

1 Sistema Banatuak - Sarrera

Edukia

1.1 Sistema banatuen zergatia	2
1.1.1 Sistema motak	2
1.1.2 Sistema banatuen definizioa	4
1.1.3 Sistema banatuen abantailak eta arazoak	5
1.1.4 Joerak	6
1.2 Sistema banatuen propietateak	7
1.2.1 Gardentasuna	7
1.2.2 Eskalagarritasuna	8
1.2.3 Fidagarritasuna eta hutsegite tolerantzia	9
1.2.4 Sendotasuna	10
1.3 Aplikazio banatuak	11
1.4 Hardware euskarria	12
1.5 Baliabideen banaketarako software euskarria	15
1.5.1 Sistema irekiak	15
1.5.2 Komunikaziorako euskarria	17
1.5.3 Sistema eragilearen euskarria	21
1.6 Sistema banatuen egitura	22
1.7 Ariketak	25
1.8 A eranskina: Urruneko prozeduren deiak (RPC)	27
1.9 B eranskina: Urruneko metodoen deiak (RMI)	30

1.1 Sistema banatuen zergatia

Sistema banatuak, sistema informatikoen garapenean, beste urrats bat dira, alde batetik aplikazioek dituzten beharreatatik, eta bestetik teknologiak eskaintzen dituen posibilitateengatik. Sistema banatuen definizio bat eman baino lehen, ikus dezagun sistema informatikoen eboluzio historikoaren bitartez, gaur egungo sistema banatuetan oinarri diren hainbat kontzeptu, ezaugarri nagusietariko bat elkarloturiko konputagailuetan baliabideen banatze fisikoa eta konpartitza delarik.

Era orokorrean, **baliabide** terminoa erabiliko dugu banatu daitekeen edozein dispositibo edo zerbitzu, hardware edo software naturazkoa, adierazteko.

1.1.1 Sistema motak

Ikuspegi historiko batetik, konputaziorako sistemak sailkatzerakoan eredu desberdinetaz hitz egin daiteke. Eredu hauek sistemen funtzionalitatea eta egitura baldintzatzen dituzte, baita bere sistema eragilearen ezaugarriak baliabideen kudeatzaile gisa ere, eta baita bere aplikazio eta erabilpen eremua ere.

- **Lotekako sistemak.** Sistema hauek lehen sistema eragileak dira, aurrez prestatutako txartel zulatuez osaturiko datu paketeak era diferitu eta sekuentzialean prozesatzen zituztelarik. Gaur egun dauden sistema multiprogramatuekin, bere erabiltzaile-interfaze interaktiboekin, lotekako sistemak kalkulu intentsiboko aplikazioetan erabiltzen dira bakarrik, adibidez superkonputazioan.
- **Denbora banatuko sistema zentralizatuak.** Hurrengo urratsa izan zen, 60. hamarkada erdialdean. Sistema hauen helburua PUZaren erabilpen maila (eraginkortasuna) handitzea zen, garai hartan oso baliabide garestia eta urria baitzen. Horretarako, erabiltzaileek interaktiboki lan egiten dute, erantzun denbora gutxitu egiten delarik. Baliabide guztiak zentralizatuak daude, eta sistema atzitzeko terminalak erabiltzen dira.
- **Teleprozesuko sistemak.** Aurreko ereduarekin konparatuz, sistema hauetan terminalak —gaur egun sistema pertsonalak— urruti daude, eta sare azpiegitura baten bidez (adibidez, sare telefonikoa) sistema zentralizatu bat atzitzen dute, normalean itxia den komunikazio protokolo bat jarraituz. Sistema zentralizatu honek baliabide guztiak kudeatzen ditu. Eredu honetan oinarritutako aplikazioen adibideak bidai erreserba edota banku transakzio sistemak ditugu.

- **Sistema pertsonalak.** Sistema mota honen zergatia erabiltzaile bakar bati sistema dedikatu bat eskaintzean datza. Hau posiblea izan zen hardwarearen prezioa asko jaitsi zelako 80. hamarkadaren hasieran, mikroprozesadorearen sorrerari esker. Bere prezio txikia da, hain zuzen ere, sistema hauen ezaugarri nagusia. Hasiera batean, sistema pertsonalen (PCen) sistema eragilea erabiltzaile bakarrekoa zen, babes mekanismorik gabe. Sinpletasunagatik, hasierako sistema eragileak monoprogramatuak ziren (MS-DOS). Gaur egun, aldiz, hardwarearen hobekuntzak sistema eragile multiprogramatuak (multiatazak) izateko aukera ematen du (MacOS, OS/2, Windows95/98/Me/XP), baita denbora banatuko sistema eragileak ere, UNIX eta WindowsNT bezalakoak¹. Sistema pertsonal batek bere baliabide lokalak ditu. Hasiera batean, hauek ziren erabil zitezkeen baliabide bakarrak, baina gaur egun egoera aldatu egin da konputagailu sareei esker. Bestalde, hardwarearen eboluzioak ordenagailu pertsonal mugikorrek ekarri digu, hala nola PC eramangarriak, PDAk edota telefono mugikorrek.
- **Sareko sistemak.** Kasu honetan, sistema pertsonal batzuk sare azpiegitura baten bidez konektatzen dira. Teleprozesuko sistemen eboluzio bezala ere ikus daiteke, non terminalak konputazio ahalmena eta funtzionalitateak handitzen doazen, sistema autonomoak izateraino. Sareko sistemetan konputagailu zentral eta nagusiaren kontzeptua desagertzen da, eta edozein konputagailu beste edozeinekin zuzenean komunikatzen da sarearen bidez. Makina bakoitzak bere sistema eragilea du. Baliabide konkretu bati sarbidea eskaintzen duen makinak baliabide horren **zerbitzaria** dela esaten da, besteak **bezeroak** direlarik. Bezeroek **baliabide lokalak** izan ditzakete ere, baita beste makinetan dauden **urruneko baliabideak** erabili, dagokion zerbitzariari eskaera eginez. Nahiz eta askotan ezaugarri eta sistema eragile desberdinak izan, makina guztiak beraien artean komunikatzeko ahalmena protokolo ireki eta komun erabilpenean oinarritzen da (interoperatibitatea), TCP/IP ezagunena delarik. Protokolo ireki eta komun erabilerak **hedapen zabaleko sareak** eraikitzea posiblea egin du, hala nola **Internet** famatua. Badira bestalde itxiak diren protokoloak ere, komunikazioa mota

¹ Sistema mota hau **lan-estazioa** edo *work-station* izenez ezagutzen zen 90. hamarkadaren hasieran. Gaur egun PCak denbora banatuko sistema eragileak exekutatzeko gai dira, eta ez da horrelako bereizketarik egiten. PCa osagai estandarra bihurtu da, hainbat erabilpen desberdinerako balio duena, hala nola sistema pertsonal gisa, baliabide konpartituen zerbitzari gisa, edota errendimendu handiko sistema bateko elemento gisa (*cluster*).

berdineko makinaren artean mugatzen dutenak, adibidez Novell PCentzako, edota AppleTalk Macintoshentzako.

- **Sistema banatuak.** Sare baten bidez konektatutako makinaren baliabideak *integratu* egiten dira, lokala/urruna dualtasuna desagertzen delarik. Sareko sistemekin konparatuz, desberdintasun nagusia aplikazio eta erabiltzaileentzat baliabideen kokapena *gardena* dela da. Ikuspuntu honetatik, denbora banatuko sistema zentralizatuekin ez dago alderik. Erabiltzaileari interfaze grafiko bat aurkezten zaio bere terminal grafikoan edo lan-estazioan, sistema banatuko baliabideak atzitzeko aukera ematen diona, baliabideen kokapenaz arduratu gabe. Aplikazioek sistema-deiak egiten dituzte sistema zentralizatu baten gainean egongo balira bezala, adibidez POSIX araua erabiliz. Urruneko deiak egiteko zerbitzu batek (prozeduretara zuzenduta, adibidez RPC, edota objektutara zuzenduta, adibidez RMI) baliabide urrunen atzipenak ahalbidetzen ditu, horretarako sare interfazea erabiliz. Sistema banatuek baliabideen konpartitzea era gardenean eskaintzen dute, atzipena eta kudeaketa erraztuz, eta eraginkortasuna eta eskuragarritasuna hobetuz.

Erreparatu berri ditugun sistemen artean sistema banatuen eredia orokorra izanik, gaur egungo joera sistema hauen aldekoa da, nahiz eta oraindik merkatu mailan baliabide mota guztientzako integrazio berdina lortzea dagoen. Beste arrazoi nagusia sistema hauen alde azken urteetan eboluzio teknologikoak suposatu dituen kostu erlazioak dira. Gaur egun kostu txikiko hardware estandarra dugu, ordenagailu pertsonalak (PCak), sistema banatuetako oinarritzko osagaiak izaten direnak. Bestalde, komunikazio sarearen kostua ere nahiko merkea da, baldin eta oso prestazio altuak ez badira behar. Esate baterako, kasu batzutan oinarritzko sare telefonikoa erabil daiteke (edota sare elektrikoa ere). Bestalde, haririk gabeko sareak gero eta gehiago erabiltzen dira.

