

KTH ROYAL INSTITUTE OF TECHNOLOGY



Middleware

Remote Invocation

Vladimir Vlassov and Johan Montelius

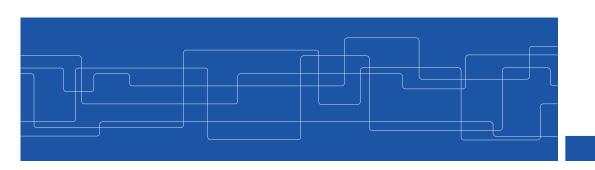


Remote invocation / indirect communication

Socket layer

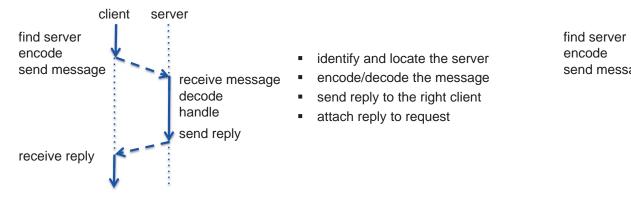
Network layer

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION



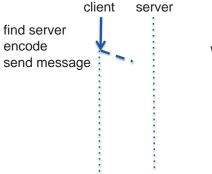


Request / Reply



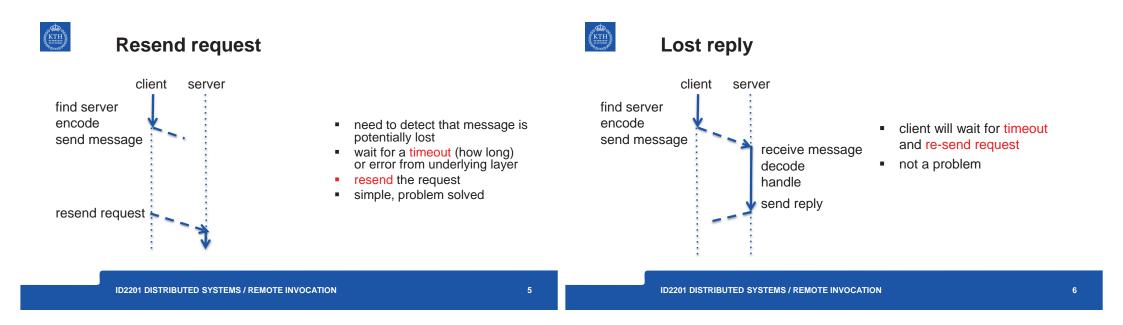


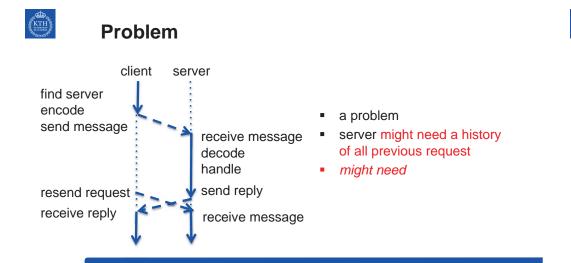
Lost request



What do we do if request is lost?

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION







Idempotent operations

- add 100 euros to my account
- · what is the status of my account
- Sweden scored yet another goal!
- The standing is now 2-1!



History

If operations are not idempotent, the server must make sure that the same request is not executed twice.

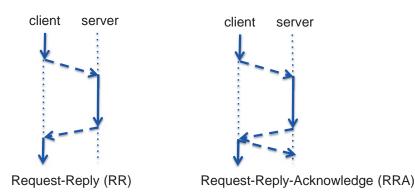
Keep a history of all request and the replies. If a request is resent the same reply can be sent without re-execution.

For how long do you keep the history?

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

(KTH)

Request-Reply-Acknowledge



ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

(KTH)

At-most-once or At-least-once

How about this:

If an operation succeeds, then..

at-most-once: the request has been executed once.

Implemented using a history or simply not re-sending requests.

at-least-once: the request has been executed at least once.

No need for a history, simply resend requests until a reply is received.



At most or At least

How about errors:

Even if we do resend messages we will have to give up at some time.

