如何批量调用多个合约读取请求

在以太坊等区块链网络中，批量调用多个合约的读取请求是优化性能、降低Gas成本的关键技术。以下是四种主流方案及其实战示例：

⚙ 一、Multicall 合约方案（最优解）

原理：通过部署专用的聚合合约（如MakerDAO的Multicall），将多个读取请求打包为单次链上调用，返回聚合结果。
优势：

- **节省Gas**：单次交易完成多请求，避免重复基础开销（如21,000 Gas基础费） 1 17
- **原子性**：所有调用在同一区块状态执行，数据一致性高 17 44
- **跨链兼容**：支持EVM兼容链（以太坊/BSC/Polygon等） 1 74

实现步骤（Python示例）：

Python

```python
from web3 import Web3

# 初始化连接
w3 = Web3(Web3.HTTPProvider('https://mainnet.infura.io/v3/YOUR_PROJECT_ID'))

# Multicall合约配置（以太坊主网地址）
multicall_abi = [...]  # 包含aggregate函数的ABI
multicall_address = '0xeefBa1e63905eF1D7ACbA5a8513c70307C1cE441'
multicall_contract = w3.eth.contract(address=multicall_address, abi=multicall_abi)

# 构建调用列表（如查询多个代币余额）
calls = [
  {'target': '0xTokenA', 'callData': w3.eth.codec.encode_abi(['address'], ['0xUserAddress'])},
  {'target': '0xTokenB', 'callData': w3.eth.codec.encode_abi(['address'], ['0xUserAddress'])}
]

# 执行批量调用
result = multicall_contract.functions.aggregate(calls).call()
return_data = result[1]  # 获取返回数据列表

# 解码结果
balance_a = w3.eth.codec.decode_abi(['uint256'], return_data[0])[0]
balance_b = w3.eth.codec.decode_abi(['uint256'], return_data[1])[0]
```

适用场景：批量查询余额、合约状态变量（如`totalSupply`） [74]

二、RPC 批量请求（轻量级替代）

原理：通过JSON-RPC的`eth_call`批量发送请求，无需部署合约，但依赖节点支持。
优势：

- **零成本**：仅读取状态，不消耗Gas [55] [66]
- **灵活性**：适用于任意公开可读函数 [55]

实现示例（Web3.js）：

```JavaScript
const batch = new web3.BatchRequest();
const results = [];

// 添加多个请求到批处理
batch.add(web3.eth.getBalance.request('0xAddress1', 'latest', callback));
batch.add(contract.methods.balanceOf('0xAddress2').call.request({}, callback));

// 执行并获取结果
batch.execute();
```

⚡ 三、异步并行调用（高并发场景）

原理：利用异步库（如Python的`asyncio`或Node.js的`Promise.all`）并行发起多个独立请求。
适用场景：

- 节点无批量调用限制时
- 需要快速响应的前端应用 [66]

Python异步示例：

```Python
```

```python
import asyncio
from web3 import AsyncWeb3

async def fetch_balance(token_address, user_address):
    contract = w3.eth.contract(address=token_address, abi=erc20_abi)
    return await contract.functions.balanceOf(user_address).call()

# 并行查询多个代币余额
tasks = [
    fetch_balance('0xTokenA', '0xUser'),
    fetch_balance('0xTokenB', '0xUser')
]
results = await asyncio.gather(*tasks)
```

四、自定义优化技巧

1. **数据打包** 合约设计时使用`struct`打包相关变量，减少调用次数[35]：

```solidity
struct UserData { uint balance; uint reward; }
mapping(address => UserData) public userInfo; // 单次调用获取多数据
```

2. **状态变量缓存** 对频繁读取的数据使用内存缓存，避免重复链上查询[35]。

3. **事件日志过滤** 通过历史事件批量获取数据（如`Transfer`事件追溯余额）[55]。

方案对比与选型建议

方案	成本	速度	适用场景	工具链
Multicall合约	低Gas	快	复杂批量查询（>5个调用）	Web3.py/ethers.js
RPC批量请求	零Gas	中	节点支持批量调用的简单查询	Web3.js
异步并行调用	零Gas	极快	低延迟前端应用，少量请求	asyncio/Promise.all
合约层优化	最低	快	新合约设计阶段	Solidity

> **实战优先推荐**：
>
> - 已有合约查询 → **Multicall方案**（跨链通用、成本低）[1][74]
> - 新合约设计 → **数据打包+事件日志**（长期优化）[35][55]
> - 高并发前端 → **异步并行+RPC批量**（即时响应）[66]