1.1.2 Sistema banatuen definizioa

Goazen bada sistema banatuen definizio posible bat ematera:

- (1) konputagailu multzo bat,
- (2) elkarkonektatuta daudenak,
- (3) egoera bat konpartitzen dutenak,
- (4) sistema bakarraren ikuspegia emanez.

Alde batera utzita *konputagailu* terminoaren esanahiaren eztabaida, (2) ezaugarria sistema banatuek eta sareko sistemek konpartitzen dute, eta sistema pertsonalengatik bereizten ditu. Aldiz, (3) ezaugarria sistema banatuena propioa da, bere ondorioa (4) sistema bakarraren ikuspegia² delarik, baliabideak homogeneoki aurkeztuz, bere banaketa gordez: *erabiltzaileak eta aplikazioek ez dute sarerik ikusten, baizik eta sistema zentralizatu batengandik bereizteza den sistema bat*. Sareko sistemak, egoera independentea duten konputagailu multzo bat bezala ikus daitezkeen moduan, sistema banatuetan **egoera globala** definitzen da.

Sistemaren topologia eta sarearen ezaugarri fisikoak sareko protokoloek gordetzen dituzte. Makina bakoitzaren arkitekturari dagokionez, sistema eragileak gordetzen du. Sistema banatuen osagaiak heterogeneoak izan daitezkeenez, software maila bat behar da (*middleware* izenez ezagutzen dena) sistema bakarraren ikuspegia emateko. Ideia honi buruz berriro hitz egingo dugu sistema banatuen egitura aztertzerakoan.

1.1.3 Sistema banatuen abantailak eta arazoak

Sistema banatuen definiziotik sistema zentralizatuekiko eta sareko sistemekiko abantaila batzuk ondorioztatu dira. Sistema zentralizatuekiko, sistema banatuek honako abantailak dute (sareko sistemekin konpartituak):

- Kostu txikia. Sistema pertsonal (PC) estandarrez osatuta egon daitezke, eraginkortasun/kostu erlazio on bat emanez.
- Handitzeko ahalmena (eskalagarritasuna), duten modulartasunaren ondorioa.
- Malgutasuna. Betebehar batzuentzako zaharkituta gelditutako makinak exijentzia txikiagoko lanetarako erabilgarriak izaten jarrai dezakete.
- Eskuragarritasuna. Baliabideen errepikapenaren ondorioa.
- Paralelotasuna ahalbidetzen dute.
- Baliabide urrunak atzitzea ahalbidetzen dute.

Sareko sistemekiko, sistema banatuek:

² *Single System Image*, SSI [BAK00].

- Aplikazioek eskuragarri dituzten baliabideen erabilpen eraginkorragoa egitea ahalbidetzen dute.
- Baliabideen atzipen gardena ahalbidetzen dute, bere kokapenaz arduratu gabe.

Sistema banatuek badituzte bere arazoak ere. Esate baterako, sistema zentralizatuekiko honako desabantailak aipa daitezke:

- Kostu berdineko sistema zentralizatu batek sistema banatuko osagai bakoitza baino prestazio hobea emango du, komunikazio euskarririk behar ez duelako. Honek garrantzia izan dezake kalkulu aplikazioetan, izaera sekuentziala dutenak.
- Komunikazio kostua kontuan hartu behar denez, baliabideen banaketa fisikoa desegokia bada, baliabide batzuk oso gutxi erabiliak egotera eta beste batzuk itota egotera eraman gaitzake.
- Egoera globala sendoa mantentzea kostu handia izan dezake, eraginkortasun ikuspuntutik.
- Komunikazio sarea arazo iturri berri bat da.
- Segurtasunaren kudeaketa konplexuagoa da.

1.1.4 Joerak

Informatikaren inguruko hainbat arlotan gertatzen ari den eboluzio teknologikoa --prozesadoreak, informazioaren gordeketa, sareak (haririk gabekoak bereziki), bateriak, pantailak--, eta ondorioz agertzen ari diren gailu berriak, testuinguru berri bat jartzen du sistema banatuetako aplikazioentzat, aukera zabalagoak emanez. Eboluzio hauetariko bat, 90. hamarkadan hasitakoa ordenagailu eramangarriekin batera, **informatika mugikorra** izenez ezagutzen den arloa da: ordenagailu eramangarria jatorrizko sare lokaletik atera egiten da, baliabide berriak aurkituz. Sarearen egoera aldakorretara egokitu behar da, eta kasu batzuetan konexiorik gabe geldi daiteke, funtzionatzen jarraitu behar duelarik. *Ad-hoc* motako sareak era espontaneoan eratu daitezke, protokolo espezifikoak behar dituztenak, adibidez, Mobile IP [PER98]. Testuinguru honetan, nabariak dira haririk gabeko komunikazioen abantailak.

Urrats bat aurrerago **sistema perbasiboak** aurkitzen ditugu (*ubiquitous systems*). Hardwarearen eboluzio ikaragarriari esker, konputazio gailuak gero eta txikiagoak dira. Konputagailu portatilen presentzia nabaritzen ez denean

perbasiboak bihurtzen direla esaten da (ezkutatzen den teknologia, *disappearing computing* [WEI91]). Bestalde, gailu asko konputagailu baten ahalmena izatera iritsi dira, bai konputazio mailan eta baita komunikazio mailan ere. Gailu hauen gaur egun edota etorkizun laburrean izango diren adibide bezala aipa daitezke: agenda elektronikoa (PDAk), telefono mugikorrak, automobilen nabigazio sistemak, txartel inteligenteak, giltzak, erlojuak... Ingurune perbasiboak izaeraz aldakorrek dira. Baliabideak eta zerbitzuak dinamikoki aurkitu eta konfiguratu dira, aplikazioak eta interfazeak inguruneari egokitu egiten direlarik.

Sistema banatuak eta informatika mugikorrek planteatzen dituzten arazoetatik aparte, informatika perbasiboak erroka berriak irekitzen ditu hainbat arlotan, besteak beste hardwarea, sareak eta komunikazio protokoloak (bereziki gailuen aurkikuntzarako, Jini³ eta UPnP⁴ bezalakoak), erabiltzaileentzako interfazeak (gailu berriak interakzio modu berriak eskatu dezakete), pribazitatea eta segurtasuna. Sistema perbasiboan aplikazio eremua oso aberatsa da: etxea (domotika), automobila, merkataritza eta industria, laguntza zerbitzuak (adibidez, 3. adina), hezkuntza, robotika...

1.2 Sistema banatuen propietateak

Sistema bakarraren ikuspegia eman nahi duen sistema banatu batek honako propietateak izan beharko lituzke⁵.

1.2.1 Gardentasuna

Sistema banatu baten oinarrizko helburua erabiltzaileari eta aplikazioei baliabide guztiak alegiazko makina bakar batek kudeatuta bezala aurkeztea da. Baliabideen banaketa fisikoa gardena da. Gardentasun mota desberdinak aipa daitezke:

- **Identifikazioan.** Baliabideen **izen-esparruak** sarearen topologiarekiko edota banaketa fisikoarekiko independenteak dira.

³ www.jini.org

⁴ www.upnp.org

⁵ Propietate hauek *desiragarriak* bezala ulertu behar dira. Behin baino gehiago ikusiko denez, praktikan eraginkortasun ona lortzeko propietate hauetako batzuk erlaxatu egiten dira.

- **Baliabideen kokapen fisikoan.** Ez erabiltzaileek, ezta aplikazioek ez dakite zein makinatan dauden kokaturik atzitzen ari diren baliabideak, edota lokalak ala urrunak diren. Honek suposatzen du baliabideak **tokiz aldatu** daitezkeela (*migration*) makinaren artean, aplikazioetan eraginik izan gabe.
- **Erreplikazioan.** Erabiltzaileek eta aplikazioek ez dakite baliabide bakoitzeko zenbat unitate dauden, ezta erreplikarik gehitu ala gutxitu egiten diren dinamikoki. Sistema banatuetan erreplikazioak **hutssegiteen gardentasuna** ere inplikatu du, hau da, erreplikaren bat zerbitzuz kanpo gelditzea ezkatu egiten da. Propietate hau, *hutssegite tolerantzia* izenez ezaguna, bere aldetik aztertuko dugu aurrerago.
- **Paralelotasunean.** Baliabideen erreplikazioaren beste ondorio bat aplikazioak paraleloan exekuta daitezkeela da, aplikazioak ezer adierazi behar ez duelarik, eta exekuziorako inongo ondoriorik gabe, errendimendu kontuak alde batera utzita.
- **Konpartizioan.** Aplikazio bat baino gehiago konpartitutako baliabide bat aldi berean atzitzen saiatzeak ez du eraginik aplikazioen exekuzioan (sistema bakar batekin konparatuz).
- **Eraginkortasunean.** Sistema banatuen propietate guztiak eskaintzea errendimendu galera bat dakar nahitaez. Adibidez, baliabideen identifikazioa eta kokapen gardena lortzeko askotan izenen ebazpena era urrunean egitea eskatzen du, ebazpen-denbora edo latentzia handituz. Beraz, errendimendu galera zaildu egiten duenean propietateren bat lortzea, erdibideko edo konpromisozko soluzioak aurkitzea ezinbestekoa da.