If an operation fails/is lost, then..

at-most-once:

at-least-once:



At most or At least

Pros and cons:

- at-most-once without re-sending requests:
 simple to implement, not fault-tolerant
- *at-most-once with history:* expensive to implement, fault-tolerant
- at-least-once: simple to implement, fault-tolerant

Can you live with at-least-once semantics?



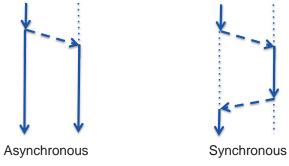
UDP or TCP

Should we implement a request-reply protocol over UDP or TCP?

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	13	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	14

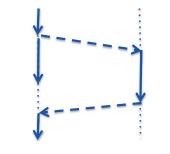
(KTH)

Synchronous or Asynchronous





RR over Asynchronous

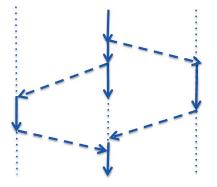


- send request
- continue to execute
- suspend if not arrived
- read reply

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION



Hide the latency





HTTP

A request reply protocol, described in RFC 2616.

Request = Request-Line *(header CRLF) CRLF [message-body]

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

GET /index.html HTTP/1.1\r\n foo 42 \r\n\r\nHello

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

17

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION



HTTP methods

- **GET**: request a resource, should be idempotent
- HEAD: request only header information
- POST: upload information to a resource, included in body, status of server could change
- PUT: add or replace a resource, idempotent
- DELETE: add or replace content, idempotent



