замурованной» в монолит. Под этим эвфемизмом подразумевается одна из проблем масштабирования: монолит, построенный с помощью технологии X, может интегрироваться с технологией Y только в виде приращения к нему другого монолита. То есть, если скажем, возникла бизнес-задача, которая требует, чтобы к системе-монолиту, написанному на Java, были добавлены Python и TensorFlow (открытая программная

библиотека для машинного обучения), то сделать это путем интеграции нового модуля в монолит — не получится.

В то же время, микросервисная архитектура располагает возможностями, добавлять в цепочку своих приложений — новое, построенное с использованием другой технологии.

Микросервисная архитектура, являясь, как говорилось в начале, абстракцией, которая собирает под собой самые необходимые параметры предмета абстрагирования и превращает это в объект — может быть применена не только к готовым программным модулям, но и, например, к классам ООП. В исследовании Д. Н. Гаврилина, И. А. Кустова и А. В. Манциводы, как раз описывается подход, при котором «микросервис» представляет собой класс, объект, который наделен всеми возможностями микросервиса [5].

В наше время, когда такие задачи экономического прогнозирования, как поиск корреляций в огромном потоке «сырых» данных, может быть адаптирован под работу с искусственным интеллектом, может возникнуть потребность «обновить» СППР с учетом новых технологических возможностей, но если система написана с учетом монолитной архитектуры, то сделать это будет гораздо сложнее и, что главное, дороже, чем с учетом микросервисной архитектуры.

Также стоит отметить, что благодаря разделению на самостоятельные компоненты, тяжелонагруженные и ресурсоёмкие задачи могут выполняться параллельно, значительно сокращая время, необходимое на обработку данных.

Среди таких задач, которые напрямую ставятся в процессе экономического моделирования можно выделить следующие:

1. Агентно-ориентированное моделирование:

– необходимо моделирование большого количества экономических агентов, взаимодействующих по сложным правилам;

- высокие требования к процессорной мощности и оперативной памяти;
- обязательна возможность масштабировать части модели независимо.

Для пересчета состояния масштабной системы с нетривиальной логикой поведения и взаимодействия агентов требуются значительные вычислительные ресурсы [4].

2. Сценарное моделирование с множеством параллельных симуляций.

Это методика, используемая в финансовом анализе для оценки различных сценариев развития событий и их влияния на финансовые показатели компании. Этот подход позволяет лучше понять возможные риски и возможности, а также подготовиться к разным вариантам развития ситуации.

Этап внедрения представляет собой ценность в связи с тем, что сценарный метод является достаточно дорогостоящим и ресурсоёмким мероприятием [6].

3. Методы машинного обучения и глубокого обучения

Большие модели, особенно нейросети (например, LSTM, Transformer), требуют значительных процессорных мощностей, параллелизма, а иногда распределённого обучения.

Обучение на больших наборах временных рядов или экономических индикаторов плохо укладываются в монолитную архитектуру, но, как говорилось ранее, в микросервисной архитектуре благодаря инкапсуляции различных типов ответственности в самостоятельные модули, которые могут обособленно обрабатывать данные, в виду наличия в каждом модуле собственной базы данных — это представляется возможным.

Сравнение производительности архитектур.

Схематически, производительность архитектур можно представить рисунком 1.