1.2.2 Eskalagarritasuna

Sistema banatuen ezaugarrietariko bat bere **modulartasuna** da. Honek malgutasuna ematen die, eta bide batez **eskalagarritasuna** ere, azken hau honela defini daitekeelarik: *sistema batek handitzeko duen ahalmena, bere konplexutasuna handitu gabe eta bere eraginkortasuna kaltetu gabe*. Sistema banatuak diseinatzerakoan helburuetariko bat eskalagarritasuna zerbitzuen integrazioa hedatzea da.

Eskalagarritasunak bi alderdi nagusi ditu. Alde batetik, sistema banatuak (1) tamaina egokiko **izen-esparruak** izan behar dituzte, izen-esparrua bera muga bat izan ez dadin sistema handitzerakoan, eta bestalde (2) errendimendu maila

on bat mantendu behar dute baliabideen atzipenean sistema handitzen denean (beharbada erreplikazioaz baliatuz).

- **Izen-esparruak.** Izen-esparruak, sistema zentralizatueta bezala, natura desberdineko objektuak identifika dezakete, hala nola fitxategiak, prozesuak, aldagaiak, edota memoria helbideak (*memoria banatua konpartitua* sistemetan, DSM). Esparru linealen kasuan, adibidez memoria, muga inplizitua dago izenaren tamainari lotuta, gaur egun 32 biteko helbide-esparruak nahikoak ez direla begi-bistakoa delarik. Orokorrean izen-esparruak hierarkikoak dira, eta beraz eskalagarriak izaeraz.
- **Errendimendua.** Sistema banatuen handitzeak itoguneak eta latentzia handiagoak sor ditzakete, sistemaren errendimendua kaltetuz. Hau ekiditeko erreplikazioa erabili daiteke. Kasu batzuetan, baliabidea *makina lokalean* erreplikatu da aldi bereko atzipenak ahalbideak, zerbitzuaren errendimendua hobetuz; beste batzuetan baliabideen banaketa geografikoa erabiltzen da (*mirroring*), edota makina lokalean baina behin-behineko izaeraz gordetzen da baliabidea (*caching*), atzipenak deszentralizatuz eta lokaltasuna bultzatuz. Edozein kasutan, erreplikazioak, neurri batean edo bestean, konplexutasuna ekartzen du: kopia guztien arteko sendotasuna kudeatzeko beharra (gardentasuna erreplikazioan), honek dakarren errendimendu galerarekin. Beraz, erdibideko soluzioak erabili beharko dira praktikan.

1.2.3 Fidagarritasuna eta hutsegite tolerantzia

Sistema baten fidagarritasuna bere eginkizuna une oro zuzen egiteko ahalmena bezala ikus daiteke. Fidagarritasuna bi alderditan konkretatuko dugu:

- **Eskuragarritasuna.** Sistema funtzionamenduan dagoen denbora portzentaia bezala definitzen da. Eskuragarritasuna neurtzeko parametro nagusia hutsegiteen arteko batezbesteko denbora izaten da (MTBF), konpontze denbora ere kontuan hartu behar delarik. Eskuragarritasuna bi modutan handitu daiteke: (a) kalitate handiagoko osagaiak erabiliz, edota (b) erreplikazioan oinarritutako diseinuak eginez, osagaien batek huts egin arren sistema funtzionatzen jarraitu dezan. Bi teknikak, konbinatu daitezkeenak, sistemaren kostua handitzen dute. Gaur egun dugun egoera teknologikoan erreplikazioa da, orokorrean, bi soluzioetatik merkeena. Sistema banatuek, bere izaeraz, baliabide batzuen erreplikazioa eskaintzen dute (adibidez prozesaketa unitateak, CPUak), eta normalean konpartituek diren

baliabideak (adibidez fitxategi zerbitzari bat) errepikatu egin daitezke eskuragarritasuna handitzeko. Bestalde, sistemaren osagaietan hutsegiteak gertatuko ez direnik, bai hardware eta bai software aldetik, ezin da inoiz ziurtatu, beraz neurri batetik gora eskuragarritasuna handitzeko gelditzen den bide bakarra erreplikazioa da. Hutsegite probabilitatea erreplikazioaren funtzio esponenzial bezala jaisten da. Adibidez, osagai baten hutsegite probabilitatea %1 bada (denbora tarte zehatz batean), osagaiaren 4 erreplika erabiliz guztien hutsegite probabilitatea %0,000001⁶ baliora jaitsiko litzateke.

- **Hutsegite tolerantzia.** Oso eskuragarritasun handia izanda ere, une kritikoetan hutsegite batek oso ondorio larriak izan ditzake. Pentsa dezagun denbora errealeko sistema kritikoetaz, bizitzetan eragina izan dezaketen gailuak kontrolatzen dituztenak (adibidez medikuntzan, zentral nuklearretan...). Nahiz eta erreplikazioak eskuragarritasuna handitzen duen, ez du berez zerbitzuaren funtzionamendua era gardenean beti bermatzen. Osagaien hutsegiteen aurrean, hutsegite tolerantziak sistemak zuzen funtzionatzen jarraitzeko duen ahalmena adierazten du, hutsegiteak erabiltzaileari edota aplikazioei ezkutatzuz. Beraz, hutsegite tolerantziak (1) hutsegiteak detektatzea eta (2) zerbitzuaren jarraipen zuzena eskatzen ditu.

1.2.4 Sendotasuna

Baliabideen erreplikazioak abantaila handiak ditu. Alde batetik, errendimenduaren hobekuntza laguntzen du, paralelotasunaren bidez eta baliabideen erreplika lokalean atzipenak hobetsiz (aplikazioen komunikazio kostua jaitsiz). Bestetik, ikusi berri den bezala eskuragarritasuna handitzen du, eta hutsegite tolerantziaren oinarria da. Dena dela, baliabideak errepikatzeak arazo batzuk ere baditu:

- Baliabide errepikatuen elkarkonexioa hutsegite iturri berri bat da.
- Sistemaren segurtasuna, informazioaren pribazitateari dagokionez, hausgarriagoa da baimenik gabeko erasoaren aurrean. Erreplika bakoitza independenteki babestu behar da.

⁶ Hau matizatu egin behar da: alde batetik, erreplika bakoitzaren hutsegitearen artean korrelazioa egon daiteke arrazoi desberdinengatik, adibidez erroreak diseinuan edota hondamendi naturalak. Bestalde, errepliken arteko konexioa hutsegite iturri gehigarria da.

- Sistemaren egoera globalaren kudeaketa konplexuagoa da, errepliken arteko sendotasun gabeko egoerak ekidin behar direlako. Gai hau garrantzi handikoa da sistema banatuak diseinatzerakoan, ondorioz arretea bereziaz aztertuko dugu.

Arazoa, osagai batzuk errepikaturik dituen sistema banatuan, egoera global sendoa mantentzeko beharrez datza, erreplika bakoitza bere egoera lokala duelarik. Erreplikak fisikoki banaturik daudenez, egoera globalaren kudeaketa komunikazio mekanismoen menpekotasuna izango du, azken hauek hutsegitek izan ditzakeen sare baten gainean funtzionatzen dutelarik. Gainera, gehienetan errepliken banaketa fisikoak ez du ahalbidetzen erloju global baten erabilera (denbora erreferentzia absolutuak emango lituzkeena) sistemako gertaera guztiak ordenatu ahal izateko, eta beraz erreplika bakoitzaren egoera trantsizioenak ere.

Sendotasunaren kudeaketa sistemako makinaren arteko komunikazioa eskatzen du normalean. Hau dela eta, batzutan hobe izaten da sendotasun maila erlaxatzea, sistemaren eraginkortasuna maila on batean mantentzeko, aplikazioen beharren arabera.

1.3 Aplikazio banatuak

Hasteko, aplikazio banatuak eta aplikazio paraleloak bereiztea komeni da. Aplikazio paraleloak, prozesaketa unitate desberdinetan konkurrenteki exekutatu daitezkeen atazetan zatitzen dira, exekuzio denbora ahalik eta txikiena izateko helburuarekin. Esan beharra dago aplikazio gehienak neurri batean edo bestean paraleloan exekutagarriak direla, konputazio eredu zehatz batzuei jarraituz (adibidez, *pipeline* eredua), aplikazioaren izaeraren eta exekutatzeko erabiliko den hardwarearen arabera. Aplikazio paraleloen atazak prozesaketa unitateetan banatzerakoan, unitate bakoitzaren zama eta komunikazio kostuak hartzen dira irizpide gisa. Aplikazio paraleloen helburu nagusia atazen exekuzio konkurrentea dela esan daiteke.

Aplikazio banatuak, aldiz, oso motibazio desberdinak izan ditzakete:

- **Errendimendu handia.** Aplikazio paraleloak banatuak ere izan daitezke. Adibidez, sare lokal bat erabil daiteke atazak sareko makinaren artean banatzeko, konputazio baliabideak (gehienetan kostu txikiko PCak) ahalik eta hoberen erabiliz, exekuzio denborarik txikiena lortzeko. Hain zuzen ere, era honetako konputazio eskemak (*cluster computing*) errendimendu/kostu erlazio oso onak eskaintzen dituzte gaur egun, gero eta hedatuagoak direlarik ohizko superkonputagailuen orde. Nahiz eta

oraindik mezu-truke esplizituan oinarritutako sistemak nagusi izan (adibidez, MPI), banaketa mekanismo gardenik gabekoak, joera konputazio baliabideen integrazio gardenaren aldekoa da. Zentzu honetan sistema berriak agertzen ari dira (adibidez, MOSIX⁷). Zentzu berean, memoria banatua konpartitua (DSM) integrazio osoa lortzeko erronketariko bat bezala ikusten da.

