系统间服务调用与监控

Feign

feign，声明式服务调用接口

通用配置

服务调用方

```
feign:
  httpclient:
    enabled: true    #
  hystrix:
    # feignhystrixfeignhystrix
    enabled: true
#Thread, THREADSEMAPHOREtoken userid
hystrix.command.default.execution.isolation.strategy: THREAD

ribbon:
  ReadTimeout: 50000
  ConnectTimeout: 50000

# loggingdebugfeign
logging:
  level:
    com.acca.opra.mas: DEBUG
    com.acca.opra.sys: DEBUG
```

默认重试1次

代码中配置扫描路径

```java
@ComponentScan(basePackages = {"com.acca.opra.mas", "com.acca.opra.sys"})
@EnableFeignClients({"com.acca.opra.mas", "com.acca.opra.sys"})
@SpringBootApplication
public class AppServer extends ApplicationServer {

    /**
     * .
     *
     * @param args args
     */
    public static void main(String[] args) {
        SpringApplication.run(AppServer.class, args);
    }

}
```

服务熔断

服务降级的情况为，当服务提供方不可用时，api方法返回一个默认值。

需要业务上评估一下，这种情况该如何确定默认值，是否会带来数据的不一致。

```java
/**
 * MasCityApi.
 *
 * @version OPRA v1.0
 * @author Yu Tao, 201999
 */
@Api(value = "MasCity", tags = "MasCity")
@FeignClient(value = "mas-global", url = "${feign.url.mas-global:}",
fallback = MasCityFallback.class)
public interface MasCityApi {

    /**
     *
     *
     * @param cityCode CityCode
     * @return boolean
     */
    @ApiOperation("")
    @ApiImplicitParam(name = "strCode", value = "masCity CityCode",
required = true, dataType = "String")
    @GetMapping("/masCity/existCityCode/{cityCode}")
    boolean existCityCode(@PathVariable String cityCode);

    /**
```

```java
     * id.
     *
     * @param id id
     * @return MasCityVO
     */
    @ApiOperation("id")
    @ApiImplicitParam(name = "id", value = "MasCity id", required = true,
dataType = "String")
    @GetMapping("/masCity/id/{id}")
    MasCityVO findOne(@PathVariable String id);

    /**
     *
     *
     * @param queryVO AutoCompleteQueryVO
     * @return List
     */
    @ApiOperation("cityCode/")
    @ApiImplicitParam(name = "SimpleQueryVO", value = "", required =
false, dataType = "String")
    @GetMapping("/masCitys/simple")
    List<SimpleEntity> findSimpleCity(@RequestParam Map<String, Object>
queryVO);

    /**
     * .
     *
     * @param vo DevQueryVO
     * @return Page<MasCityVO>
     */
    @ApiOperation("")
    @ApiImplicitParam(name = "DevQueryVO", value = "dev", required =
false, dataType = "String")
    @GetMapping("/masCitys")
    Page<MasCityVO> findAll(DevQueryVO vo);

    /**
     * .
     *
     * @param vo MasCityDTO
     * @param id id
     * @return vo
     */
    @ApiOperation("")
    @ApiImplicitParam(name = "id", value = "MasCity id", required = true,
dataType = "String")
    @PutMapping("/masCity/id/{id}")
    MasCityVO put(@PathVariable String id, @RequestBody @Valid MasCityDTO
vo);

    /**
     * .
     *
```

```java
 * @param vo MasCityDTO
 * @return MasCityVO
 */
@ApiOperation("")
@PostMapping("/masCity")
MasCityVO add(@RequestBody @Valid MasCityDTO vo);

/**
 * .
 *
 * @param id id
 * @return String
 */
@ApiOperation("id")
@ApiImplicitParam(name = "id", value = "MasCity id", required = true,
dataType = "String")
@DeleteMapping("/masCity/id/{id}")
boolean delete(@PathVariable String id);

/**
 *
 *
 * @param cityCode cityCode
 * @return Result<String>
 */
@ApiOperation("cityCode")
@ApiImplicitParam(name = "cityCode", value = "cityCode", required =
true, dataType = "String")
@GetMapping("/masCity/getCityEnName/{cityCode}")
Result<String> getCityEnName(@PathVariable String cityCode);

/**
 *
 *
 * @param cityCode cityCode
 * @return Result<String>
 */
@ApiOperation("cityCode")
@ApiImplicitParam(name = "cityCode", value = "cityCode", required =
true, dataType = "String")
@GetMapping("/masCity/getCountryCode/{cityCode}")
Result<String> getCountryCode(@PathVariable String cityCode);


/**
 * .
 *
 *
 * @version OPRA v1.0
 * @author Yu Tao, 2020421
 */
@Component
class MasCityFallback implements MasCityApi {
```

```java
        @Override
        public boolean existCityCode(String cityCode) {
            // false.
            return false;
        }

        @Override
        public MasCityVO findOne(String id) {
            return null;
        }

        @Override
        public List<SimpleEntity> findSimpleCity(Map<String, Object>
queryVO) {
            return Collections.emptyList();
        }

        @Override
        public Page<MasCityVO> findAll(DevQueryVO vo) {
            return null;
        }

        @Override
        public MasCityVO put(String id, @Valid MasCityDTO vo) {
            return null;
        }

        @Override
        public MasCityVO add(@Valid MasCityDTO vo) {
            return null;
        }

        @Override
        public boolean delete(String id) {
            return false;
        }

        @Override
        public Result<String> getCityEnName(String cityCode) {
            return Result.of("");
        }

        @Override
        public Result<String> getCountryCode(String cityCode) {
            return Result.of("");
        }

    }
}
```

服务降级

当系统资源紧张时，我们可以将一些边缘系统停掉，回收部分资源。那这些服务不存在之后，调用这些服务就会超时或者报错，这时，对于这些停掉的服务就可以采用熔断的策略，使服务迅速返回。

zipkin服务调用链路监控

采用zipkin服务器监控服务调用链路。每个微服务应用通过sleuth发送调用信息到kafka队列。zipkin服务器将kafka队列中的数据持久化到elasticsearch中，用于分析查询。

客户端配置

```
#
spring.sleuth.sampler.probability: 1.0
spring.zipkin.sender.type: kafka
spring.zipkin.kafka.topic: zipkin
spring.kafka.bootstrap-servers: 10.1.21.23:9092
```

查看服务器UI

http://ip:8080/zipkin

如图：