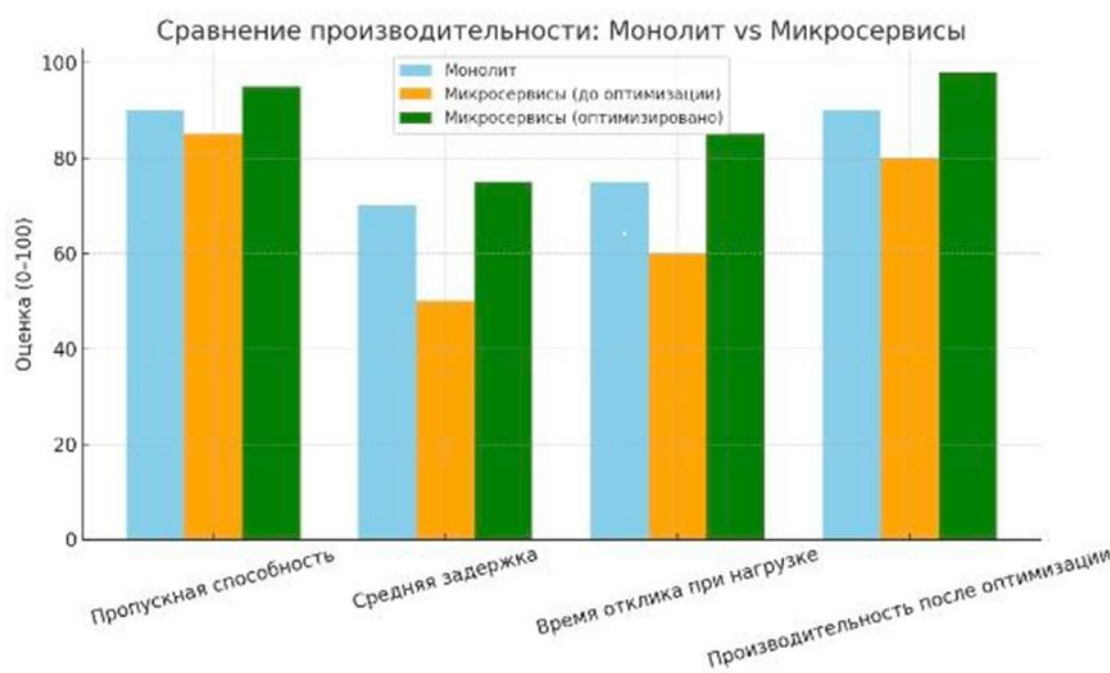


Рисунок 1 – Гистограмма сравнения производительности монолитной и микросервисной архитектуры.

Отметим, что монолит по производительности выигрывает у микросервисов в том случае, когда микросервисная система не оптимизирована. Также стоит заметить, что монолит хорошо себя показывает, только в том случае, когда система имеет малый объем трафика, и сама является достаточно небольшой.

Что же касается математического моделирования, то нет необходимости утверждать, что на любые вопросы есть один единственный ответ, и это — микросервисы. Наоборот, существуют методы математического моделирования, которые могут решать бизнес-задачи и помогать принимать решения без специализированных аппаратных средств в принципе. Например, метод асинхронного гармонического анализа [7], для выполнения которого требуются мощности обычного ПК, и выводы полученные им тоже имеют бизнес-ценность.

Для оценки производительности в срезе экономических моделей, мы рассмотрим упомянутый ранее подход агентно-ориентированного моделирования, требующего значительных ресурсов. Для анализа выбрана агентно-ориентированную модель в экономике, поскольку такие модели симулируют поведение тысяч или миллионов агентов (домохозяйства, фирмы и пр.), активно используют условно-рандомизированные правила взаимодействия. Известно, что при реализации на таких моделях нагрузка на CPU, память и межкомпонентное взаимодействие может быть значительной, особенно при росте числа агентов.

Бенчмарки моделирования системы представлены в следующих проекциях: среднее время выполнения, навигационные накладные расходы, пиковая CPU-нагрузка, память, производительность и ошибки при нагрузке.

Отталкиваясь при сравнении планируется от обычного монолита и от оптимизированного микросервиса, причем, за сессию моделируется поведение ста тысяч агентов.

Таблица 1 – Бенчмарки сравнения производительности микросервисной и монолитной архитектуры, при подходе агентно-ориентированного моделирования.

Параметр	Монолит	Микросервисы	Источник/ комментарий
Среднее время выполнения	~120 сек	~140 сек + ~20% сетевые задержки	[8]
Накладные расходы (latency)	~1 мс на вызов функции	~5–20 мс RPC	[9]
Пиковая CPU-нагрузка	~85–90% на одном узле	~60–70% на сервис, лучше в кластере	[10]
Использование памяти	~64 ГБ суммарно	~80 ГБ (RAM-overhead ~25%)	[11]
Пропускная способность (throughput)	~3 000 агентов/сек	~2 500 агентов/сек на сервис	[12]
Ошибки при нагрузке	Сбой компонента = сбой всей системы	Локализация отказа (10–15%)	[13]

Выводы

На основании сравнения бенчмарков при моделировании одной и той же ситуации, можно сделать вывод, что оптимизированные микросервисы, в случае, когда необходимо решение ресурсоемкой задачи, справляются значительно быстрее, однако, требуется больше вычислительных ресурсов. По показателям *Среднее время выполнения*, *Накладные расходы* и *Использование памяти*, монолит явно опережает микросервисы, даже с учетом вычета из последних погрешности на то, что разные модули одной микросервисной системы могут находиться банально физически далеко друг от друга, и необходимы дополнительные накладные расходы на передачу данных.