- **Hutsegite tolerantzia.** Beste aplikazioetan banaketaren arrazoia informazioaren integritatea izaten da. Adibidez, banketxeetako sistema informatikoetan bezeroen kontuen egoerari buruzko informazioa makina desberdinetan errepikatuta mantentzea ezinbestekoa da, zeren eta bestelakoan makina bakar baten hutsegitearen ondorioak (informazioaren galera) onartezinak izango lirakeke. Sistema hauetan garrantzi handikoa da errepliken eguneraketa era sendoan egitea. Gaur egungo sistema komertzialek teknika nahiko kontserbadoreak erabiltzen dituzte, batez ere errendimendu arrazoiengatik. Hala ere, ikusiko dugun bezala, erreplikazioaren kudeaketa sendoaren oinarri teorikoak ondo ezarrita daude.
- **Eskuragarritasun handia.** Azkenik, badira aplikazioak non banaketa edota erreplikazioa erabiltzaileari informazioa gerturatzeko erabiltzen den, erantzun denbora ahalik eta txikiena izateko. Aplikazio hauetan kokapen geografikoa kontuan hartzen duen erreplikazio teknikak erabiltzen dira (*caching*, *mirroring*). Sendotasuna eguneraketetan garrantzitsua da, baina ez kritikoa (banketxeetako sistemekin konparatuta). Aldiz, sistema hauek oso eskalagarriak izan behar dira.

Aplikazio banatuak eskenatoki edo ingurune desberdinetan exekutatu daitezke. Gaur egun dugun ingurunerik orokorrena Internet da (*World-Wide-Web*-a, *peer-to-peer* edota P2P aplikazioak, *grid* konputazioa, etab.). Badira ere ingurune espezifikogotarako zuzendutako aplikazio banatuak, hala nola intranet-ak (*cluster* konputazioa), etxea (aplikazio domotikoak), edota aireportu bat (nonahiko konputazioa).

⁷ <http://www.mosix.cs.huji.ac.il/>

1.4 Hardware euskarria

Lehen emandako sistema banatuen definizioa (egoera konpartitzen duten elkarkonektatutako konputagailu multzoa) nahiko zabala da, besteak beste honako galderak egin gaitetzkeelarik:

- (1) zer esanahia du *konputagailu* terminoak?
- (2) nola definitzen dira egoerak?

Galdera hauei erantzuteko konputagailuen arkitektura arloan dauden alternatibak aztertuko ditugu. Flynn-ek proposatutako irizpide klasikoan arabera, arkitekturak datu-parallelismoa edota agindu-parallelismoa eskaintzen dituzten neurrian sailka daitezke, 1. taulako lau taldeak daudelarik.

	Datu bakarra aldi berean	Datu asko aldi berean
Agindu bakarra aldi berean	SISD Von Neumann-en arkitektura klasikoak	SIMD Prozesadore bektorialak
Agindu asko aldi berean	MISD Ez daude inplementaziorik	MIMD Multiprozesadoreak, multikonputagailuak, konputagailu-sareak

1. taula. Flynn-en sailkapena.

Sailkapena egina, esan beharra dago gaur egungo prozesadoreek (segmentatuak eta supereskalarrak) neurri batean paralelotasuna eskaintzen dutela. Baina orokorrean, edozein motako baliabideen *banaketa* onartzen duten arkitekturak zehazterakoan, MIMD kategorian zentratuko gara, hainbat prozesaketa unitate⁸ duten *makinak*. Prozesaketa unitate bakoitza deitzeko **nodo** terminoa erabiltzen da askotan.

⁸ Prozesaketa unitatea makina-aginduak exekutatzeko gai den elementua da, hau da, *prozesadorea* terminologia klasikoa erabiliz.

MIMD kategoria bi irizpide jarraituz azpitaldetan bana daiteke: akoplamendu maila (nodo bakoitza memoria fisikoko helbide-esparru propioa duen ala ez), eta nodoak nola dauden elkarkonektatuta (bus konpartitua ala interkonektio sarea). 2. taulak sailkapen honek ematen dituen sistema motak erakusten ditu.

Berriro, azken sailkapen honetako azpitalde guztiak baliabideen banaketa orokorra onartzen duten ala ez galde gaitetzke. Erantzuna emateko, nodoen artean banatu daitezkeen hiru motako baliabideak hartuko ditugu kontuan:

- Prozesua
- Memoria esparrua
- S/I esparrua (fitxategiak eta dispositiboak)

	Akoplamendu maila	
	Memoria fisikoko helbide-esparru konpartitua	Memoria fisikoko helbide-esparru independenteak
Bus konpartitua	Multiprozesadore klasikoak	Multikonputagailuak, LAN sareak
Interkonektio sarea	UMA eta NUMA multiprozesadoreak	Multikonputagailuak, WAN sareak

2. taula. MIMD sistema motak.

Multiprozesadore sistemetan memoria eta S/I definizioz baliabide zentralizatuak dira. Ondorioz, makina hauetan soilik **prozesaketa banatuari** buruz hitz egin daiteke. Nodoez S/I-ren eta memoriaren egoera konpartitzen dute, prozesadoreen arteko distantzia fisikoa txikia da, eta sistema osorako erloju bakarra dago. Gaur egungo multiprozesadoreetan prozesadore bakoitzak bere cache memoria propioa du, baina cache-en koherentzia hardware mailako protokoloen bidez bermatuta dago. **Sistema eragile multiprozesadore** batek ez du printzipioz aparteko nobedaderik izaten⁹. Memoria, S/I-a eta fitxategiak sistema eragile tradizionalan bezala kudeatzen dira, soilik prozesadoreen kudeaketa eta prozesuen esleipena behar berriak sortzen dituztelarik. Hau

⁹ Hau itxuran bakarrik da. Multiprozesaketarentzako euskarri eraginkor batek ondorio inportateak izaten du sistema eragilearen egiturari.

guztiagatik, multiprozesadoreak ez dira gaur egun sistema banatuak bezala ikusten.

Aldiz, multikonputagailuetan eta konputagailu-sareetan baliabide guztiak, erlojua barne, banatuak dira (edo bana daitezke)¹⁰. Egoera globala sendoa bermatuko duten mekanismoak eta zerbitzuak inplementatu behar dira, maila onargarri batean sistema bakarraren ikuspegia eman dezaten. Hurrengo atalean mekanismo eta zerbitzu hauen egitura aztertuko da.

Bukatzeko, komunikazio sareei dagokionez, hainbat teknologia desberdin existitzen dira gaur egun, esparru gertu eta zabalentzat, bai kabledunak (USB, Ethernet, ATM), bai kablerik gabekoak ere (Bluetooth, WiFi, WiMAX, UMTS).

1.5 Baliabideen banaketarako software euskarria

Aurreko atalean sistema banatuak memoria helbide-esparru eta S/I-a propioak dituzten nodoez osatutako sistemetara mugatu ditugu. Nodo bakoitzak bere sistema eragile propioa izango du, normalean sareko oinarritzko zerbitzuak barne dituena. Deskribapen honetan konputagailu-sare estandarrak sartzen dira, baita multikonputagailuak ere.

Orokorrean nodoak heterogeneoak izango dira hardwareari eta sistema eragileari dagokionez. Hau honela izanda, lehen urrats bat espezifikazio amankomunak sortzea izango litzateke, espezifikazio hauetan oinarrituta sistema banatuaren zerbitzuak eraikitzeko.

1.5.1 Sistema irekiak

Sistema banatuetako baliabideen integrazioarako arazo nagusi bat, osagai diren sistema desberdinen **heterogeneotasuna** da, datu-komunikazioetan eta aplikazioen edota erabiltzaileen bateragarritasunean eragina duena. Sistema heterogeneoen gainean sistema banatuak eraikitzeko funtsezko oinarria, osagaiak **sistema irekiak** izatea da. Sistema ireki batek bere interfazearen **espezifikazio publikoa** eskaintzen du, ekoizleen aldetik onartua dena. Sistemaren hedapenak, normalean merkatuko arauetara egoten dena, baldintzatzen du neurri handi batean sistemaren irekitasun kontsiderazioa. Ildo honetatik, badira sistema ireki *ofizialen* proposamenak, erakundeen eta kontsorzio handien artean definituak, porrot egin dutenak. Aldi berean,

¹⁰ Multikonputagailuetan S/I-a bai konpartitua eta bai banatua izan daiteke. Orokorrean bigarren suposatuko dugu.

sorkuntzan apalagoak ziren sistemak gaur egun **de facto-ko estandarrak** bihurtu dira, adibidez TCP/IP protokolo multzoa. Bestalde, **sistema itxien** kasuan ekoizleak ez du bere interfazearen espezifikaziorik ematen, eta ondorioz bere erabilpena konfigurazio pribatuetara (*proprietary*) mugatzen da.