Wireshark

72 9.474170000 130.277.28.40 130.277.28.140 TCP 00 85-53800 [Fm, 40] Seq0 44k-1 WinH180 Lm 73 9.474170000 130.277.28.40 130.277.28.40 TCP 50 5500-00 [Act] Seq1 4k-1 WinH180 Lm 75 9.474170000 130.277.28.40 130.277.28.40 TCP 50 5500-00 [Act] Seq1 4k-1 WinH180 Lm 75 9.47842000 130.277.28.40 130.277.28.40 TCP 50 5500-00 [Act] Seq1 4k-1 WinH180 Lm 75 9.478742000 130.277.28.40 130.277.28.40 TCP 50 55300-00 [Act] Seq4 6k-1300 WinH1842000 Lm 76 9.478742000 130.277.28.40 130.277.28.40 TCP 50 55300-00 [Act] Seq4 6k-1300 WinH1842000 Lm 77 seq 74: 000 bytes on vire (5500 bits), 000 bytes captured (5000 bits) on interface 0 Ethernet I/r Science Captured (5500 bits), 000 (53000), Det 130.277.28.40 (130.277.28.40) Frame Verbool WinH1842000 Lm 74 rmm Pixel bytes (100.000 bytes) Seq00 (53000), Det Fort: 00 (BO) (50000 Cortor): 2:::04) Frame Verbool WinH1842000 Lm 94 rmm Wint (100.000 bytes) Seq00 (53000), Det Fort: 00 (BO) (5000 Cortor): 2:::04) Fort 94 rmm Vint (100.000 bytes) Seq00 (53000), Det Fort: 00 (BO) (5000 Cortor): 2:::04) Fort 94 rmm Vint (100.000 bytes) Seq00 (5700 Cortor) Seq1 (100.0000 Cortor): 2:::04)		130.237.215.140	130.237.215.140 130.237.28.40	TCP	74 5396	0-80 [SYN	y response Oxa4c5 AAAA 2001:6b] Seq=O Win=29200 Len=O MSS=146
249_4454046000 100.227_315.140 100.227_316.10 100.227_316.							
75.9.478942000 130.227.23.9.40 130.227.23.9.40 TCP 356 [TCP segment of a reassembled POU] 75.9.478942000 130.227.23.9.40 TCP 545 2000-00 [AC] Second Ackingto Menna842048 L Frame 74: 090 bytes on vire (5502 bits), 090 bytes captured (5502 bits) on interface 0 terface 1 termin842048 L Finames Protocol Version 4, src: 180.237.215.140 (180.237.215.140), Dit: 141-099 routers 4(00000-00.071.8:04) termin842048 L Finames Protocol Version 4, src: 180.237.215.140 (180.237.215.140), Dit: 141-099 routers 4(00000-00.071.8:04) termin842048 L * Intermet Protocol Version 4, src: 180.237.215.140 (180.237.215.140), Dit: 141.045 termin842048 L * Mportext Transfer Protocol Version 4, src: 180.237.215.140 (180.237.215.140), Dit: 141.045 termin842048 L * Mportext Transfer Protocol Version 4, src: 180.237.215.140 (180.237.215.140), Dit: 141.045 termin842048 L * Mportext Transfer Protocol Version 4, src: 140.000 (str.410.000 (str.410.000) termin842048 L * Mportext Transfer Protocol Version 4, str.110, str.210.210, Str.210.0100 (str.410.000, Str.210.0100 (str.410.000, Str.200.01000 (str.410.000, Str.200.0100 (str.410.000, Str.200.0100 (str.410.000, Str.200.0100 (str.410.000, Str.200.0100 (str.410.000, Str.200.0100, Str.200.0100 (str.200.0100, Str.200.0100, Str.200.0100 (str.200.0100, Str.200.0100, Str.200.0100 (str.200.0100, Str.200.0100, Str.200.0100, Str.200.0100, Str.200.0100, Str.200.0100, Str.200.0100, Str.200.0100, Str.200.0100, Str.200.							
709.478872000 130.227.23.140 130.227.28.40 TCP 54.5380-40 [ACK] Seconds Acknows Ac							
<pre>Prame 74: 000 bytes on wire (5502 bits), 000 bytes captured (5502 bits) on interface 0 thermat II, Src: Assustatic_Boxcotd (000:1ext:000:dd), bet: All:stopperenters.dd (000:00:000:dd); thermatission control Protocol, Src: Port: 53000 (53000), bet: All:stopperenters.dd (000:00:000:dd); thermatission control Protocol, Src: Port: 53000 (53000), bet: Port: 80 (80), Seq: 1, Ack: 1, Len: 045 'Hyperext Transfer Protocol 'Got / HTD/Livin Mott Vancesting, app. dd: Mott Port: 53000 (53000), bet Port: 80 (80), Seq: 1, Ack: 1, Len: 045 'Hyperext Transfer Protocol 'Accept: test/Unit, application/interface, application/eniged.00; Ox600, Directory, Directory</pre>							