По нашему мнению, ключевыми факторами выступают три показателя: *Пропускная способность*, *Ошибки при нагрузке* и *Пиковая CPU-нагрузка*. Объясняется это потребностями и сущностью бизнеса: как правило, надежность системы ставится выше накладных расходов, поскольку в таком случае есть гарантии того, что система выполнит свою бизнес-задачу исправно.

Исходя из таблицы напрашивается вывод о том, что малую разницу в ресурсопотреблении между монолитом и микросервисом возможно нивелировать горизонтальным расширением системы, то есть докупить новые программные компоненты и сервера. В то же время, горизонтальное расширение монолиту не доступно, поскольку нагрузка, как показано в таблице, приходится на один узел, который един, в то время как в микросервисах нагрузка распределяется на каждое отдельное приложение, что увеличивает отказоустойчивость системы в целом.

Также в пользу отказоустойчивости микросервисов, стоит заметить, что при выходе из строя одного компонента монолитной архитектуры — перестает работать вся система полностью, в то время как в

микросервисной архитектуре — перестает работать лишь одно отдельное приложение.

Бизнес-цель процесса заключается в получении экономического эффекта от соответствующего вида деятельности [14], следовательно, бизнесу больше соответствует совокупность отказоустойчивости, масштабирования и скорости выполнения, а горизонтальное расширение системы можно отнести к расходам, которые имеют возможность окупиться, в отличие от проблем, связанных с ошибками в системе, или её полная реставрация, из-за невозможности добавления новых программных компонентов.

Исходя из всего вышеперечисленного, можно сделать вывод, что в условиях современного бизнеса, использующего экономическое прогнозирование, как инструмент принятия решений, микросервисная архитектура является более предпочтительным выбором, чем монолит.

Заключение

В представленной работе выделены и разобраны достоинства перехода информационных систем экономического прогнозирования в виде систем поддержки принятия решений на микросервисную архитектуру. Проведено сравнение с противоположным подходом к организации архитектуры программного обеспечения — монолитной архитектурой. Приведены доводы, по которым монолитная архитектура в области экономического прогнозирования с использованием систем поддержки принятия решений является неэффективным решением. Научная новизна данной статьи заключается в обосновании выбора аппаратной архитектуры именно для программных средств экономического прогнозирования — систем поддержки принятия решений. Удалось связать задачи экономического прогнозирования с потребностями бизнеса и обосновать, почему монолитная архитектура не релевантна для современных бизнес-задач.

Экономическое моделирование занимает важную часть бизнес пространства. Из данных прогноза складывается вектор развития той или иной компании, и инструменты, которые помогают осуществлять прогноз, улучшаются ежедневно, особенно в наши дни, когда искусственный интеллект открывает новые возможности для ускорения работы систем поддержки принятия решений. Архитектура СППР — это всего лишь одна из возможных дверей, сквозь которые можно впустить новые возможности.

