**Проблемы и риски при росте нагрузки:**

1. **Задержки ответов**:
   * Увеличение количества страховых компаний с 5 до 10 усложнит процесс агрегации данных, увеличив время ожидания ответов от всех страховых компаний.
   * Задержки могут привести к истечению времени ожидания в сервисах core-app и ins-comp-settlement, вызывая сбои в работе системы.
2. **Ошибки при взаимодействии с API страховых компаний**:
   * Поскольку процесс агрегации синхронный, любая ошибка на стороне одной из страховых компаний может остановить весь процесс, что приведет к невозможности получить данные о продуктах.
3. **Непостоянство данных**:
   * Обновление данных в core-app и ins-comp-settlement происходит с разной частотой (15 минут для core-app и раз в сутки для ins-comp-settlement). Это приводит к несогласованности данных между сервисами и возможным устаревшим данным в случае сбоя при обновлении.
4. **Увеличение нагрузки на сервисы**:
   * Постоянные опросы API страховых компаний увеличивают нагрузку на сервисы ins-product-aggregator и на сети, что может замедлить работу системы при увеличении количества компаний.
5. **Синхронные запросы**:
   * Синхронные запросы на сбор данных требуют постоянного соединения и могут быть неэффективными при увеличении количества страховых компаний.

**Предложенные решения:**

1. **Переход на Event-Driven архитектуру**:
   * **Event-Streaming**: Внедрение Kafka брокера сообщений для передачи данных о продуктах между сервисами ins-product-aggregator, core-app и ins-comp-settlement.
     + Ins-product-aggregator будет публиковать обновления о продуктах как события, а сервисы core-app и ins-comp-settlement будут подписываться на эти события и обновлять локальные реплики данных.
   * **Преимущества**: Ускорение взаимодействия за счет асинхронного обновления данных. Обновления данных будут поступать в реальном времени, что избавит от необходимости синхронных запросов.
2. **Использование паттерна Transactional Outbox**:
   * Паттерн Transactional Outbox позволит ins-product-aggregator записывать обновления в базу данных и отправлять события в Kafka в рамках одной транзакции, что гарантирует согласованность данных.
3. **Разделение агрегации данных**:
   * Вместо того чтобы запрашивать данные от всех страховых компаний синхронно, ins-product-aggregator может публиковать события для каждой страховой компании по мере получения данных, позволяя другим сервисам начать обработку частичных данных.
4. **Изменение периодичности обновления данных**:
   * Сервис ins-comp-settlement может получать данные по мере их обновления через Event-Streaming, что устранит необходимость раз в сутки делать REST-запрос в core-app.