Sistema irekiak honako propietateak dituzte [QUA93]:

- **Interoperatibitatea.** Sarearen bitartez, makinaren artean informazioa bidali eta jasotzeko ahalmena. Komunikazio-protokolo estandarrei esker lortua (adibidez, TCP/IP).
- **Aplikazioen garraigarritasuna.** Sistemako makinaren artean programak mugitzeko ahalmena. Sistema eragile irekiek estandar ofizialetan (adibidez, POSIX) edota *de facto*-koetan oinarritutako interfaze amankomuna eskaintzen dute, aplikazioak iturburu-kode mailan garraigarriak bihurtuz¹¹.
- **Erabiltzaileen garraigarritasuna.** Erabiltzaileek makina desberdinetik sistema atzitzeko ahalmena, makina bakoitzaren ezaugarri partikularrak jakin gabe. Erabiltzaileen interfaze grafikoek (GUI), izaeraz, propietate hau eskaintzen dute neurri handi batean, naiz eta desberdinak izan daitezkeen makina desberdinetan.

Beraz, sareko protokoloek eta sistema eragileak zehazten dute nagusiki sistema baten irekitasun ezaugarria. Espezifikazioen publikotasunari esker, osagai heterogeneoak dituzten sistema banatuetan programatzaileek software maila desberdinak (askotan *middleware* izenez ezagutuak) inplementatu ditzakete, sistemaren ikuspegi bakarraren helburua lortzeko. Aurrerago ikusiko den bezala, komunikazio-eskemak eta sistema eragileen egitura egokiak diseinatu dira, sistema irekiak definitzeko eta baliabideen banaketaren euskarri eraginkorra izateko.

Esan den legez, sistema irekiek ez dute aplikazioen garraigarritasuna bermatzen exekuzio-kode mailan. Iturburu-kodea arkitektura partikular bakoitzeko konpilatu behar da. Honen aurrean bada iturburu-kodearen **interpretazio** zuzenaren alternatiba, errendimendu galera inportantea duena. Esate baterako, Java lengoia gaur egun *de facto*-ko estandarra bihurtu da (Javan

¹¹ Java aplikazioak sistema eragile desberdineko makinaren artean garraigarriak dira. Hau honela da sistema eragilearen gainean jarritako *middleware* maila bati esker (Javaren alegiazko makina, JVM), Java aplikazioei interfaze estandarra eskaintzen diena.

iturburu-kodea ez da zuzenean interpretatzen, baizik eta tarteko kodea, *bytecode* izenekoa).

Azkenik, lengoia desberdinetan idatzitako softwareen arteko komunikazioaren arazoari ere konponbidea aurkitu zaio, alde bakoitzean interfazeak definituz eta informazioen adierazpenerako formatu amankomuna erabiliz. **Interfazeen definiziorako lengoaiak** (IDL) eskaintako notazioaren bidez argumentu bakoitzaren sarrera/irteera formatua zehaztu daiteke. Lengoaia mota hauetako adibidea Sun-en IDL-XDR urruneko prozeduren deientzako dugu, eta CORBA IDL urruneko metodoen deientzako

1.5.2 Komunikaziorako euskarria

2. taulako sailkapenetik komunikazioei buruzko ezaugarriak ere ondorioatu daitezke. **Memoria konpartitua** duten sistemetan komunikaziorako primitiboak makina-aginduetan oinarritzen diren bitartean (agindu hauek busa edota interkonexio-sarearen atzipena sinkronizatzen dute, *test&set* motakoak direlarik), nodoek memoria konpartitzen ez dutenean sareko protokoloak dira **mezu trukearentzako** euskarria eskaintzen dutenak.

Aurrekoa esanda, memoriaren banaketa fisikoa¹² eta **komunikazio ereduaren** arteko bereizketa egin beharra dago. Makinaren akoplamendu mailak, komunikazio ereduaren inplementatzeko beharrezko **euskarria** baldintzatuko du, baina eredu bera bai memoria konpartituan, bai mezu trukean oinarrituta egon daiteke. Komunikazio ereduak sistema banatuaren ezaugarrietako batzuk definituko ditu, baina memoriaren banaketa fisikoa ezkutuan gelditzen da, errendimendua kontuan hartzen ez bada.

Komunikazio eredu konkretuen inplementazioek, arkitektura baten edo bestean, mekanismo desberdinak ematen dituzte, 3. taulan ikus daitezkeenak. Memoria konpartitutako arkitekturetan, aldagai konpartituen mekanismoa aspalditik aztertua izan dira, eta ondo finkatuta dago. Aldagai hauen atzipena sinkronizatzeke atzipen eksklusiboko sarrailak erabiltzen dira normalean, baita baldintza-aldagaiak eta irakurle-idazle sarrailak ere. Beste hainbeste esan daiteke mezu trukeari buruz, memoriako bufferrak erabiliz FIFO ilaren bidez

¹² Gaur egun prozesadore bakoitzari jartzen zaion cache memoria dela eta, MIMD makina guztiak (multiprozesadoreak barne) nolabaiteko memoria banaketa fisikoa dutela esan daiteke. Dena dela, multiprozesadoreen kasuan cache memoriaren sendotasuna hardwarea berak bermatzen duenez, memoria fisikoa konpartituko makinatza hartzen ditugu.

inplementatzen dena, sarritan fitxategi sisteman integraturik (UNIX-eko *fifo*-ak eta *pipe*-ak).

Akoplamendu maila baxuko sistemetan (konputagailu sareak normalean), mezu trukea dirudi komunikazio mekanismo *naturala*. Sistema banatuatan, komunikazio beharrak baliabideak kudeatzerakoan, mezu trukean oinarritutako eskema espezifikoak erabiltzera eramaten du. Orokorrean, baliabide bakoitzaren ardura gestore prozesu batek izango du. Baliabidea atzitu nahi duten prozesuek baliabideari dagokion gestorearekin komunikatu beharko dira, **bezero-zerbitzari** eskema jarraituz. Hau da, baliabideen atzipena **eskaera-erantzuna** motako protokolo baten bidez egingo da.

Komunikazio ereduak	Akoplamendu maila	
	Memoria fisikoa konpartitua	Memoria esparru independenteak
Memoria konpartitua	Aldagai konpartituak	Memoria banatua konpartitua (DSM)
Mezu trukea	UNIX motako <i>pipe</i> -ak	<i>Socket</i> -ak
Goi-mailako lengoaien semantikak	Funtzio eta prozedurei deiak, objektuen metodoei deiak	Urruneko prozedurei deiak (RPC), urruneko objektuei deiak (RMI)

3. taula. Komunikazio mekanismoak.

Bezero-zerbitzari eskema, mezu-truke mekanismo konkretu baten bidez inplementatu daiteke, adibidez UNIX-eko *socket*-en interfazea, TCP/IP eta UDP/IP protokoloetan oinarritzen dena sareko sistema bateko prozesuak komunikatzeko.

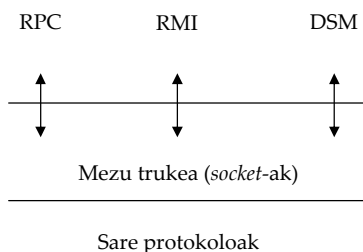
Zerbitzu lokalak atzitzen dituzten programen (sistema-deien bidez) eta zerbitzu urrunak atzitzen dituzten arteko alde semantikoa ahalik eta txikiena izateko, eskaeraren/erantzunaren bidalketaren/jasoketaren sinkronizazioa eta sareko helbideratzearen detaileak ezkutatzen duen mekanismoa garatu da, **urruneko prozeduren deiak** (*Remote Procedure Call*, RPC) izenekoa. RPC mekanismoak funtzio-deia motako sintaxia eskaintzen du zerbitzu urrunak atzitzeko interfaze bezala. A eranskinean RPC mekanismoaren deskribapen laburra aurki daiteke.

RPC mekanismoaren alternatiba bat objektuetan oinarritutakoa da. Baliabide bakoitza objektu bat da, sisteman identifikatzaile bakarra duena. Objektuak atzitzeko bere objektu-klasean definitutako metodoak deitu beharko dira.

Sistema banatua denez, objektuen atzipena **urruneko metodoen deiak** (*Remote Method Invocation*, RMI) izenez ezagutzen den mekanismoaren bidez egingo da. RMI-ri buruzko informazioa (Java-ko RMI barne) B eranskinean aurki daiteke.

Azkenik, memoria fisikoa banatuko sistemetan alegiazko helbide esparru konpartitua maneiatzea ahalbidetzen duten interfazeak diseinatu dira: **memoria banatua konpartitua** (*Distributed Shared Memory*, DSM) sistemak. Sistema hauen helburua, memoria konpartituko eredia jarraituz diseinatutako aplikazioei ingurune banatu batean exekutatzeko euskarria eskaintzea da.

1 irudian sistema banatuetaiko komunikazioaren egitura laburtzen da. Ikusten den moduan, programazio eskema desberdinak (RPC, RMI, DSM) azpiko sareko sistemak eskaintitako mezu-truke interfazean oinarritzen dira.



1 irudia: Komunikazioaren egitura sistema banatuetaiko.

Orain bezero-zerbitzari ereduko komunikazioen oinarritzko kontzeptu batzuk errepasatuko ditugu labur. Mekanismo hauen erabilera sakontzea (konkretuki TCP/IP eta UDP/IP protokoloak) Informatikan Ingeniaritzako beste ikasgai batzuen helburua da. UNIX-en komunikazioei buruz gehiago jakiteko [MIG98] eta [BRO94] kontsulta daitezke.

