Servies are deployed in a server hierarchy. A response page is sent to a client server for each request. Most requests are internalized.
If msg, send

Try

op = 2 + f2 (cmd)

> (proj)

switch op

case

loop msg

deal msg

Check

Next

msg

Send

Commit

Prevent commit

op = f \cdot f (cmd)
we're known for

Get support services

Resolution & Live Maintenance

Services have service part & names
\[ F \text{ is a vector}
\]

Choose answer from any source

\[ \text{sent to GT}
\]

\[ \text{consider} \]

\[ \text{and}\]

\[ \text{sort (near to GT)}
\]
Client

Send code here

Copy this code here

Server

Save code

Separate process

Store
Serve global state

⇒ stateful servers

⇒ stateless servers considered better

⇒ global state + stateless

⇒ possible? maybe
buy item → server (stateless)

Usage of cookies to store state
Let $k$ be a global object.

If $\text{ok}(\text{List}(x)) \Rightarrow \text{OK}$

 endl of

Check

Step 1

$\text{send}(\text{request})$

Step 2

Compute x

Note: The diagram is hand-drawn and includes symbols and arrows indicating relationships and conditions.
on next request

sent by someone to someone

not understood by others

sent by someone to someone

cooler -> server data
3. Send us stuff new code

- Check with sales
- Purchase new board space
- Do research

- Update board stuff
- Get pcbs + code

Seren
CPU/IO
for overhead
is not
for workload

sum

send

read

server

multiple services

performance

multiple entries in services
Start of procedure

Proc 1

Proc 2

Server 1

Server 2

Part

use multi-threading
2. Use dynamic threads (1 server thread)
   - Start thread (first server)
   - Parse input
   - New command
   - Loop
   - Send
   - Write
   - Do x
   - f(x)
Also神州数码 - house locks

Is for money free?

Electronic - thread creation/derogation

There's

a heavy focus - large &

dynamic threaded
First # \rightarrow

\text{Some results}

\uparrow

\text{Green}

\text{Words}

\downarrow

\text{Try again}

\rightarrow

\text{Pass}

\rightarrow

\text{Success!}

\rightarrow

\text{Titular}

\rightarrow

\text{Success 2}
Use a neural network or max pooling to decode.

\[ \text{处处正确} \text{ need dynamic threshold.} \]

\[ \text{Decodes} \]

\[ \text{Line} \]

\[ \text{段落} \]

\[ \text{段落} \]

\[ \text{段落} \]

\[ \text{段落} \]

\[ \text{段落} \]

\[ \text{段落} \]

\[ \text{段落} \]

\[ \text{段落} \]