<pre># Etherner IT, Sriz Auwartekc_Baccobide (00:1e:0c:Baccobide). Det: All-PEPA-routers_64 (00:00:0c:00:ac:02:ac:edd) > Thermer Protocol Version 4, Briz: 10:207.20, 201.201.201.201.201.201.201.201.201.201.</pre>	76 9.478572000	130.237.215.140	130.237.28.40	TOP	54 5390	DHBD [ACK] Seg=646 Ack=305 Win=3842048 L
	User Agent: Notilla/5. Accept: text/html,appl	lication/shtml+sml, ap			srefox/40	.0\r\n	
	Accept-Encoding: gzip, ([truncated]Cookie: Connection: keep-alive trun [Full request UR1: htt [HTTP request 1/1]	. deflate\r\n utma=154244322.99916 e\r\n 1p://www.kth.se/]			389.6; mod	alVisitor	≫ll=participate#l ; csrftoken=
	Accept-Encoding: gzip, • [truncated]Cookie: Connection: keep-alive \r\n [Full request URI: htt [HTTP request 1/1] 0050 10 74 88 24 73 85	<pre>deflate\r\n utma=154244322.99916 e\r\n tp://www.kth.se/l 01 04 55 73 65 72 3</pre>	d 41 67 65 thise	Liber Age	388.6; mod	alVisitor	Poll-participate#11; csrftoken=
0060 Se 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 nt: Mozi lla/5.0	Accept-Encoding: gzip, • [truncated]Cookie: Connection: keep-alive VrN [Full request UR1: htt [HTTP request 1/1] 0050 de 74 34 20 44 6f	. deflate\r\n utme=154244322.99918 e\r\n 10://www.kth.se/l 	d 41 67 65 5 2e 30 20 nti Nez	User-Age	888.6; mod	alVisitor	≫ll∝participate#ll; csrftoken=
0000 6a 74 3a 20 4d 6f 7a 60 6c 6c 6l 2f 25 2a 20 20 nti Mozi Ila/5.0 0070 28 58 31 31 36 20 55 62 75 6a 74 75 36 20 4c 60 (X11; 06 untu; La	Accept-Encoding: gzip, (truncated)Cookie: Connection: keep-alive Vrin [Full request URI: htt [HTTP request 1/1] 0050 ds 74 ds 20 4d df 0070 ds 58 33 31 26 50	. deflate\r\n otma=154244322.99918 prinn tp://www.kth.se/1 00 0% 56 73 65 72 3 74 69 6c 6c 61 2f 3 76 69 76 66 74 75 3	d 41 67 65 http: 5 2# 30 20 nti Moz 5 20 4< 69 ()11; 0	Liser-Age 1 lla/5.0 6 untu: La	388.6; mod	alVisitor	>>lloparticipate#l[; corftoken=
0080 de 74 3a 20 4d df 7a 60 6c 6c 61 2f 25 2e 20 20 nt: Mezi lla/5.0 0070 28 58 31 31 36 20 55 62 75 6e 74 75 36 20 4c 69 (X11; U6 untu; Li	Accept-Encoding: gaip, (Iruncated)Cookie: Connection: keep-alave Irun [Full request LRI: htt [HTTP request 1/1] 0050 6 874 68 as 73 65 0060 6 874 84 as 24 d f 0070 26 59 31 31 26 20 0080 6 87 78 30 78 28	deflate\r\n utma=154244322.99916 e\r\n tp://www.kth.te/l 7a 60 6c 6c 61 21 3 65 62 75 6e 74 75 3 55 63 a 34 36 20 7	d 41 67 65 ith set 5 2e 30 20 nt Moz 5 20 44 60 (X11) 0 276 2e 34 rus s00	User-Age 1 Lla/5.0 6 untu: La 64; rv:4	389.6; mod	alVisitor	boll∝participate#1/; csrftoken∘
00050 de 74 3a 20 4d 6f 7a 680 6c 6c 61 2f 25 2a 30 20 vit. Huzi lla/5.0 0070 2d 58 31 31 36 20 55 62 75 6e 74 75 36 30 4c 69 (Xili 0 burniu Li 0006 6e 75 72 20 71 38 36 75 30 34 36 20 72 76 3a 96 mark 800 64 rive	Accept-Encoding: gaip, [fruncated]Coskie: Connection: keep-alave Vrun [hill request 1/1] [HTP request 1/1] [0500 G 74 Da 20 44 df] 0070 G 53 13 12 50 20 0080 G 77 78 20 78 25 0080 G 75 20 25 20 45 0090 D 0 78 20 25 20 45 0000 D 0 78 20 25 0 78 20 45 0 78 20 0 78 20	. deflate\r\n utma=154244322.9991f virun tp://www.kth.se/l 	d 41 67 65 kthise 2a 30 20 kthise 5 20 46 69 (XII) 0 2 76 2a 34 ktor 3 30 20 0 0.0 68	User-Age 1 lla/5.0 b untui Li 54: rv:4 c ke/20100	389.6; mod	alVisitor	≫ll-participate#ll; csrftoken≉