Mezu-truke kontzeptuak lotura (edo komunikazio kanala) baten bidez prozesuak komunikatzea ahalbidetzen duten mekanismo multzoa barneratzen ditu. Komunikazioa zuzenean prozesu igoerle eta jasotzaileen identifikadoreak jaso eta bidali primitiboetan erabilia egin daiteke, edota komunikazio kanala esplizituki identifikatzen duen buzoi baten bidez. UNIX-en, nodo bakar batean mezu trukeko oinarritzko mekanismoak *fifo*-ak (izendunak) eta *pipe*-ak (izenik gabekoak) dira. Nodo desberdineko prozesuak komunikatzerakoan helbideratzearen arazoa sortzen da. UNIX-en *socket*-ak komunikatzeko bi modu desberdin eskaintzen dute: *socket*-ari fitxategi-sisteman identifikatzaile bat emanez (UNIX motako *socket*-ak), edota IP helbide eta komunikazio portu bat

lotuz (INET motako *socket*-ak, *Internet*-en erabiliak). Azken kasu honetan helbideratzearen kudeaketa esplizitua beharrezkoa da.

Nahiz eta mezu trukeko programazio eredia berdina izan nodo desberdineko edota nodo bereko prozesuen arteko komunikazioarako, sareko protokoloetan oinarritzeak bai eraginkortasunean (atzerapenak) eta bai prozesuen arteko komunikazio kanalaren fidagarritasunean (deskonexioak) ondorioak izan ditzake.

Alde batetik, komunikazio kanala okupaturik badago, mezua bidaltzeko primitiboak prozesu igoerlea blokea dezake kanala libre gelditzen den arte (baita kanalak **buffering** onartzen duen ala ez arabera ere) mezua utzi ahal izateko (**blokeatzaile** modua, sinkronoa). **Ez blokeatzaile** modua (asinkronoa) prozesu igoerleari jarraitzen uzten dio nahiz eta bere mezua bidaltzea lortu ez, mezua gelditzen den erabiltzaile mailako bufferraren kudeaketa aplikazioaren ardura izaten delarik. Modu hauek mezuak jasotzeko primitiboari ere aplikatzen dira. UNIX-eko *socket*-ak printzipioz blokeatzaileak dira, baina ez blokeatzaile bezala konfiguratu daitezke.

Komunikazioaren fidagarritasunari dagokionez, mezu trukearen mekanismoa azpiko sareak eskaintitako euskarriaren menpe dago. Garraio mailako protokolo *fidagarri* baten gainean inplementatzen bada, komunikazioa **fidagarritzat** hartzen da, hau da, igoerleak jasotzailea mezua zuzen jasoko duela edota errore abisua jasoko duela suposa dezake. Hau gauzatzea sarean gainzama pixka bat eragiten du, protokolo fidagarrietan jasotzeak konfirmatu behar baitira. Aldiz, mekanismo **ez fidagarriek** komunikazio "merkeagoa" ahalbidetzen dute, baina komunikazioaren arrakastaren konprobaketa aplikazioari uzten diote. UNIX-ek *socket*-en bidezko bi komunikazio mota eskaintzen ditu: *konexiora zuzendutako* komunikazioa, TCP/IP protokoloan oinarritua, fidagarria, eta *datagramen* bidezko komunikazioa, UDP/IP protokoloan oinarritua, ez fidagarria.

Sistema banatuetaiko 1:N motako mezu trukeko primitiboaren euskarria oso interesgarria da. **Hedapenaren** bidez (**broadcast**) igoerle batek eskura dituen helbide guztietara mezu bat bidali dezake, sare lokaletan erabiltzen delarik. Hedapenaren kasu partikularra **multicast** motakoa da, non igoerleak helbideen azpimultzo bat aukeratu dezake mezua bidaltzeko. *Multicast* hedapenaren euskarria sistema erreplikatuetaiko oso baliagarria da, aurrerago ikusiko den bezala. Adibide gisa, IP protokoloak *multicast*-erako helbide multzoa erreserbatuak ditu.

Azkenik, *Internet*-en hazkundera sare mailako protokoloen eboluzioa behartzen ari da. Adibidez, gaur egun IP protokoloaren 4. bertsioak (IPv4), 32 biteko

helbideetan oinarritua, eskalagarritasun arazoak ditu. Hau konpontzeko IPv6 (6. bertsioa), 128 biteko helbideetan oinarritua, pixkanaka sartuaz doa.

1.5.3 Sistema eragilearen euskarria

Baliabideen banaketari dagokionez, sistema eragileari honakoa eska diezaiokegu: (1) irekia izatea, zerbitzu banatuen garapena ahalbidetzeko, eta (2) zerbitzuak kokatzeko eta eraginkor kudeatzeko beharrezko malgutasuna eskaintzea. Dena dela, sistema eragile klasikoak, adibidez UNIX, monolitikoak dira, eta ez daude diseinaturik helburu hauetara begira. Sistema-dei interfaze bakarra eskaintzen dute, zerbitzu guztiak sistemaren nukleoan (*kernel*) daude, eta baliabideak kudeatzeko politika aurredefinituak dituzte.

Konputagailu-sare batean sistema banatu bat eraikitzeke era klasikoa, makina bakoitzeko sistema eragilearen gainean bezero-zerbitzari eredu jarraituz baliabide urrunak atzitzeko mekanismoak instalatzea da. Modu honek malgutasun gutxi eskaintzen du, zeren eta baliabide baten zerbitzariak normalean sistema eragile konkretu baten exekuzioa eskatzen du. Bestalde, bezeroaren atalean, sarritan plataforma espezifikoa euskarritzat duten aplikazioak exekutatzeko beharrezkoa da. Hain malgutasun gutxiko eskemak zerbitzu eta aplikazio espezifikoko hauetarako makina konkretuak esleitzera eramaten du, malgutasunaren eta kokapen-gardentasunaren kalterako.

Sistemako edozein nodok edozein zerbitzu edo aplikazio arazorik gabe exekuta dezan soluzio bat, sistema eragile nagusi baten gainean beste sistema eragile gonbidatu baten **emulazioa** da. Honek nodo bakoitzari¹³ **pertsonalitate anitz** ematen dio, sistema eragile desberdineko aplikazio edota zerbitzuak elkarrekin exekuta ditzakeelarik, batzuk *natibo* moduan eta besteak emulatuak. Mota honetako produktuen adibideak *Virtual PC*, *VMware* eta *win4lin* dira. Hardware ingurune baten emulazioa eskaintzen dute, bere gainean sistema eragile gonbidatua instalatzeko. Produktu hauen arrakastaren arrazoi nagusia teknika berezi baten erabilpena da, *virtualizazio* izenekoa. Teknika honen bidez aplikazio "gonbidatuen" kode gehiena modu natiboan exekuta daiteke, soilik sistema eragile gonbidatuaren sistema-deiak emulatzeko direlarik. Era honetan, emulazioak berez dakarren errendimendu galera, gaur egungo makinekin, jasagarria da.

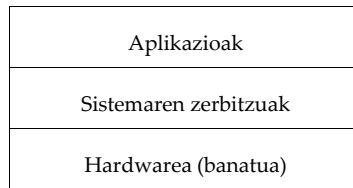
¹³ Praktikan hau terminal batean, leiho desberdinetan, sistema eragile desberdinak exekutatzeko isladatuko da.

Dena dela, orain dela gutxi arte hardware emulazioaren errendimendu galera onartezina zen, eta hainbat sistema eragile osorik gorde eta kargatu beharra garestiegia. Guzti honek 80. hamarkadaren bukaeran sistema eragileen egituraren berrantolaketara bultzatu zuen. Banatu zitezkeen zerbitzu guztiak (bere kudeaketa politikekin) sistema eragilearen esparrutik ateratzea zen helburua, sistema eragilearen esparruan, exekuzio modu babestuan, soilik hardwarearen kontrolerako oinarritzko funtzioak (memoria eta etenen oinarritzko kudeaketa) eta testuinguru-aldaketarako eta komunikaziorako euskarria utziz. Mota honetako sistema eragileari **mikronukleoa** (*microkernel*) deitzen zaio. Bere gainean, erabiltzailearen esparruan eta bezero-zerbitzari ereduari jarraituz, sistemak emango dituen zerbitzu espezifikoak soilik inplementatzeko aukera ematen duenez (modulu independente bezala), zerbitzuen banaketarako euskarri egokia eskaintzen du. Sistema-deien interfazea erabiltzaile mailako azpi-sistema bezala inplementatzen da. Ondorioz era naturalean eta gainzama txikiarekin pertsonalitate anitz eta sistema eragile desberdineko zerbitzu edota aplikazioen elkarexekuzioa eskaini ditzake. Mikronukleoen artean ezagunena Mach 3.0 da, UNIX sistema komertzial batzuen oinarri bezala erabili dena, adibidez OSF/1, eta baita Macintosh makinentzako Apple-en X sistema eragilearen oinarri bezala ere.

Gaur egungo dispositibo mugikor batzuk (PDAk, telefono mugikorrak) sistema eragile komertzialen bertsio egokituak exekutatzen dituzte, batzuk gainera Java aplikazioak eta Web nabigatzailea exekuta ditzaketelarik. Dispositibo mugikorren arloan zerbitzu urrunak atzitzeko HTTP eta SOAP protokoloen erabilpenaren aldeko joera ikusten da, RPC eskema klasikoaren eboluzio bezala. Nahiz eta dispositibo hauen prestazioak gero eta handiagoak izan, sistema eragilearen eta (haririk gabeko) sareko zerbitzuen euskarria gero eta eraginkorragoa eskaintzea beharrezkoa izaten jarraitzen du, zeren eta alde batetik dispositiboak gero eta txikiagoak dira, eta bestetik beste motako dispositiboetan (musika reproduktoreak, argazki kamera digitalak, erlojuak, txartelak...) gero eta funtzionalitate gehiago sartzen ari direlako.