通过结合业务需求选择合适方案，可提升DApp性能10倍以上，尤其在DeFi、资产管理等高频查询场景中效果显著[1][44]。

https://blog.csdn.net/gitblog_00010/article/details/137988821

https://learnblockchain.cn/article/9294

https://blog.csdn.net/o2233445566/article/details/149465032

https://learnblockchain.cn/article/11817

https://zhuanlan.zhihu.com/p/710651022

https://www.secrss.com/articles/61571

https://docs.feishu.cn/v/wiki/Cg09wptWViMOyikIoXKc2sein4Q/a7

https://juejin.cn/post/7229601048381308989

https://blog.csdn.net/gitblog_00010/article/details/137988821

https://learnblockchain.cn/article/9294

https://oacia.dev/%E6%89%B9%E9%87%8F%E7%BC%96%E8%AF%91%E6%99%BA%E8%83%BD%E5

https://github.com/jambestwick/web3jdemo

https://www.theblockbeats.info/news/27412

https://learnblockchain.cn/article/17753

https://blog.csdn.net/yujunlong3919/article/details/80011990

https://www.jos.org.cn/jos/article/html/6664

https://learnblockchain.cn/article/12009

https://blog.csdn.net/qq_36838406/article/details/121306553

https://blog.csdn.net/gitblog_00010/article/details/137988821

https://cloud.tencent.com/developer/article/1175996?policyId=1003

https://docs.pingcode.com/baike/2953117

https://learnblockchain.cn/docs/web3.js/web3-eth.html

https://github.com/jambestwick/web3jdemo

https://developer.aliyun.com/article/1266690

https://cloud.baidu.com/article/2926424

https://learnblockchain.cn/article/9294

https://blog.csdn.net/gitblog_00010/article/details/137988821

https://github.com/jambestwick/web3jdemo

https://blog.csdn.net/mpegfour/article/details/125930205

https://zhuanlan.zhihu.com/p/710651022

https://juejin.cn/post/6917427859154239501

https://www.cnblogs.com/peteremperor/category/1145390.html?page=1

https://learnblockchain.cn/article/16688

https://juejin.cn/post/6844903778089451527

https://learnblockchain.cn/article/11235

https://learnblockchain.cn/article/16801

https://zhuanlan.zhihu.com/p/1889109677708670433

https://blog.csdn.net/gitblog_00010/article/details/137988821

https://developer.aliyun.com/article/1487059

https://www.cnblogs.com/zhanchenjin/p/18631122

https://docs.kaia.io/zh-CN/build/transactions/cookbooks/how-to-optimize-gas-fees/

https://blog.csdn.net/zhuqiyua/article/details/141786409

https://www.rpubs.com/liam/optmizeGas

http://dcbcl.haut.edu.cn/ups/files/20210415/1618461486161863.pdf

https://kb.bsnbase.com/webdoc/view/Pub4028813e711a7c39017134b23a4c1b9b.html

https://learnblockchain.cn/article/10464

https://www.techflowpost.com/article/detail_21176.html

https://www.infoq.cn/article/divide-and-conquer-in-blockchain

https://blog.csdn.net/u013288190/article/details/123721057

https://www.jos.org.cn/josen/article/html/6528

https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/manual/transaction_parallel.html

https://www.gate.com/zh/learn/articles/understanding-smart-contracts-read-write-and-audit/878

https://blog.csdn.net/qq_36838406/article/details/121306553

https://learnblockchain.cn/docs/web3.js/web3-eth.html

https://blog.csdn.net/david2000999/article/details/120176244

https://www.cnblogs.com/wanghui-garcia/p/9507168.html

https://cloud.tencent.com/developer/article/2218003

https://zhuanlan.zhihu.com/p/366293993

https://learnblockchain.cn/docs/web3js-0.2x/

https://www.aizws.net/course/ethsrc/ethsrc-83178752

https://juejin.cn/post/7172952243203735559

https://blog.csdn.net/gitblog_00010/article/details/137988821

https://learnblockchain.cn/article/9294

https://developer.aliyun.com/article/1487059

https://www.infoq.cn/article/xsagm03hi7zm34hhtc0k

https://doc.opendatachain.cn/Guide/4.3.html

https://blog.csdn.net/daisy_ciaotool/article/details/147484936

https://ftp.iij.ad.jp/pub/osdn.jp/bytom/71324/Bystack-White-Paper-ZH.pdf

http://cjc.ict.ac.cn/online/onlinepaper/wyh-2025616152918.pdf

https://learnblockchain.cn/article/11815

https://www.informat.cn/qa/300195

https://learnblockchain.cn/article/9294

https://blog.csdn.net/gitblog_00010/article/details/137988821

https://www.oryoy.com/news/shi-yong-python-shi-xian-duo-he-yue-multicall-de-gao-xiao-diao-yong-ji-qiao-yu-shi-zhan-an-li-jie-xi.html

https://clark-cui.top/posts/%E5%90%88%E7%BA%A6%E7%9A%84multicall.html

https://learnblockchain.cn/article/11817

https://www.cnblogs.com/Junewu/articles/15952728.html

https://blog.csdn.net/o2233445566/article/details/149465032

https://ethersjs.cn/docs/advanced/multicallquery/

https://blog.cpbox.io/2025/06/06/%E5%88%A9%E7%94%A8%E6%89%B9%E9%87%8F%E8%B0%83%E

https://www.cobo.com/developers/v2_cn/guides/transactions/batch-transfer