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

18



HTTP GET

GET / HTTP/1.1 Host: www.kth.se User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:40.0) Gecko/20100101 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept.Encoding: gzip, deflate Cookie:

Connection: keep-alive



HTTP Response

HTTP/1.1 200 OK Date: Tue, 08 Sep 2015 10:37:49 GMT Server: Apache/2.2.15 (Red Hat) X-UA-Compatible: IE=edge Set-Cookie: JSESSIONID=CDC76A3;Path=/; Secure; HttpOnly Content-Language: sv-SE Content-Length: 59507 Connection: close Content-Type: text/html;charset=UTF-8 <!DOCTYPE html> <html lang="sv"> <title>KTH | Valkommen till KTH</title>

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

21

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

(KTH)

The web

On the web the resource is often a HTML document that is presented in a browser.

HTTP could be used as a general-purpose request-reply protocol.



REST and SOAP

Request-reply protocols for Web-services:

- REST (Representational State Transfer)
 - content described in XML, JSON, . . .
 - · light weight,
- SOAP (Simple Object Access Protocol)
 - over HTTP, SMTP . . .
 - content described in SOAP/XML
 - standardized, heavy weight



HTTP over TCP

HTTP over TCP - a good idea?



Masking a request-reply

Could we use a regular program construct to hide the fact that we do a request-reply?

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	25	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	26

(KTH)

Masking a request-reply

Could we use a regular program construct to hide the fact that we do a request-reply?

• RPC: Remote Procedure Call

• RMI: Remote Method Invocation



Procedure calls

What is a procedure call:

- find the procedure
- give the procedure access to arguments
- · pass control to the procedure
- · collect the reply if any
- continue execution

How do we turn this into a tool for distributed programming?



Operational semantics

int x, n;	int x, arr[3];
n = 5;	arr[0] = 5;
proc(n);	proc(arr);
x = n;	x = arr[0];



Call by value/reference

Call by value

- A procedure is given a copy of the datum

Call by reference

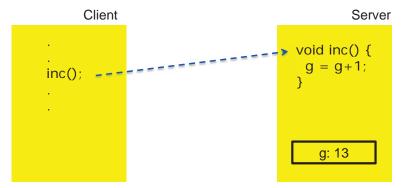
- A procedure is given a reference to the datum

What if the datum is a reference and we pass a copy of the datum? Why is this important?

		ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	29	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION	30
--	--	--	----	--	----



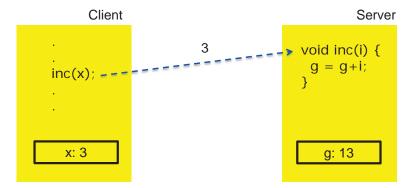
RPC: Remote Procedure Call



(KTH)

31

RPC: Remote Procedure Call



ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

(KTH)	RPC: Remote Procedure Call			Open Network Computing (ONC) RPC (Sun		
	Client ; inc(a); ; a: {1,2,3,4}	Server void inc(int[] h) { g = g+h[2]; h[2] = g; } g: 13		 targeting intranet, file servers etc at-least-once call semantics procedures described in Interface Definition Language (IDL) XDR (eXternal Data Representation) specifies message structure used UDP as transport protocol (TCP also available) 		
	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION		33	ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION		



Java RMI (Remote Method Invocation)

- similar to RPC but:
 - we now invoke methods of remote objects
 - at-most-once semantics
- Objects can be passed as arguments, how should this be done?
 - by value
 - by reference



Java RMI

We can do either:

A *remote object* is passed as a reference (*by reference*) i.e. it remains as at the original place where it was created.

A *serializable object* is passed as a copy (*by value*) i.e. the object is duplicated.



Finding the procedure/object

How do we locate a remote procedure/object/process?

Network address that specifies the location or..

a known "binder" process that keeps track of registered resources.

Remote invocation design decisions

- failure handling: maybe / at-most-once / at-least-once
- call-by-value / call-by-reference

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

- message specification and encoding
- specification of resource
- procedure binder

ID2201 DISTRIBUTED SYSTEMS / REMOTE INVOCATION

37

Examples

- SunRPC: call-by-value, at-least-once, IDL, XDR, binder
- JavaRMI: call-by-value/reference, at-most-once, interface, JRMP (Java Remote Method Protocol), rmiregistry
- Erlang: message passing, maybe, no, ETF (External Term Format), local registry only
- CORBA (Common Object Request Broker Architecture): call-by-reference, IDL, ORB (Object Request Broker), tnameserv
- Web Services: WSDL (Web Services Description Language), UDDI (Universal Description, Discovery, and Integration)



Summary

Implementations of remote invocations: procedures, methods, messages to processes,

have fundamental problems that needs to be solved.

Try to see similarities between different implementations.

When they differ, is it fundamentally different or just implementation details.