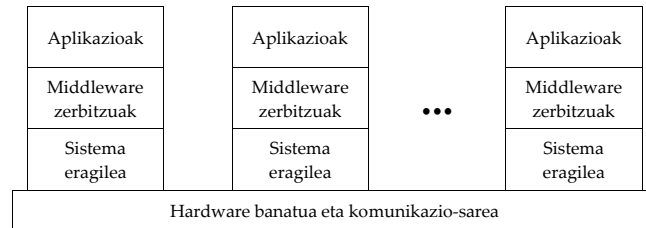
1.6 Sistema banatuen egitura

Sistema banatuen helburua komunikazio-sare baten bidez konektatutako baliabide eta zerbitzuak bateratzea da. Erabiltzaile eta aplikazioen ikuspuntutik, sistema banatuak makina bakar baten ikuspegia eskaintzen dute, eta ez dago alderik sistema zentralizatuekiko (2a irudia). Aldiz, sistemaren diseinatzailearen ikuspuntutik (hau da, sistema banatua baliabideen kudeatzaile bezala), bere barneko egitura kontuan hartu behar da, baliabideen banaketa fisikoagatik baldintzatua dagoena (2b irudia).



2a irudia: Sistema banatuak erabiltzailearen ikuspuntutik.

Elkarkonektatutako nodo multzo batetik abiatuz, nodo bakoitza bere sistema eragilearekin, software maila batek (*middleware*) sistema banatuaren zerbitzuak eta propietateak eskainiko ditu.



2b irudia: Sistema banatuen barneko egitura.

Nodo bakoitzeko sistema eragileak baliabide lokalen kudeaketarako euskarria eskainiko du, eta baita informazioa saretik bidali eta jasotzeko euskarria izango den mezu trukeko oinarritzko mekanismoa. *Middleware*-ak baliabideen banaketa gardena ahalbidetzen duen mekanismoak inplementatuko ditu, ikasturtean zehar aztertuko direnak. Mekanismo aipagarri batzuk hauek dira:

- Urruneko prozeduren deierako eta urruneko metodoen deierako euskarriak.
- Denbora eta gertaeren kudeaketa.
- *Multicast* motako komunikaziorako euskarria.
- Sendotasunaren kudeaketa.
- Sinkronizazio banaturako mekanismoak.
- Izen zerbitzuak.

- Segurtasun politikak.

Aurrerago ikusiko denez, sistema banatuetako *middleware*-ak duen konplexutasunagatik, normalean mailaka antolatzen da. Adibidez, ohizkoa da hutsegite tolerantzia den zerbitzu bat propietate konkretu batzuk betetzen dituen *multicast* motako komunikazioan oinarritzea.

1.7 Ariketak

1 UNIX motako sareko sistemetan honako aplikazioak erabiltzen dira, besteak beste: *telnet, ftp, who, finger, mail*. Eztabaidatu ezazu aplikazio hauetako zeintzuk egon beharko ez liratekeen erabiltzaileen eskura sistema banatua balitz.

2 Hurrengo taulan sistema-mota desberdinen ezaugarri batzuk agertzen dira. Bete ezazu taula, sistema bakoitzak ezaugarria betetzen duen ala ez adieraziz, zein neurrian, zein baldintzatan, ...

	Sistema zentralizatu handiak	PC multzoa (konektaturik gabe)	Sareko sistemak	Sistema banatuak
Kostu/etekina				
Baliabideen konpartizio maila				
Paralelotasuna?				
Komunikazio-aplikazioetarako euskarririk?				
Eskalagarritasun maila				
Eskuragarritasun maila / hutsegite tolerantzia				
Segurtasun arazoak				

Sistema-mota desberdinen arteko konparazioa ([TAN95]-tik egokitua).

3 Hurrengo taulak sistema-mota desberdinak aurkezten ditu. Erantzun itzazu sistema bakoitzarentzat planteatzen diren galderak, arrazoituz.

	Denbora banatuko sistema eragile klasikoa	Sistema eragile multiprozesadore	Sareko sistema eragilea	Sistema banatua
Makina birtuala bakarraren ikuspegia eskaintzen du?				
Nodo guztiek sistema eragile berdina exekutatzen dute?				
Sistema eragilearen zenbat kopia daude?				
Zein da komunikazio eredia?				
Komunikaziorako protokolo estandarrak behar dira?				
Zenbat exekuzio-ilara daude?				

3. ariketarako taula.

1.8 A eranskina: Urruneko prozeduren deiak (RPC)

Mezu-trukea bezero-zerbitzari eredu adierazteko oso erabilia den teknika da. Hain zuzen, mezu-trukean oinarrituz eskaera-erantzuna protokolo bat diseinatu daiteke zerbitzari bat atzitzeko. Horrelako protokoloek sarrera/irteerara zuzendutako semantika daukatela esan daiteke. Aldiz, baliabideen atzipena sistema eragile klasikoetan, sistema-deien interfazearen bidez, funtzioetara eginiko deiak eta bueltan jasotako emaitzetan oinarritutako semantika daukate. Baliabideen kokapenarekiko gardentasuna helburu nagusietariko bat izanik sistema banatueta, argi dago mezu-trukeak ez duela semantikarik egokiena eskaintzen baliabide urrunak atzitzeko.

Urruneko prozeduren deiko mekanismoak (RPC, 1984an Sun Microsystems-ek garatua), hutsune semantiko hau betetzen du. Mekanismo honekin, bezero prozesu batek zerbitzu bat atzitzeko funtzio deien bidezko sintaxia erabiliko du, adibidez:

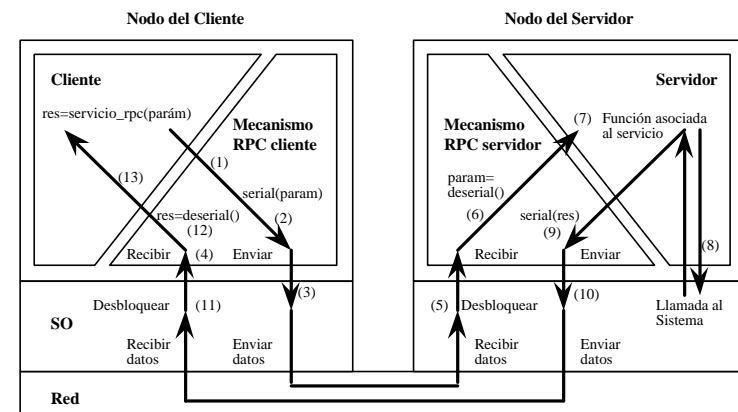
```
emaitza = rpc_zerbitzua (parametroak)
```

Funtzioaren izena atzitu nahi den zerbitzua identifikatu dezake edota orokorra izan daiteke. Azken kasu honetan, RPCen interfazeak atzitu nahi den zerbitzua parametroetako baten bidez identifikatuko du. Zerbitzua eskaintzen duen zerbitzariak bere aldetik, RPCen mekanismoaz baliatuko da eskariak zerbitzatzeko, funtzio baten bidez eskarien zain geldituko delarik blokeatuta. Hona hemen RPC baten exekuzioa ematen diren urratsak:

- (1) Atzitu nahi den zerbitzua urruna dela ikustean, bezeroak deitutako RPCen mekanismoak dagokion zerbitzaria aurkitzen du.
- (2) Parametroak formatu estandar batean (adibidez, XDR¹⁴) paketatzen dira, mezu baten gorputza osatzeko. Prozesu honi **serializazioa** deitzen zaio (*marshalling*).
- (3) Sistema eragilearen sareko interfazea erabiliz, eskaera duen mezua bidaltzen zaio zerbitzariari (adibidez, UDP/IP protokoloa erabiliz).
- (4) Bezero prozesua blokeatu egiten da, erantzunaren zain.

¹⁴ eXternal Data Representation.

- (5) Zerbitzaria dagoen nodora mezua iristean, sistema eragileak zerbitzaria desblokeatzen du. Zerbitzaria *multithread* bada, exekuzio hari bat arduratuko da eskaeraz.
- (6) Zerbitzarian itxarote-funtzioaren RPCen mekanismoa exekutatzen da, mezua edukia aztertuz (*unmarshalling*, serializazioaren kontrakoa) eskaeraren parametroak lortzeko eta eskaeraren iturria jakiteko.
- (7) Zerbitzariaren RPCen mekanismoak eskatutako zerbitzuari dagokion funtzioa exekutatzen du. Funtzio hau aurretik instalaturik egongo da, hain zuzen zerbitzaria hasieratzerakoan.
- (8) Zerbitzariaren funtzioa dei lokal bat bezala exekutatzen da.
- (9) Funtzioa amaitzen denean, RPCen mekanismoak emaitza paketatu (serializatu) egiten du.
- (10) Mezua emaitzarekin bezeroari bidaltzen zaio bueltan.
- (11) Bezeroa dagoen makinaren sistema eragileak emaitza mezua jasotzean bezeroa desblokeatzen du.
- (12) Bezeroaren RPCen mekanismoak emaitza ateratzen du jasotako mezutik (*unmarshalling*).
- (13) Emaitza bezeroaren aplikazioari bueltatzen zaio, exekuzioa jarraituz.