СПИСОК ИСТОЧНИКОВ

1. Кабарухин А.П. Выгоды перехода от монолитной к микросервисной архитектуре приложения // Проблемы современной науки и образования. 2022. URL: [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/vygody-perehoda-ot-monolitnoy-k-mikroservisnoy-arhitekture-prilozheniya> (дата обращения: 06.06.2025).
2. Маличенко С.В. Проблемы перехода от монолитной к микросервисной архитектуре // Евразийский научный журнал – 2023 / [Электронный ресурс]. – Режим доступа: URL: <https://journalpro.ru/pdf-article/?id=14519> (дата обращения: 01.05.2025).
3. Осипов Д.Б. Проектирование программного обеспечения с помощью микросервисной архитектуры // Вестник науки и образования. 2018. №5 (41). URL: [Электронный ресурс]. Режим доступа: <https://scientificjournal.ru/images/PDF/2018/VNO-41/proektirovanie.pdf> (дата обращения: 03.05.2025).
4. Макаров В.Л., Бахтизин А.Р., Сушко Е.Д. Технология поддержки агент-ориентированного моделирования для суперкомпьютеров. 2015. URL: [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/tehnologiya-podderzhki-agent-orientirovannogo-modelirovaniya-dlya-superkompyuterov> (дата обращения: 06.06.2025).
5. Гаврилин Д. Н., Кустова И. А, Манцивода А. В. Объектные модели как микросервисы: язык запросов // Известия Иркутского государственного университета. Серия Математика. 2022. Т. 42. С. 121–137.
6. Писарева О.М., Перекальский В.А. Выгоды перехода от монолитной к микросервисной архитектуре приложения // Проблемы современной науки и образования. 2016. URL: [Электронный ресурс]. Режим доступа: <https://economy.spbstu.ru/article/2016.60.21/> (дата обращения: 06.06.2025).
7. Дроговоз П. Ф., Драгун Е. А, Драгун М. А. Математический метод прогнозирования динамики инновационного развития на основе асинхронного гармонического анализа // Modern Economy Success 2020 .- № 6 .- С. 224 – 229
8. Blinowski G., Krolkowski W., Kaleta M., Jureczko M. Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation // IEEE Access. — 2022. — Vol. 10. — P. 26035–26049. — DOI: 10.1109/ACCESS.2022.3157173
9. Sikora A., Tomaszewski M., Piotrowski J. Performance Comparison of Monolith and Microservices Based Applications // Proceedings of the 25th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI). — 2021. — P. 75–80. — URL: [https://www.iiisci.org/Journal/CV\\$/sci/pdfs/ZA215YP21.pdf](https://www.iiisci.org/Journal/CV$/sci/pdfs/ZA215YP21.pdf) (дата обращения: 04.06.2025).

10. Børndal Ø., Bækgaard L., Simonsen J.G. Benchmarks and Performance Metrics for Assessing the Migration to Microservice-Based Architectures // *Journal of Object Technology*. — 2021. — Vol. 20, No. 2. — P. 1–14. — DOI: 10.5381/jot.2021.20.2.a3
11. Amaral J., Polo J., Carrera D. Performance Evaluation of Microservices Architectures using Containers // *arXiv preprint*. — 2015. — arXiv:1506.01345. — URL: <https://arxiv.org/abs/1506.01345> (дата обращения: 04.06.2025).
12. Barzotto R., Farias M. Evaluation of the Impacts of Decomposing a Monolithic Application into Microservices: A Case Study // *arXiv preprint*. — 2022. — arXiv:2212.03052. — URL: <https://arxiv.org/abs/2212.03052> (дата обращения: 04.06.2025).
13. Faustino C., Nunes D., Rodrigues L., Antunes L. Stepwise Migration of a Monolith to a Microservices Architecture: Performance and Migration Effort Evaluation // *arXiv preprint*. — 2022. — arXiv:2207.06170. — URL: <https://arxiv.org/abs/2207.06170> (дата обращения: 04.06.2025).
14. Фёдоров И.Г. Методика выявления целей, задач и требований бизнес-процесса // *Прикладная информатика*. 2014. URL: [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/metodika-vyyavleniya-tseley-zadach-i-trebovaniy-biznes-protsessa/viewer> (дата обращения: 09.06.2025).

References

1. Kabaruxin A.P. Vy`gody` perexoda ot monolitnoj k mikroservisnoj arxitekture prilozheniya // *Problemy` sovremennoj nauki i obrazovaniya*. 2022. URL: [E`lektronny`j resurs]. Rezhim dostupa: <https://cyberleninka.ru/article/n/vygody-perehoda-ot-monolitnoy-k-mikroservisnoj-arhitekture-prilozheniya> (data obrashheniya: 06.06.2025).
2. Malichenko S.V. Problemy` perexoda ot monolitnoj k mikroservisnoj arxitekture // *Evrazijskij nauchny`j zhurnal* – 2023 / [E`lektronny`j resurs]. – Rezhim dostupa: URL: <https://journalpro.ru/pdf-article/?id=14519> (data obrashheniya: 01.05.2025).
3. Osipov D.B. Proektirovanie programmogo obespecheniya s pomoshh`yu mikroservisnoj arxitektury` // *Vestnik nauki i obrazovaniya*. 2018. №5 (41). URL: [E`lektronny`j resurs]. Rezhim dostupa: <https://scientificjournal.ru/images/PDF/2018/VNO-41/proektirovanie.pdf> (data obrashheniya: 03.05.2025).
4. Makarov V.L., Baxtizin A.R., Sushko E.D. Texnologiya podderzhki agent-orientirovannogo modelirovaniya dlya superkomp`yuterov. 2015. URL: [E`lektronny`j resurs]. Rezhim dostupa: <https://cyberleninka.ru/article/n/tehnologiya-podderzhki-agent-orientirovannogo-modelirovaniya-dlya-superkompyuterov> (data obrashheniya: 06.06.2025).
5. Gavrilin D. N., Kustova I. A., Mancivoda A. V. Ob`ektny`e modeli kak mikroservisy`: yazy`k zaprosov // *Izvestiya Irkutskogo gosudarstvennogo universiteta. Seriya Matematika*. 2022. T. 42. C. 121–137.
6. Pisareva O.M., Perekal`skij V.A. Vy`gody` perexoda ot monolitnoj k mikroservisnoj arxitekture prilozheniya // *Problemy` sovremennoj nauki i obrazovaniya*. 2016. URL: [E`lektronny`j resurs]. Rezhim dostupa: <https://economy.spbstu.ru/article/2016.60.21/> (data obrashheniya: 06.06.2025).
7. Drogovoz P. F., Dragun E. A., Dragun M. A. Matematicheskij metod prognozirovaniya dinamiki innovacionnogo razvitiya na osnove asinxronnogo garmonicheskogo analiza // *Modern Economy Success 2020* .- № 6 .- S. 224 – 229
8. Blinowski G., Krolkowski W., Kaleta M., Jureczko M. Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation // *IEEE Access*. — 2022. — Vol. 10. — P. 26035–26049. — DOI: 10.1109/ACCESS.2022.3157173
9. Sikora A., Tomaszewski M., Piotrowski J. Performance Comparison of Monolith and Microservices Based Applications // *Proceedings of the 25th World Multi-Conference on*

Systemics, Cybernetics and Informatics (WMSCI). — 2021. — P. 75–80. — URL: [https://www.iiisci.org/Journal/CV\\$/sci/pdfs/ZA215YP21.pdf](https://www.iiisci.org/Journal/CV$/sci/pdfs/ZA215YP21.pdf) (data obrashheniya: 04.06.2025).

10.Bjørndal Ø., Bækgaard L., Simonsen J.G. Benchmarks and Performance Metrics for Assessing the Migration to Microservice-Based Architectures // Journal of Object Technology. — 2021. — Vol. 20, No. 2. — P. 1–14. — DOI: 10.5381/jot.2021.20.2.a3

11.Amaral J., Polo J., Carrera D. Performance Evaluation of Microservices Architectures using Containers // arXiv preprint. — 2015. — arXiv:1506.01345. — URL: <https://arxiv.org/abs/1506.01345> (data obrashheniya: 04.06.2025).

12.Barzotto R., Farias M. Evaluation of the Impacts of Decomposing a Monolithic Application into Microservices: A Case Study // arXiv preprint. — 2022. — arXiv:2212.03052. — URL: <https://arxiv.org/abs/2212.03052> (data obrashheniya: 04.06.2025).

13.Faustino C., Nunes D., Rodrigues L., Antunes L. Stepwise Migration of a Monolith to a Microservices Architecture: Performance and Migration Effort Evaluation // arXiv preprint. — 2022. — arXiv:2207.06170. — URL: <https://arxiv.org/abs/2207.06170> (data obrashheniya: 04.06.2025).

14. Fyodorov I.G. Metodika vy`yavleniya celej, zadach i trebovanij biznes-processa // Prikladnaya informatika. 2014. URL: [E`lektronny`j resurs]. Rezhim dostupa: <https://cyberleninka.ru/article/n/metodika-vyyavleniya-tseley-zadach-i-trebovaniy-biznes-protsessa/viewer> (data obrashheniya: 09.06.2025).