RPC baten exekuzioa.

Ikusten denez, zerbitzaria itxarote-funtzio baten blokeaturik egoten da, RPCen interfazea horretarako funtzio espezifiko bat duelarik. Horretaz gain, hasieratze fasean zerbitzariak eskaintzen dituen zerbitzuei dagokien funtzioak instalatu behako ditu.

RPCko liburutegi baten maila semantikoa altuagoa ala baxuagoa izan daiteke. Sun Microsystems-en RPCek hiru atzipen/erabilpen maila dituzte. Goiko mailan funtzio eta aplikazio *arruntak* (ohizkoak) aurkitzen dira. Erdiko mailan bezeroentzat orokorra den funtzio generiko bat dago, *callrpc()*, eta beste bi funtzio zerbitzarien garapenerako. Azkenik, beheko atzipen-mailan komunikazioen konfiguraziorako funtzio espezifiko batzuk aurki daitezke, hala nola dei asinkronoak (ez blokeatzaileak) egiteko aukera ematen dutenak, edota bezero eta zerbitzariaren artean konexio iraunkorra ahalbidetzen dutenak (*persistent binding*).

1.9 B eranskina: Urruneko metodoen deiak (RMI)

Urruneko metodoen deiak (RMI) objektuei zuzendutako programazioan, ohizko programazioan (prozeduretara zuzendutakoa) urruneko prozeduren deiko (RPC) mekanismoaren baliokidea da. Lehendabizi objektuen eredu orokorrean metodoen deietan erabiltzen diren kontzeptuen errepasoa egingo dugu, ondoren objektu banatuen ereduaz aztertuz.

1.9.1 Objektuen ereduaz

Objektuei zuzendutako programazio lengoaietan (C++, Java, Smalltalk), **objektu** bat, datu eta **metodo** batzuk elkartzen dituelarik, beste objektu batzuekin komunikatzen da beraien metodoak deituz. Metodo hauek normalean parametroak izaten dituzte, eta emaitza bat bueltatzen dute. Lengoi hauetan objektuen barneko aldagaiak zuzenean atzitzeko erarik egon arren, ondoren ikusiko den objektu banatuen ereduan objektuen barne-aldagaiak metodoen bidez soilik atzitu daitezke.

Objektuak erreferentziaz atzitzen dira¹⁵. Objektu baten metodoaren deian (*jasotzaile* objektua deituko diogu), jasotzailearen erreferentzia, metodoa eta deiaren parametroak adierazi behar dira. Objektuen erreferentziak aldagaiei esleitu daitezke, parametro bezala pasa daitezke, edota metodo dei baten emaitza bezala jaso daitezke.

Interfazea metodo multzo baten atzipena definitzeko erabiltzen da, hau da, metodoen izenak, parametroen datu-motak, bueltatutako emaitzak eta ez ohizko gertaerak (*exceptions*). Baina interfazeak ez du metodoen inplementazioari buruz ezer adierazten.

Metodo baten deian, objektu jasotzaileak metodoari dagokion instrukzioak exekutatzeko, eta emaitzarekin bat kontrola objektu deitzaileari pasatzen dio. Deiaren ondorioz, jasotzailearen egoera alda daiteke, baita beste metodo batzuen deiak gerta ere. Azkenik, metodoa exekutatzeko den bitartean errore-kondizioak gerta daitezke (ez ohizko gertaerak, *exceptions*).

¹⁵ Erreferentzia kontzeptu hau ez da nahastu behar C-ko *memoria-erakusle* kontzeptuarekin.

1.9.2 Objektu banatuen eredua

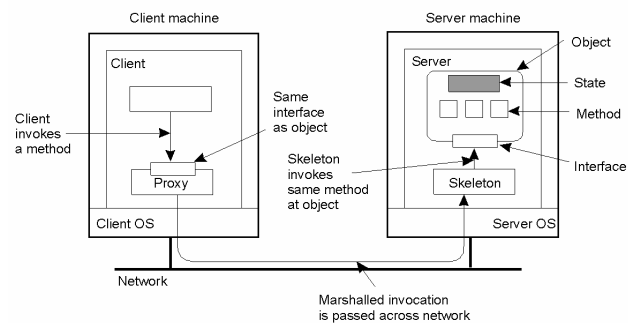
Sistema banatu batean, aplikazioen objektuak sistemako nodo desberdinetan banaturik egon daitezke. RPCetan gertatzen den bezala, **urruneko metodoen deiek** bezero-zerbitzari eskema jarraitzen dute, objektu jasotzaileak zerbitzariak kudeatzen dituztelarik. Urruneko deiak objektu jasotzaileen kokapenarekiko gardenak dira.

Objektu banatuen ereduan prozesuek objektuak dituzte, objektu hauek dei lokalak edota urrunak erabiliz komunikatzen direlarik. Dei lokalak prozesu bereko beste objektu bati egindakoak dira. Aldiz, dei urrunak beste prozesu bateko objektu bati egindakoak dira, prozesu hau nodo berean edota sistema banatuko beste nodo batean egon daitekeelarik. Objektu batzuk dei lokalak soilik jaso ditzakete, aldiz beste batzuk, **objektu urrunak**, bai dei lokalak eta bai urruneko deiak ere jaso ditzakete. Urruneko objektu baten metodo bat deitu ahal izateko, jasotzailearen **urruneko objektu-erreferentzia** izan behar da. Erreferentzia hau objektuaren identifikadore bakarra da sistema osoan.

Urruneko objektu bakoitzak **urruneko interfazea** du, bere metodoetatik urrunetik zeintzuk atzitu daitezkeen adierazten duena. Metodo hauek urruneko objektuaren klasean implementatuak egongo dira. *Java RMI*-n, urruneko interfazeak beste edozein interfaze bezala definitzen dira, *Remote* interfazea hedatu behar delarik. *CORBA*-k interfazeak definitzeko lengoia eskaintzen du (*CORBA IDL*), urruneko objektuak eta bezero aplikazioak lengoia desberdinetan programaturik egotea ahalbidetzen duena.

Urruneko deietan, objektuen eredu orokorreko ez ohizko gertaeretz gain, deiaren natura urrunarekin lotutako gertaerak ere ager daitezke (adibidez: denboragailuak, sare-konexioetan arazoak, ...).

Urruneko metodoen deiak ([TAN02]-tik hartua).



Urruneko metodoen deien eredua inplementazioa RPC eredua antzekoa da. Bezeroan, deitutako objektuaren klasearen ordezkari lokal bat (*proxy*) serializazioa eta objektu jasotzailea duen zerbitzariarekin komunikazioa inplementatzen du, era garden batean. Zerbitzarian, objektu jasotzailearen klasearen inplementazioak (*skeleton*) serializazioa eta bezeroarekin komunikazioa kudeatzen ditu, bezeroaren eskaerari dagokion metodoa deituz eta ez ohizko gertaeretz arduratuz.

1.9.3 Java RMI (*Remote Method Invocation*)

Java RMI-k *Java* lengoaiaren objektu eredua hedatzen du, urruneko objektuentzako euskarria eskainiz era integratua programazio lengoia. Honela, urruneko metodoen deiak guztiz gardenak dira, bi berezitasun izan ezik: alde batetik, bezeroek ez ohizko gertaera urrunetaz (*RemoteException*) arduratu behar dira, eta bestetik zerbitzariak urruneko objektuen interfazea definitzerakoan *Remote* interfazea hedatu behar dute.

Urruneko interfazeak *Remote* interfazea hedatuz definitzen dira, interfaze hau *java.rmi* paketeak eskaintzen duelarik. Metodoen parametroak sarrerakoak dira, eta irteera deiaren emaitzan lortzen da. Objektu urrunak erreferentziak pasatzen dira, eta objektu lokalak balioz, serializazioa aplikatuz. Jasotzaileak ez duenean balioz pasatako objektuaren inplementazioa, *Java* alegiazko makinak objektuari dagokion klasearen deskarga automatikoa eskaintzen du.

Objektu urrunak dituzten nodoek izen-zerbitzu bat eskaintzen dute, objektu urrunen erreferentziak gorde eta zerbitzatzen dituena. Zerbitzu honi *objektu urrunen erregistroa* deitzen zaio (*rmiregistry*).

Java RMI erabiliz aplikazio banatu bat egiterakoan, urruneko interfazeak definitu behar dira lehenabizi. Ondoren, objektu urrunak inplementatuko dira, baita objektu hauek atzitu dituzten bezeroak ere. Iturburu-fitxategiak konpilatu ondoren, pausu hauek jarraituko dira aplikazioa martxan ipintzeko:

1. Urruneko klaseen *proxy*-ak sortu (*Java RMI* terminologian, *stub*-ak), *RMI* konpiladorea erabiliz (*rmic*).
2. Klaseak deskargatzeko atzipen-bidea adierazi, eta segurtasun politika bat definitu.
3. Objektu urrunen erregistroa martxan ipini zerbitzarian, prozesu aparte batean (*rmiregistry*).
4. Urruneko zerbitzua martxan ipini, eta ondoren bezeroak abiatu.